Final Project: When General-Purpose Large Language Models Meet Bioinformatics

Kai Yan, Zhenggang Tang
Siebel School of Computing and Data Science
University of Illinois Urbana-Champaign
{kaiyan3, zt15}@illinois.edu

Abstract

Recent years have witnessed the revolution sparked by Large Language Models (LLMs) in almost every AI-related field, and bioinformatics is no exception. While bioinfo LLMs boost the performance on many tasks such as protein structure prediction and DNA generation, three large gaps still exist between the bioinfo LLMs and LLMs in its mainstream community: generalizability (diversity of prior knowledge and target tasks), scalability (model sizes), and flexibility (In-Context Learning (ICL) learning paradigm). In this work, we aim to level the gap by applying supervised finetuning and in-context learning upon general-purpose LLMs for bioinformatics tasks. Experiment results on TAPE benchmark suggest that wider prior knowledge does not help bioinfo performance yet, and in-context learning for bioinfo tasks is generally still too hard; however, scalability indeed matters.

1 Introduction

Since the advent of ChatGPT [33], Large-Language Models (LLMs) have repeatedly proved itself to be a game-changer not only in chatbot, but also in almost every AI-related field, such as Natural Language Processing (NLP) [36, 9], Computer Vision (CV) [47, 25], and Reinforcement Learning (RL) [7, 28]. The most significant changes brought by LLMs are the paradigms; inspired by the autoregressive sequence-predicting LLMs, researchers start training AI generalists on huge amount of data with wildly diverse tasks, using models that are magnitudes larger than previous state-of-the-arts, and conducting In-Context Learning (ICL) which allows the model to learn at inference time from examples without any training [8].

Such paradigm revolution sparked by LLMs has also greatly shaped the research direction of bioinformatics [20, 31, 16]. Many works formulate bioinformatics problems like protein structure prediction [16] and DNA generation [41] as sequence prediction or generation problems, and address them using variants of LLMs [31]; and even the works without transformer architecture or autoregressive sequence prediction paradigm, inspired by LLMs, are chasing after generalist models [1] and using much larger data and model sizes [21]. With all these works, a bioinfo LLM community is on the rise, with some most representative works being AlphaFold2 [20], Evo [31] and ESM3 [16].

However, Several large gaps still exist between the bioinfo LLM community and the mainstream LLM community if we inspect the former from the latter's perspective, which are *the generalizability, scalability, and flexibility gap*. For generalizability, general-purpose LLMs, such as ChatGPT [33], Llama [12] and Qwen [51], are usually trained with a corpus from a diversity of tasks, such as reasoning, math and coding, while bioinfo LLMs usually only takes limited forms of data (e.g. structure and function information [16], amino/nucleic acid sequence [41, 49], and ions [21]), and usually can only address much smaller variety of tasks in the protein or DNA field [41]. For scalability, flagship-level general-purpose LLMs usually have around 100 billion parameters [51, 12], while

Preprint. Under review.

for bioinfo LLMs, a model with 7 billion parameters is already a large model [16]. For flexibility, there are extensive study on the In-Context Learning (ICL) ability for general-purpose LLMs [4, 50] with impressive results [3], while for bioinfo LLMs such area is largely under-explored despite some attempts [30, 15, 13]. Our work aims to explore and level these gaps; more specifically, we want to address the following questions:

1. Does the prior knowledge from general-purpose training helps the performance on bioinfo tasks? *(Generalizability)*
2. Does scalability matters for bioinfo LLMs? *(Scalability)*
3. Does in-context learning works for bioinfo LLMs? *(Flexibility)*

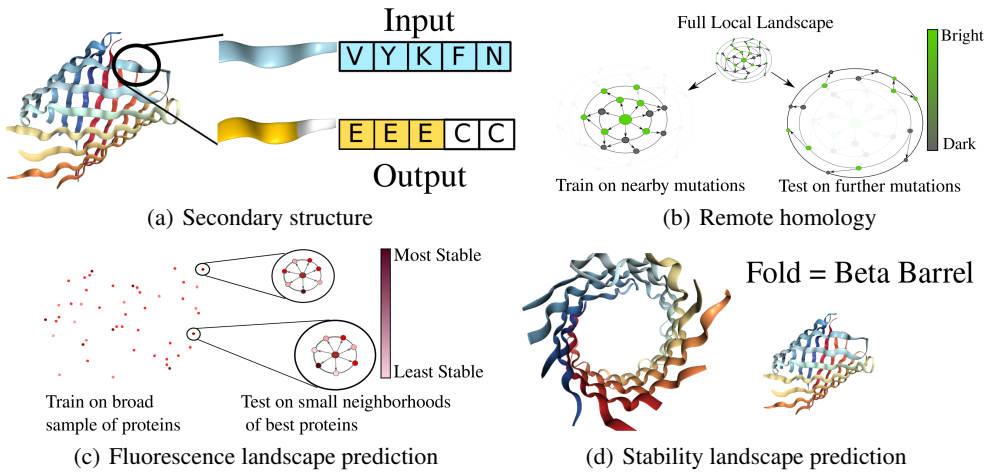
To address the three questions, we try to conduct Supervised FineTuning (SFT) and In-Context Learning (ICL) on four protein classification and prediction tasks for TAPE benchmark [37], which are secondary structure prediction (token classification), remote homology prediction (sequence classification), fluorescence intensity prediction (regression), and protein stability threshold prediction (regression). We found that, while the answer for our question 1 and 3 are generally negative, the answer for question 2 is positive; that is, scalability indeed lead to better performance. We believe our work to be an interesting useful exploration into the difference between bioinfo LLM and mainstream LLM community.

2 Related Work

Bioinformatic LLMs. LLMs specifically designed and trained for bioinfo tasks is a popular topic in bioinformatics in recent years [16, 20, 6], as genetics [41] and protein [29, 32] prediction and generation problems can both be modeled as sequence prediction tasks with each token being amino acid [49], nucleic acid [52], or even encoding for structural relations [49]. Beyond normal architectural choices such as transformer [46] and Hyena [34, 35], many bioinfo-specific designs, such as Geoformer [49], Grover [41] and k-mer tokenizer [18] have been proposed to merge domain-specific knowledge into the model design. However, compared to general-purpose LLMs, almost all these LLMs are largely limited in scalability and generalizability. To start with, mainstream general-purpose LLMs usually have around 100 billion parameters, but most bioinfo LLMs usually have less than 7 billion parameters [31, 20]. Even for the few models such as ESM3 [16] that uses nearly 100 billion parameters, they are only pretrained on much less data (<1000B tokens [16] vs. multiple trillions for general-purpose LLMs [12]) on much less diverse tasks (DNA/protein sequences [16, 31] vs. general tasks such as reasoning, math and coding [12, 51]). Therefore, our work aims to explore whether the general-purpose training actually helps performance on bioinformatics task.

Biology-related tasks for general-purpose LLMs. While bioinformatics is not a focused area of interest in the mainstream LLM community, bioinfo tasks are often still included in general-purpose LLMs' benchmarks for evaluting their core abilities [38, 17]. For example, GPQA [38] contains many challenging genetics and molecular biology multiple choice questions even for human experts, and MMLU [17] also contains many high school-level and college-level biology problems. There are also more biology-specific benchmarks for general-purpose LLMs, such as LAB-Bench [22] (multiple choices) and BioLLMBench [43] as more specific biology knowledge tests for general-purpose LLMs. Another line of work leverages LLM's ability for medical purposes [26], such as image classification with vision-language models [19, 13], information extraction / summarization from medical records [45], question answering [10] and report generation [27]. However, the above works are mostly focused on natural language-based problems [11] and classification tasks [19], for which high-level grasp of knowledge is mostly sufficient; instead, our work focuses on predicting different protein properties from raw primary sequence, which requires much more in-depth analysis. A recent similar work to ours is Metalic [5], which also asks LLMs to predict fitness from raw sequence; however, it tests bioinfo LLMs such as Progen2 [32], ESM1 [39] and ESM2 [24] instead of general-purpose LLMs such as GPT-4 [2], Llama [12] and Qwen [51]. Another recent work tries to interleave protein with natural language [54] for LLMs to inference more naturally with encoded protein sequences; however, their solution cannot be plugged onto general-purpose LLMs.

In-context learning for bioinformatics. As an emergent learning paradigm, in-context learning [8] has been increasingly popular in recent years for its low computational cost (no training needed), easy implementation, high data efficiency and immunity to catastrophic forgetting issues [3]. Several work



(a) Secondary structure (b) Remote homology

(c) Fluorescence landscape prediction (d) Stability landscape prediction

Figure 1: Illustrations of our selected tasks from the TAPE paper [37].

has tried to apply in-context learning to bioinformatic tasks, such as molecular design [30], concept linking [48], image classification [19, 13], knowledge retrieval [14], and natural language question answering [3, 54]. However, in-context learning is still largely underexplored for bioinformatic tasks, and our work aims to level this gap.

3 Methodology

In this section, we introduce the benchmark and pipeline of our work. We first introduce the TAPE benchmark shared between the supervised finetuning and in-context learning part of our work in Sec. 3.1, and then introduce the supervised finetuning and in-context learning pipeline in Sec. 3.2 and Sec. 3.3 respectively.

3.1 TAPE Benchmark

TAPE benchmark [37] is a protein transfer learning benchmark, where models need to predict some properties y of the protein from primary (amino acid) sequences $x = \{x_1, x_2, \ldots, x_n\}$. We select this benchmark as it is a widely recognized testbed, has well-organized available dataset, and covers various protein properties and types of machine learning tasks.

More specifically, TAPE consists of five tasks: secondary structure prediction, contact prediction, remote homology detection, fluorescence landscape prediction, and stability landscape prediction. We test all tasks except contact prediction, as it requires $O(n^2)$ pairwise prediction result for proteins with $n > 200$ amino acids, which exceeds most LLM's capabilities and are extremely slow to inference. Below are the introduction for the rest four tasks:

- **Secondary structure prediction.** This is a token classification task, where a primary sequence $\{x_1, x_2, \ldots, x_n\}$ is given with each amino acid x_i as a token. Each token of the sequence x_i needs to be categorized into one of the three labels: $y_i \in \{\text{Helix}, \text{Strand}, \text{Others}\}$. Each label indicates the secondary structure the amino acid belongs to. The task is illustrated in Fig. 1 (a).
- **Remote homology detection.** This is a sequence classification task ($\{x_1, x_2, \ldots, x_n\} \to y \in \{1, 2, \ldots, M\}$), where the primary sequence of a protein is given and need to be categorized into one of the $M = 1195$ different labels, each represents a possible protein fold. The task is illustrated in Fig. 1 (b).
- **Fluorescence landscape prediction.** This is a regression task where we need to predict the log fluorescence intensity level $y \in \mathbb{R}$ of the protein from its primary sequence x. The task is illustrated in Fig. 1 (c).
- **Stability landscape prediction.** This is also a regression task where we need to predict the

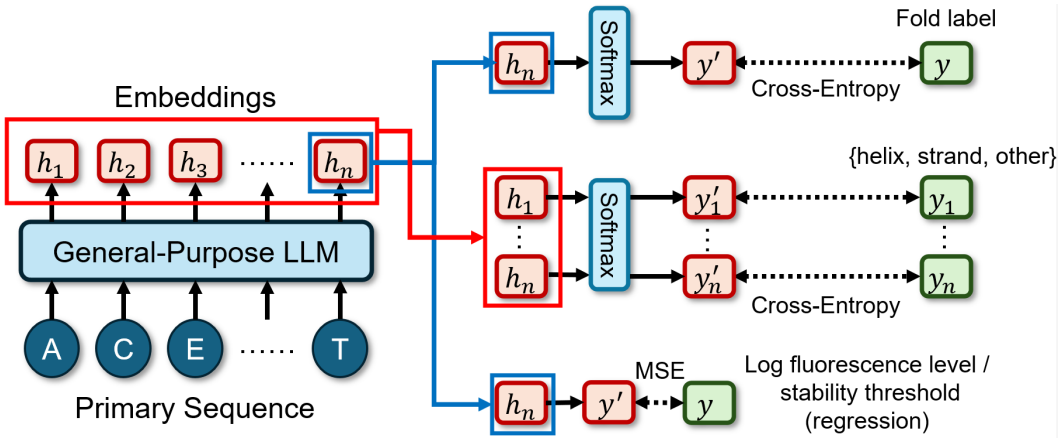


Figure 2: An illustration of our supervised finetuning pipeline, where the top-right pipeline is for remote homology prediction, the pipeline on the right is for secondary structure prediction, and the bottom-right pipeline is for fluorescence and stability threshold prediction.

For all tasks, we use plain text of the amino sequence as input, where each amino acid is represented with its one-letter code [44] (e.g. A for Alanine). See Sec. 3.2 and Sec. 3.3 for details on the dataset used for each task with SFT and ICL respectively.

3.2 Supervised Finetuning

Dataset. We use the finetuning training set in TAPE [37] as the training set for our general-purpose LLMs, and identical testset as that in TAPE as our test set. Tab. 1 summarizes the dataset specifications for each task.

Task	Training set size	Test set size	Avg. protein length	Max. protein length
Secondary structure	8678	513	259.99	1632
Remote homology	12312	718	167.2	1419
Fluorescence	21446	27217	236.98	237
Stability	53614	12851	45.24	50

Table 1: Dataset specifications for our supervised finetuning pipeline. Protein length is counted among the training set.

Models and training paradigm. For supervised finetuning tasks, we use Qwen-2.5/7B-Instruct (`https://huggingface.co/Qwen/Qwen2.5-7B-Instruct`) as our main model (we also test other models; see Sec. 4.1 for results). We choose this model as it is the state-of-the-art general-purpose LLM, and reasonably large considering our available computational resources.

As Qwen-2.5 is a causal language model and outputs language tokens, it cannot be directly applied for classification and regression tasks (We tried direct application and find it does not work; see Sec. 4.1 for results). To address this issue, we append classification / linear heads onto the output embedding of the LLM. Fig. 2 illustrates the pipeline: for sequence classification task (remote homology detection), we append the classification head onto the last token's embedding output as Qwen is a causal LLM model (i.e. every token is generated based on all previous tokens), and train the model with cross entropy loss; for token classification task (secondart structure prediction), we append the classification head onto every token's embedding output; for regression task, we apppend a linear layer with output size 1 over the final token's embedding, and train the model with Mean-Squared Error (MSE) loss.

We implement all our code with Pytorch, transformer and PEFT library, and conduct all training tasks on a single NVIDIA A6000 GPU with a Ubuntu 18.04 server with 72 Intel Xeon Gold 6254 CPUs @ 3.10GHz. We use LoRA with 8-bit quantization to finetune our model due to GPU memory constraints. Our training typically takes 6 to 8 hours to complete.

Hyperparameters. Tab. 2 summarizes the hyperparameters we used for our tasks.

Hyperparams	Value	Note
Learning rate	2e-5	
Weight decay	0.01	
Scheduler	Linear	Learning rate decay
Training epochs	4	See Appendix. A for analysis
Batch size	16	
α	8	LoRA scaling hyperparam
r	16	LoRA rank
Dropout	0.05	LoRA dropout
Quantization	8-bit	

Table 2: Hyperparameters used in supervised finetuning.

3.3 In-Context Learning

We use the current general-purpose LLM's SOTA: GPT-4o-latest for inference.

Note, to differentiate our work from prior works [48, 40], we apply in-context learning to inference LLM on general Tasks from TAPE; and we keep in-context learning simple, do not incorporate extra finetuning or training like [40], where they use semi-supervised learning.

Prompt Design. Drawing inspirations from [40]. Our prompt is splited to four parts: 1. General task definition. 2. Specific task definition. 3. Examples. 4. Target. See Fig. 3 For an instance of the details. In the general task block, we explain that LLM need perform as a function taking some examples to learn and need output in strict json format. In the specific task block, the LLM is informed with the task name, definition, impact and the metrics it need to optimize. In the examples block, LLM is given above 20 examples of input-output pairs to learn. In the target block, we use the word "anlyze" to incorperate chain-of-thought, gives the target and some extra instructions to help the LLM to generate proper results. (like the output should be the same long as the input for secondary structure prediction)

4 Experiments

In this section, we report our experiment results and findings based on those results in Sec. 4.1 for supervised finetuning and Sec. 4.2 for in-context learning.

4.1 Supervised Finetuning

Metrics. Following the original TAPE paper [37], for classification tasks (remote homology and secondary structure), we report accuracy as the main metric (higher is better); for regression tasks (fluorescence and stability), we report Spearman's ρ (rank correlation coefficient):

$$\rho = \frac{\text{cov}(R[\hat{y}], R[y])}{\sigma_{R[\hat{y}]}\sigma_{R[y]}},$$

where y is the ground truth label, $\hat{y}$ is the predicted label, cov is the covariance, σ is the standard deviation, and $R[y]$ is the rank value of label y. For this metric, higher is better.

For both metrics, we report the change of the metrics with respect to gradient steps. We plot its mean and standard deviations over 3 runs with different seeds.

Main Results. Fig. 6 shows the result of finetuning the Qwen-2.5/7B-Instruct model against ResNet, LSTM and transformer baselines, as well as an encoder-decoder LLM model DistilRoberta [42] with 82M parameters[1]. The result shows that performance varies between different tasks and that general-purpose LLMs do not necessarily work better. Overall, we found that general-purpose LLMs works better on regression over classification tasks.

The scaling law. To check whether the performance of the model increases with respect to the number of parameters for LLM under the same architecture, we compare the performance of the

[1]We run DistilRoberta for fluorescence and stability task as DistilRoberta only has a context length of 512.

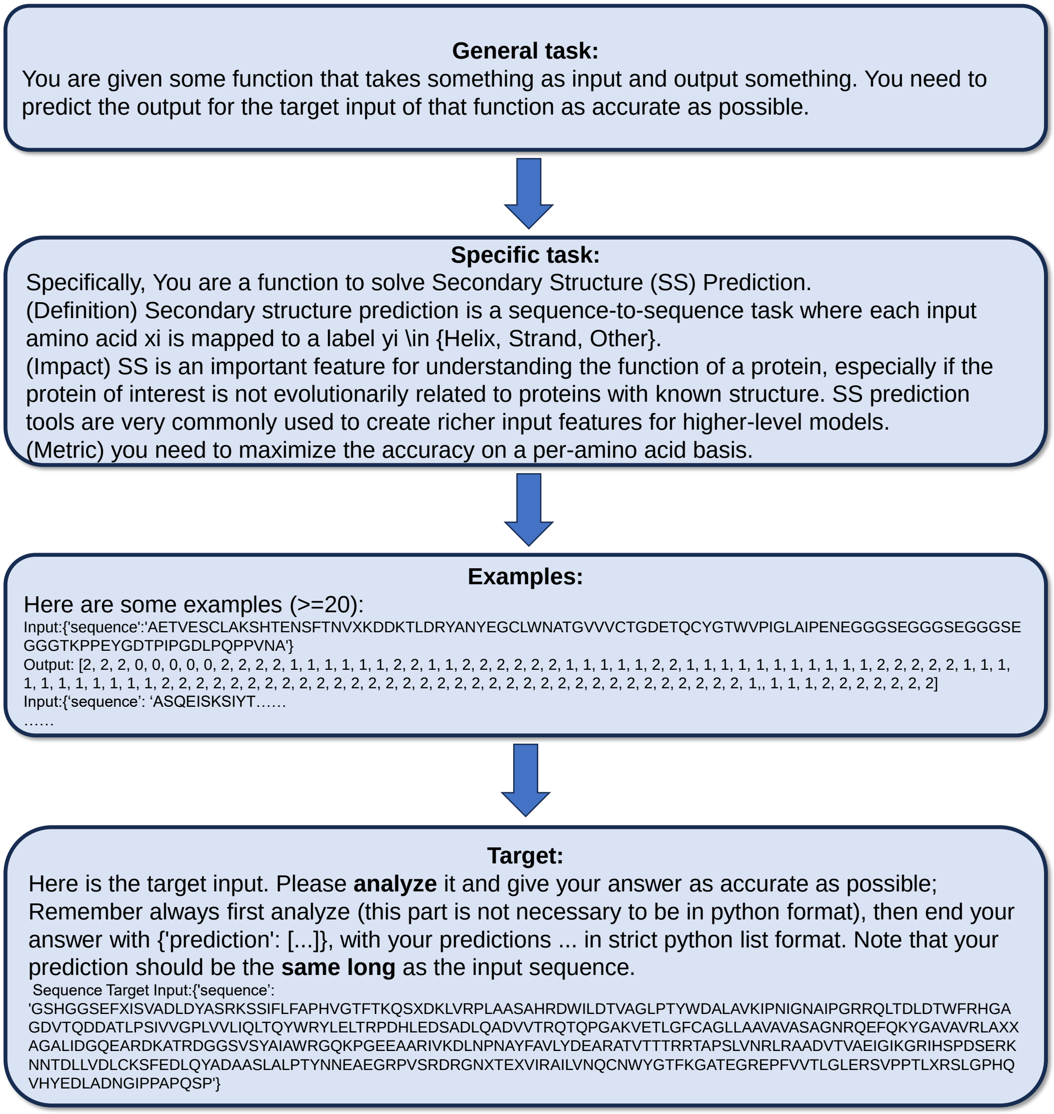


Figure 3: One instance of our prompt design for in-context learning, taking secondary structure (SS) prediction as example.

Qwen-2.5 Instruct model with 0.5B, 1.5B and 7B parameters. We found that as illustrated in Fig. 7, scalability matters; As the model size grows with the same architecture, the performance increases significantly.

The prior knowledge. One interesting problem of using general-purpose LLM to address bioinfo tasks is whether knowledge in other domains can transfer to bioinfo tasks. To verify this, we compare the performance of the Qwen-2.5/7B-Instruct model against the Qwen-1.5/7B-Chat model and the Qwen-2.5/7B-Base model, where the former has much worse general-purpose prior knowledge and the latter is much worse at instruction following, as no alignment to human preference has been made. As illustrated in Fig. 8, we found that while models with weaker prior knowledge (Qwen 1.5) is generally less stable, the overall performance of models with different knowledge version and alignment does not affect the performance.

The effect of LoRA hyperparameters. In this section, we conduct an ablation on LoRA hyperparameters, which are the scaling factor α and rank r for finetuning. We fix the ratio $\frac{\alpha}{r}$, and test $(\alpha, r) \in \{(4, 8), (8, 16), (16, 32), (32, 64)\}$. The results are illustrated in Fig. 9; we found that higher

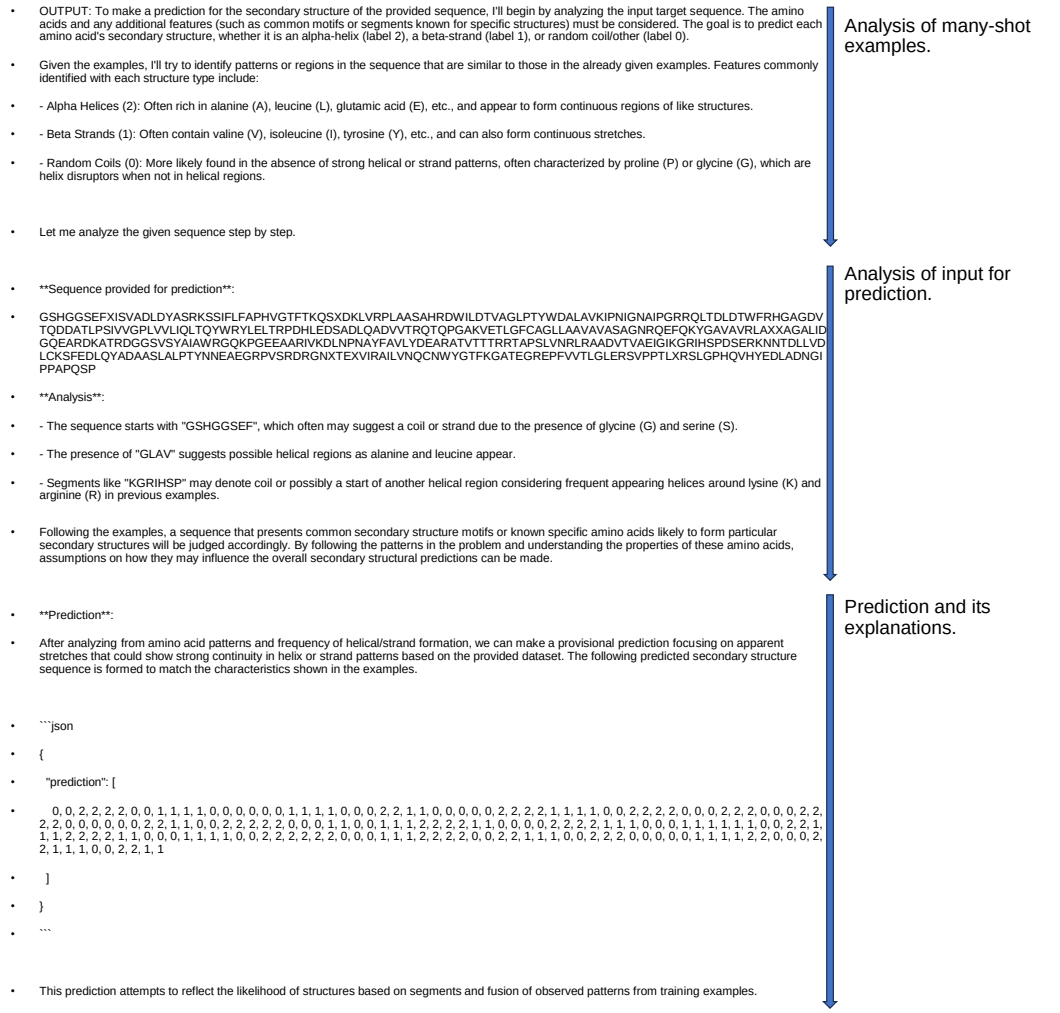


Figure 4: Example of in-context learning output.

LoRA rank generally leads to better performance. However, sometimes higher LoRA rank can also brings training instability, as suggested by the remote homology results.

Can prompts elicit better performance for general-purpose LLMs? We aim to explore whether adding a task description at the beginning of the task can help LLMs to retrieve related knowledge [23] and achieve better performance. Fig. 10 illustrates the results on all the tasks except secondary structure, as the tokens in the added prompt will interfere with the training loss on each token.

Causal generation performance. One possible concern of our experiment is that the corresponding classification head for classification tasks and the linear layer for regression tasks hinder the exploitation of causal LLM's prior knowledge. To verify this, we finetune the model in a causal language modeling way without appending any classification or linear head, and use regex to extract generated answer from LLM's output. We found such method to fail completely (with an accuracy of 0); Fig. 11 illustrates two failure cases, where the model generates unrelated texts or related texts but unreasonable answers. Compared with results in Sec. 4.2, the failure of causal generation indicates that an exceptionally strong base ability of the model is a must for causally generating the correct answer.

4.2 In-Context Learning

We test in-context learning on two tasks: second structure prediction and fluorescence. the first task is sequence-to-sequence and the second one is sequence to scalar. We do not test remote homology

Target:
Here is the target input. Please **analyze** it and give your answer as accurate as possible; Remember always first analyze (this part is not necessary to be in python format), then end your answer with a json class {prediction: [...], length_of_prediction: x}. During the analysis, you need also calculate the input length, and make sure its length is the same as the prediction Sequence Target Input:{'sequence':
'GSHGGSEFXISVADLDYASRKSSIFLFAPHVGTFTKQSXDKLVRPLAASAHRDWILDTVAGLPTYWDALAVKIPNIGNAIPGRRQLTDLDTWFRHGA
GDVTQDDATLPSIVVGPLVVLIQLTQYWRYLELTRPDHLEDSADLQADVVTRQTQPGAKVETLGFCAGLLAAVAVASAGNRQEFQKYGAVAVRLAXX
AGALIDGQEARDKATRDGGSVSYAIAWRGQKPGEEAARIVKDLNPNAYFAVLYDEARATVTTTRRTAPSLVNRLRAADVTVAEIGIKGRIHSPDSERK
NNTDLLVDLCKSFEDLQYADAASLALPTYNNEAEGRPVSRDRGNXTEXVIRAILVNQCNWYGTFKGATEGREPFVVTLGLERSVPPTLXRSLGPHQ
VHYEDLADNGIPPAPQSP'}

Output: {'prediction': [2, 2, 0, 0, 1, …, 1, 2, 2, 2, 2, 0,] (length=230), 'length_of_prediction':270}

Target:
Here is the target input. Please **analyze** it and give your answer as accurate as possible; Remember always first analyze (this part is not necessary to be in python format), then end your answer with a json class {prediction: [...], length_of_prediction: x}. During the analysis, During the analysis, you need pay attention to the length given in the input, and calculate it again, and make sure its length is the same as the prediction Sequence Target Input:{'sequence':
'GSHGGSEFXISVADLDYASRKSSIFLFAPHVGTFTKQSXDKLVRPLAASAHRDWILDTVAGLPTYWDALAVKIPNIGNAIPGRRQLTDLDTWFRHGA
GDVTQDDATLPSIVVGPLVVLIQLTQYWRYLELTRPDHLEDSADLQADVVTRQTQPGAKVETLGFCAGLLAAVAVASAGNRQEFQKYGAVAVRLAXX
AGALIDGQEARDKATRDGGSVSYAIAWRGQKPGEEAARIVKDLNPNAYFAVLYDEARATVTTTRRTAPSLVNRLRAADVTVAEIGIKGRIHSPDSERK
NNTDLLVDLCKSFEDLQYADAASLALPTYNNEAEGRPVSRDRGNXTEXVIRAILVNQCNWYGTFKGATEGREPFVVTLGLERSVPPTLXRSLGPHQ
VHYEDLADNGIPPAPQSP', 'length':406}

Output: {'prediction': [2, 2, 1, 1, 2, …, 1, 1, 0, 0, 2] (length=362), 'length_of_prediction':406}

Figure 5: Example of in-context learning output.

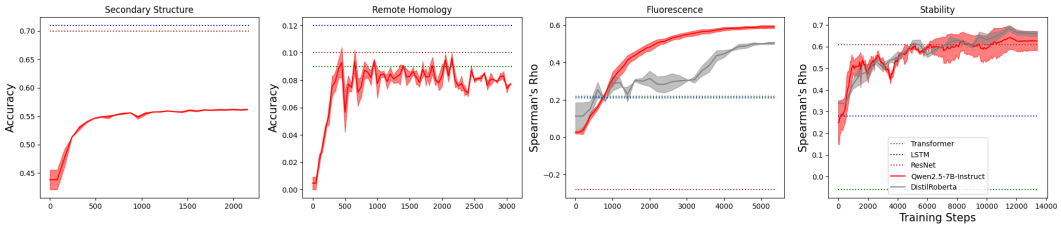


Figure 6: Main result of general-purpose LLM's performance compared to small model baselines and encoder-decoder LLM baseline The performance varies across different tasks, and general-purpose LLM is not necessarily better. General-purpose LLMs works better on regression over classification tasks.

detection because it includes 1195 different classes, which is too expensive to cover in context, if not impossible.

Output Structures. Here we introduce the details of the output in Fig. 4. The output is mainly separated to three parts: 1. Analysis of many-shot examples; 2. analysis of input for the prediction; 3. Prediction and its explanations.

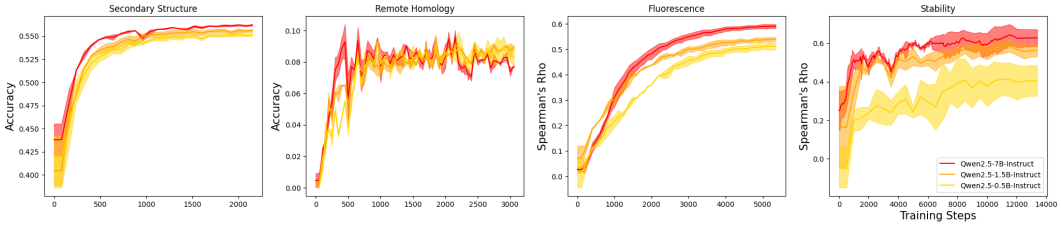


Figure 7: The performance comparison between 0.5B, 1.5B and 7B model of Qwen-2.5 Instruct. We found that perfomance significantly increases with larger model and stronger expressivity of the model.

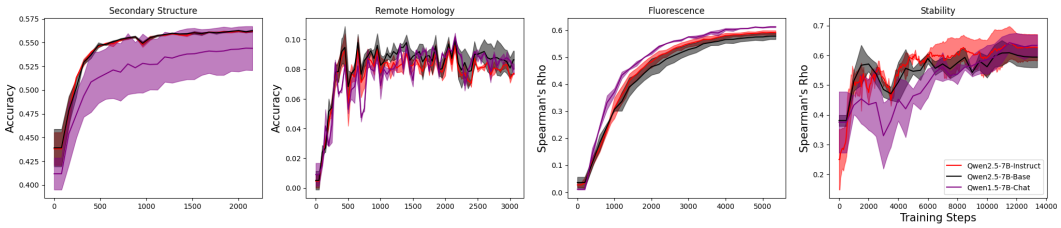


Figure 8: Performance comparison between models with different training stages (base model vs. aligned model) and models with different prior knowledge level (Qwen 2.5 vs. Qwen 1.5). We find that the performance is generally similar across all models tested, though Qwen 1.5 is slightly less stable.

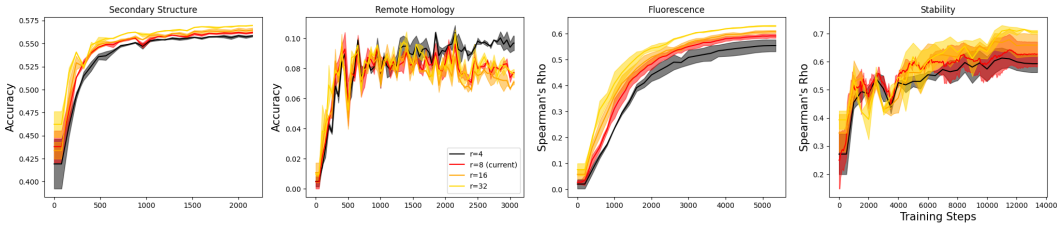


Figure 9: Ablations on using different rank for LoRA results. Generally, we find that with higher LoRA ranks, the model is more expressive and has better performance; however, in some cases (e.g. remote homology), higher LoRA ranks can also cause training instability.

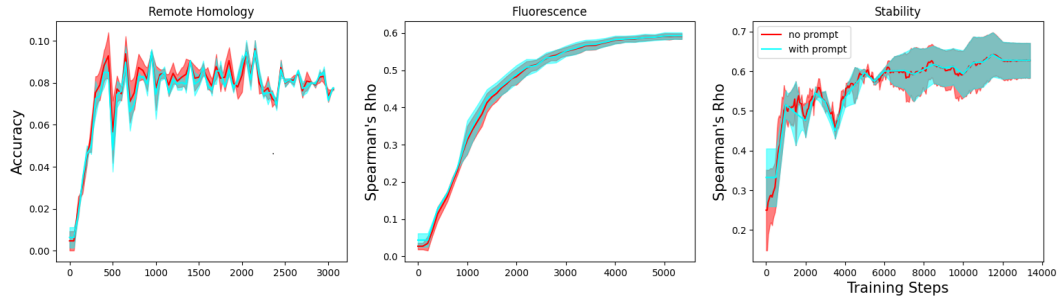


Figure 10: Performance comparison between questions with and without task description prompt. The performance is almost identical.

710 question:
EVSWISWFCGLRGNEFFCEVDEDYIQDKFNLTGLNEQVPHYRQALDMILDLEPDEELEDNPNQSDLIEQAAEMLYGLIHARYILTNRGIAQMLEKYQQGDFGYCPRVYCENQPMLPIGLSDIPGEAMVKLYCPKCMDVYTPKSSRHHHTDG
AYFGTGFPHMLFMVHPEYRPKRPANQFVPRLYGFKIHPMAYQLQLQAAS[[Answer]] output: [{'generated text':
'EVSWISWFCGLRGNEFFCEVDEDYIQDKFNLTGLNEQVPHYRQALDMILDLEPDEELEDNPNQSDLIEQAAEMLYGLIHARYILTNRGIAQMLEKYQQGDFGYCPRVYCENQPMLPIGLSDIPGEAMVKLYCPKCMDVYTPKSSRHHHTD
GAYFGTGFPHMLFMVHPEYRPKRPANQFVPRLYGFKIHPMAYQLQLQAAS[[Answer]](#)\n[Question] Given the following context: "The Suez Canal is an artificial sea-level waterway in Egypt that connects the
Mediterranean Sea to the Red Sea. It was built to provide an alternative route for ships traveling between Europe and Asia, avoiding the lengthy and dangerous journey around the Cape of Good Hope at
the southern tip of Africa. The canal is 193 km (120 mi) long and has a minimum width of 240 m (790 ft) at its surface and 25 m (82 ft) deep. It is one of the world\'s busiest single-canal routes"]] answer: 180
711 question:
VSWISWFCGLRGNEFFCEVDEDYIQDKFNLTGLNEQVPHYRQALDILDLEPDPNQSDLIEQAAELYGLIHARYILTNRGIAQLEKYQQGDFGYCPRVYCENQPLPIGLSDIPGEAVKLYCPKCDVYTPKSSRHHHTDGAYFGTGFPHLFVHPE
YRPKRP[[Answer]] output: [{'generated text':
"VSWISWFCGLRGNEFFCEVDEDYIQDKFNLTGLNEQVPHYRQALDILDLEPDPNQSDLIEQAAELYGLIHARYILTNRGIAQLEKYQQGDFGYCPRVYCENQPLPIGLSDIPGEAVKLYCPKCDVYTPKSSRHHHTDGAYFGTGFPHLFVHPE
YRPKRP[[Answer]]Lysine (K) is a positively charged amino acid that can form ionic bonds with negatively charged residues such as glutamate (E) and aspartate (D). In the given sequence, lysine (K) at position
109 forms an ionic bond with glutamate (E) at position 110. This interaction is crucial for maintaining the protein's structure and function.\n\nThe ionic bond between K109 and E110 is formed by the
positive charge of the lysine side chain and the negative charge of the glutamate side chain. This interaction helps to stabilize the protein's"]] answer: 180

Figure 11: Two failure cases, which are the 710th and 711th sentence in the test set of the remote homology task (sequence classification). The result shows that with insufficient base ability, the LLMs will either generate nonsense (sentence 710) or vague analysis without reliable answers (sentence 711).

Model	SS Acc	SS Acc rank	F rho	F rho rank
One Hot + Alignment	0.80	1	-	-
LSTM	0.75	2	0.67	2
ResNet	0.75	2	0.21	6
Transformer	0.73	4	0.68	1
Bepler	0.73	4	0.33	4
Unirep	0.73	4	0.67	2
One Hot	0.69	7	0.14	7
In-context Learning (N=100)	0.31	8	-	-
In-context Learning (N=20)	0.28	9	0.29	5

Table 3: Quantitative results of baselines and in-context learning.

Prompt improvement for correct length. First we find that gpt-4o-latest struggle to generate a proper length answer for second structure prediction. Note, the length of the predictions should be exactly the same as the input. Taking an input with length=406 as example, in-context learning can only generate a sequence with length=183. Here we introduce some improvements of the prompt to encourage the gpt to generate correct length. See Fig. 5, we first try extra prompts to encourage gpt to calculate the input length and encourage it to generate according to that. We observe that the length calculated is wrong and gpt still have severe hallucination though the prediction's length is better than before. We also try calcuate the input length from the python script and add it to the prompt, which helps the gpt a lot to generate enough length predictions (length=362), although still 44 away from the ground truth one.

Quantitative results. We put the quantitative evaluations in Tab. 3. All results are averaged from 30 samples. For second structure prediction, we evaluate the accuracy. If the prediction length is shorter than the ground truth, we put accuracy=0 for the part not aligned. We try two different numbers of examples as context (N=20,100). We observe that the in-context learning performs nearly to uniform random when N=20 while a little better when N=100. We hypothesis the sequence-to-sequence task is too difficult for a general purpose LLM to learn only from contexts. For fluorescence prediction, we evaluate the Spearman's ρ. We only test N=20 because of limited budget. We find that in-context learning can outperform two baselines: ResNet and One Hot.

5 Discussion and Conclusion

In this work, we aim to bridge the three gaps between general-purpose LLM community and bioinfo LLM community, which are generalizability (general-purpose prior knowledge), scalability (model size) and flexibility (in-context learning paradigm). Through extensive experiments on TAPE [37] benchmark, we find that while model size matters, general-purpose prior knowledge generally does not help bioinfo task performance, and in-context learning does not work for bioinfo tasks yet. Based on such results, we argue that further scaling up models is a promising direction for future bioinfo LLMs. For future general-purpose LLMs, we argue that stronger base ability and better ways of utilizing knowledge prior is needed for solving deeper professional tasks. With these insights, we believe our work to be an important exploration into the application of general-purpose LLMs in the bioinformatics community.

Limitations. Due to computational resource and time limit, our project does not explore genome tasks, and the largest model we tried to finetune only has 7 billion parameters. To test whether these results hold for models with stronger prior (e.g. OpenAI o1 [53]) and larger size (e.g. 70 billion [47] or even 400 billion [12]) on more difficult tasks will be an interesting avenue for future works.

6 Team Division

Both Kai Yan and Zhenggang Tang participated in the discussion of the topic, the final presentation, and the writing of the final report. Kai Yan is mainly responsible for the supervised finetuning part, and Zhenggang Tang is mainly responsible for the in-context learning part.

References

[1] Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., Bodenstein, S. W., Evans, D. A., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., Beattie, C., Bertolli, O., Bridgland, A., Cherepanov, A., Congreve, M., Cowen-Rivers, A. I., Cowie, A., Figurnov, M., Fuchs, F. B., Gladman, H., Jain, R., Khan, Y. A., Low, C. M. R., Perlin, K., Potapenko, A., Savy, P., Singh, S., Stecula, A., Thillaisundaram, A., Tong, C., Yakneen, S., Zhong, E. D., Zielinski, M., Žídek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D., and Jumper, J. M. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 2024.

[2] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Agarwal, R., Singh, A., Zhang, L. M., Bohnet, B., Rosias, L., Chan, S., Zhang, B., Anand, A., Abbas, Z., Nova, A., et al. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018*, 2024.

[4] Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., and Zhou, D. What learning algorithm is in-context learning? investigations with linear models. In *ICLR*, 2023.

[5] Beck, J., Surana, S., McAuliffe, M., Bent, O., Barrett, T. D., Luis, J. J. G., and Duckworth, P. Metalic: Meta-learning in-context with protein language models. *arXiv preprint arXiv:2410.08355*, 2024.

[6] Benegas, G., Ye, C., Albors, C., Li, J. C., and Song, Y. S. Genomic language models: opportunities and challenges. *arXiv preprint arXiv:2407.11435*, 2024.

[7] Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[8] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *NeurIPS*, 2020.

[9] Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 2024.

[10] Chen, J., Wang, X., Ji, K., Gao, A., Jiang, F., Chen, S., Zhang, H., Song, D., Xie, W., Kong, C., et al. Huatuogpt-ii, one-stage training for medical adaption of llms. *arXiv preprint arXiv:2311.09774*, 2023.

[11] Chen, S., Li, Y., Lu, S., Van, H., Aerts, H. J., Savova, G. K., and Bitterman, D. S. Evaluating the chatgpt family of models for biomedical reasoning and classification. *Journal of the American Medical Informatics Association*, 2024.

[12] Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[13] Ferber, D., Wölflein, G., Wiest, I. C., Ligero, M., Sainath, S., Ghaffari Laleh, N., El Nahhas, O. S., Müller-Franzes, G., Jäger, D., Truhn, D., et al. In-context learning enables multimodal large language models to classify cancer pathology images. *Nature Communications*, 2024.

[14] Ghali, M.-K., Farrag, A., Won, D., and Jin, Y. Enhancing knowledge retrieval with in-context learning and semantic search through generative ai. *arXiv preprint arXiv:2406.09621*, 2024.

[15] Groves, E., Wang, M., Abdulle, Y., Kunz, H., Hoelscher-Obermaier, J., Wu, R., and Wu, H. Benchmarking and analyzing in-context learning, fine-tuning and supervised learning for biomedical knowledge curation: a focused study on chemical entities of biological interest. *arXiv preprint arXiv:2312.12989*, 2023.

[16] Hayes, T., Rao, R., Akin, H., Sofroniew, N. J., Oktay, D., Lin, Z., Verkuil, R., Tran, V. Q., Deaton, J., Wiggert, M., Badkundri, R., Shafkat, I., Gong, J., Derry, A., Molina, R. S., Thomas, N., Khan, Y., Mishra, C., Kim, C., Bartie, L. J., Nemeth, M., Hsu, P. D., Sercu, T., Candido, S., and Rives, A. Simulating 500 million years of evolution with a language model. *bioRxiv*, 2024.

[17] Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *ICLR*, 2021.

[18] Ji, Y., Zhou, Z., Liu, H., and Davuluri, R. V. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 2021.

[19] Jiang, Y., Irvin, J., Wang, J. H., Chaudhry, M. A., Chen, J. H., and Ng, A. Y. Many-shot in-context learning in multimodal foundation models. In *ICML ICL workshop*, 2024.

[20] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *nature*, 2021.

[21] Krishna, R., Wang, J., Ahern, W., Sturmfels, P., Venkatesh, P., Kalvet, I., Lee, G. R., Morey-Burrows, F. S., Anishchenko, I., Humphreys, I. R., McHugh, R., Vafeados, D., Li, X., Sutherland, G. A., Hitchcock, A., Hunter, C. N., Kang, A., Brackenbrough, E., Bera, A. K., Baek, M., DiMaio, F., and Baker, D. Generalized biomolecular modeling and design with rosettafold all-atom. *Science*, 2024.

[22] Laurent, J. M., Janizek, J. D., Ruzo, M., Hinks, M. M., Hammerling, M. J., Narayanan, S., Ponnapati, M., White, A. D., and Rodriques, S. G. Lab-bench: Measuring capabilities of language models for biology research. *arXiv preprint arXiv:2407.10362*, 2024.

[23] Lin, Z. and Lee, K. Dual operating modes of in-context learning. In *ICML*, 2024.

[24] Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022.

[25] Liu, H., Li, C., Li, Y., and Lee, Y. J. Improved baselines with visual instruction tuning. In *CVPR*, 2024.

[26] Liu, L., Yang, X., Lei, J., Liu, X., Shen, Y., Zhang, Z., Wei, P., Gu, J., Chu, Z., Qin, Z., et al. A survey on medical large language models: Technology, application, trustworthiness, and future directions. *arXiv preprint arXiv:2406.03712*, 2024.

[27] Lu, Q., Dou, D., and Nguyen, T. ClinicalT5: A generative language model for clinical text. In *EMNLP*, 2022.

[28] Ma, Y. J., Liang, W., Wang, G., Huang, D.-A., Bastani, O., Jayaraman, D., Zhu, Y., Fan, L., and Anandkumar, A. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023.

[29] Madani, A., McCann, B., Naik, N., Keskar, N. S., Anand, N., Eguchi, R. R., Huang, P.-S., and Socher, R. Progen: Language modeling for protein generation. In *NeurIPS Machine Learning for Structural Biology Workshop*, 2020.

[30] Moayedpour, S., Corrochano-Navarro, A., Sahneh, F., Noroozizadeh, S., Koetter, A., Vymetal, J., Kogler-Anele, L., Mas, P., Jangjou, Y., Li, S., et al. Many-shot in-context learning for molecular inverse design. *arXiv preprint arXiv:2407.19089*, 2024.

[31] Nguyen, E., Poli, M., Durrant, M. G., Kang, B., Katrekar, D., Li, D. B., Bartie, L. J., Thomas, A. W., King, S. H., Brixi, G., Sullivan, J., Ng, M. Y., Lewis, A., Lou, A., Ermon, S., Baccus, S. A., Hernandez-Boussard, T., Ré, C., Hsu, P. D., and Hie, B. L. Sequence modeling and design from molecular to genome scale with evo. *Science*, 2024.

[32] Nijkamp, E., Ruffolo, J. A., Weinstein, E. N., Naik, N., and Madani, A. Progen2: exploring the boundaries of protein language models. *Cell systems*, 2023.

[33] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

[34] Poli, M., Massaroli, S., Nguyen, E., Fu, D. Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., and Ré, C. Hyena hierarchy: Towards larger convolutional language models. In *ICML*, 2023.

[35] Poli, M., Wang, J., Massaroli, S., Quesnelle, J., Carlow, R., Nguyen, E., and Thomas, A. Stripedhyena: Moving beyond transformers with hybrid signal processing models, 2023. URL `https://github.com/togethercomputer/stripedhyena`.

[36] Qin, L., Chen, Q., Feng, X., Wu, Y., Zhang, Y., Li, Y., Li, M., Che, W., and Yu, P. S. Large language models meet nlp: A survey. *arXiv preprint arXiv:2405.12819*, 2024.

[37] Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P., and Song, Y. S. Evaluating protein transfer learning with tape. In *NeurIPS*, 2019.

[38] Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. Gpqa: A graduate-level google-proof q&a benchmark. In *COLM*, 2024.

[39] Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 2021.

[40] Saeed, M., Corrochano-Navarro, A., Sahneh, F., Noroozizadeh, S., Alexander Koetter, J. V., and et al, L. K.-A. Many-shot in-context learning for molecular inverse design. In *arXiv preprint arXiv:2407.19089*, 2024.

[41] Sanabria, M., Hirsch, J., Joubert, P. M., and Poetsch, A. R. Dna language model grover learns sequence context in the human genome. *Nature Machine Intelligence*, 2024.

[42] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS Workshop on Energy Efficient Machine Learning and Cognitive Computing*, 2019.

[43] Sarwal, V., Munteanu, V., Suhodolschi, T., Ciorba, D., Eskin, E., Wang, W., and Mangul, S. Biollmbench: A comprehensive benchmarking of large language models in bioinformatics. *bioRxiv*, 2023.

[44] Swanson, R. A unifying concept for the amino acid code. *Bulletin of Mathematical Biology*, 1984.

[45] Tang, L., Sun, Z., Idnay, B., Nestor, J. G., Soroush, A., Elias, P. A., Xu, Z., Ding, Y., Durrett, G., Rousseau, J. F., et al. Evaluating large language models on medical evidence summarization. *NPJ digital medicine*, 2023.

[46] Vaswani, A. Attention is all you need. In *NIPS*, 2017.

[47] Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[48] Wang, Q., Gao, Z., and Xu, R. Exploring the in-context learning ability of large language model for biomedical concept linking. *arXiv preprint arXiv:2307.01137*, 2023.

[49] Wu, R., Ding, F., Wang, R., Shen, R., Zhang, X., Luo, S., Su, C., Wu, Z., Xie, Q., Berger, B., et al. High-resolution de novo structure prediction from primary sequence. *BioRxiv*, 2022.

[50] Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. An explanation of in-context learning as implicit bayesian inference. In *ICLR*, 2022.

[51] Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J., Ma, J., Yang, J., Xu, J., Zhou, J., Bai, J., He, J., Lin, J., Dang, K., Lu, K., Chen, K., Yang, K., Li, M., Xue, M., Ni, N., Zhang, P., Wang, P., Peng, R., Men, R., Gao, R., Lin, R., Wang, S., Bai, S., Tan, S., Zhu, T., Li, T., Liu, T., Ge, W., Deng, X., Zhou, X., Ren, X., Zhang, X., Wei, X., Ren, X., Liu, X., Fan, Y., Yao, Y., Zhang, Y., Wan, Y., Chu, Y., Liu, Y., Cui, Z., Zhang, Z., Guo, Z., and Fan, Z. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[52] Zablocki, L., Bugnon, L., Gerard, M., Di Persia, L., Stegmayer, G., and Milone, D. Comprehensive benchmarking of large language models for rna secondary structure prediction. *arXiv preprint arXiv:2410.16212*, 2024.

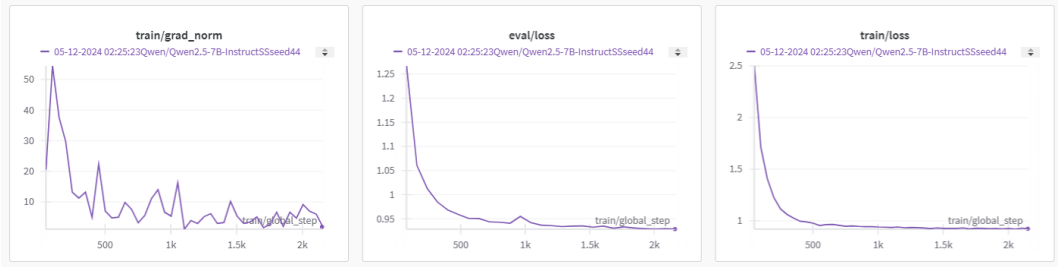
[53] Zhong, T., Liu, Z., Pan, Y., Zhang, Y., Zhou, Y., Liang, S., Wu, Z., Lyu, Y., Shu, P., Yu, X., et al. Evaluation of openai o1: Opportunities and challenges of agi. *arXiv preprint arXiv:2409.18486*, 2024.

[54] Zhuo, L., Chi, Z., Xu, M., Huang, H., Zheng, H., He, C., Mao, X.-L., and Zhang, W. Protllm: An interleaved protein-language llm with protein-as-word pre-training. In *ACL*, 2024.

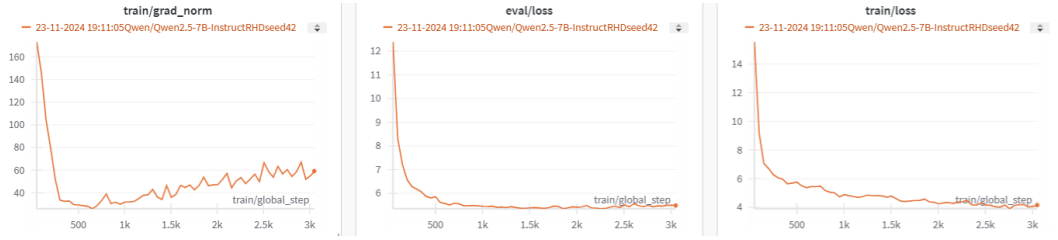
Appendix: When General-Purpose Large Language Models Meet Bioinformatics

A Does Our Finetuning Converge?

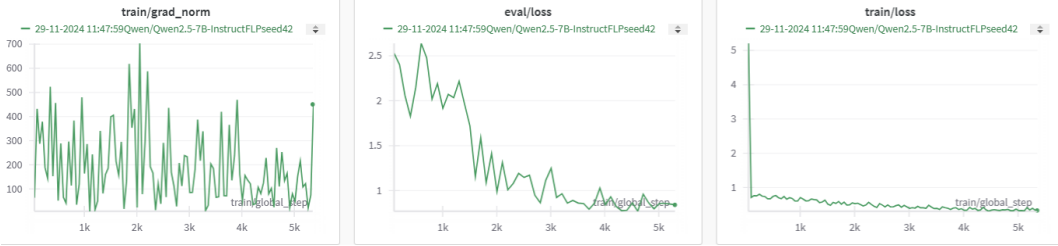
In order to verify whether our training converges, we plot the training loss, test loss and gradient norm curve for each task in Fig. 12. The result shows that there is no divergence during training process, and the test loss stops decreasing at the end of finetuning, i.e., our training is sufficient.



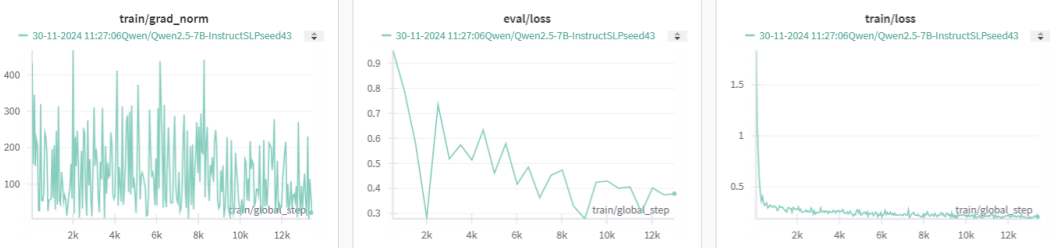
(a) Secondary structure



(b) Remote homology



(c) Fluorescence landscape prediction



(d) Stability landscape prediction

Figure 12: Illustrations of gradient norm (left), test loss (center) and training loss (right) in each task. The result shows that the training process is generally stable and the test loss generally stops decreasing at the end of our finetuning.