

TourSynbio: A Multi-Modal Large Model and Agent Framework to Bridge Text and Protein Sequences for Protein Engineering

Yiqing Shen^{1,2,†}, Zan Chen^{1,†}, Michail Mamalakis^{3,†}, Yungeng Liu^{1,6,†,§},
Tianbin Li⁴, Yanzhou Su⁴, Junjun He⁴, Pietro Liò³, Yu Guang Wang^{1,4,5,*}

¹*Toursun Synbio, Shanghai, China*

²*Department of Computer Science, Johns Hopkins University, Baltimore, USA*

³*Department of Computer Science and Technology, University of Cambridge, Cambridge, UK*

⁴*Shanghai AI Laboratory, Shanghai, China*

⁵*Institute of Natural Sciences, Shanghai Jiao Tong University, Shanghai, China*

⁶*Department of Computer Science, City University of Hong Kong, Hong Kong, China*

[†]Equal contribution. [§]Work done during the internship at Toursun Synbio. ^{*}Corresponding author.

yiqingshen1@gmail.com

yuguang.wang@sjtu.edu.cn

Abstract—The structural similarities between protein sequences and natural languages have led to parallel advancements in deep learning across both domains. While large language models (LLMs) have achieved much progress in the domain of natural language processing, their potential in protein engineering remains largely unexplored. Previous approaches have equipped LLMs with protein understanding capabilities by incorporating external protein encoders, but this fails to fully leverage the inherent similarities between protein sequences and natural languages, resulting in sub-optimal performance and increased model complexity. To address this gap, we present TourSynbio-7B, the first multi-modal large model specifically designed for protein engineering tasks without external protein encoders. TourSynbio-7B demonstrates that LLMs can inherently learn to understand proteins as language. The model is post-trained and instruction fine-tuned on InternLM2-7B using ProteinLMDataset, a dataset comprising 17.46 billion tokens of text and protein sequence for self-supervised pretraining and 893K instructions for supervised fine-tuning. TourSynbio-7B outperforms GPT-4 on the ProteinLMBench, a benchmark of 944 manually verified multiple-choice questions, with 62.18% accuracy. Leveraging TourSynbio-7B’s enhanced protein sequence understanding capability, we introduce TourSynbio-Agent, an innovative framework capable of performing various protein engineering tasks, including mutation analysis, inverse folding, protein folding, and visualization. TourSynbio-Agent integrates previously disconnected deep learning models in the protein engineering domain, offering a unified conversational user interface for improved usability. Finally, we demonstrate the efficacy of TourSynbio-7B and TourSynbio-Agent through two wet lab case studies on vanilla key enzyme modification and steroid compound catalysis. Our results show that this combination facilitates protein engineering tasks in wet labs, leading to higher positive rates, improved mutations, shorter delivery times, and increased automation. The model weights are available at <https://huggingface.co/tsynbio/Toursynbio> and codes at <https://github.com/tsynbio/TourSynbio>.

Index Terms—Deep Learning, Multi-modal Large Model, Protein Engineering, AI Agent.

I. INTRODUCTION

Protein engineering enables the modification and optimization of protein sequences or structures for diverse applications, revolutionizing our ability to manipulate biological systems at the molecular level [1], [2]. Deep learning approaches offer higher efficiency and better performance in protein engineering tasks by efficiently processing vast amounts of protein data, capturing complex patterns in sequences, predicting protein properties and structures with increasing accuracy, and facilitating rapid exploration of optimal protein designs [3]–[5]. In protein engineering, protein sequences serve as the fundamental data format, often referred to as the “language of life sciences” due to their role in encoding biological information [6], [7]. The inherent sequential similarities between protein sequences and natural language have already led to the parallel development of foundation models, namely protein language models [8] and large language models (LLMs) [9]. Because LLMs have demonstrated strong capabilities in text understanding [10], recent research has begun exploring their potential in protein understanding through multi-modal large models [11], [12]. Previous attempts have focused on integrating protein sequences or structures in the form of protein graphs with textual content using extra encoders [13], [14], as depicted in Fig. 1(a). However, these approaches fail to fully leverage the intricate connections between protein sequences and natural language, leading to higher model complexity and sub-optimal performance. This limitation highlights the need for a model that can directly understand and process protein sequences without relying on external encoders, potentially improving both efficiency and performance in protein engineering tasks. To address this gap, we present TourSynbio-7B, the first multi-modal large model specifically designed for protein engineering tasks without external protein encoders as shown in Fig. 1(b). TourSynbio-7B is post-trained and

instruction fine-tuned on InternLM2-7B [15] using ProteinLM-Dataset [16], demonstrating that LLMs themselves can learn to understand proteins in the form of language.

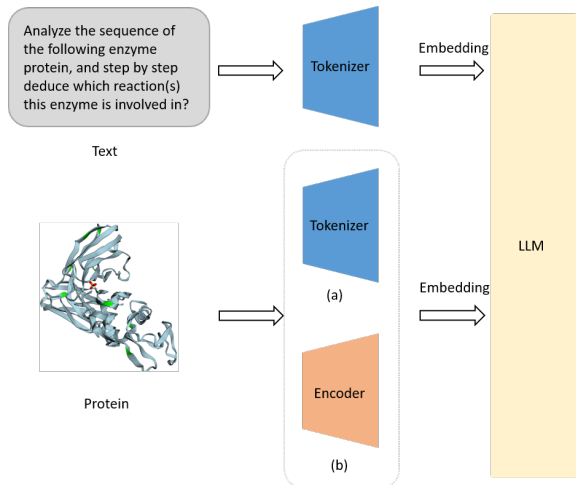


Fig. 1. Illustration of the traditional LLM and TourSynbio-7B for protein sequence understanding. (a) Traditional methods use external encoders to process protein sequences before feeding them into LLMs. (b) The proposed TourSynbio-7B method directly processes protein sequences using an LLM without the need for external encoders, thus simplifying the workflow and improving efficiency.

However, the conversation capabilities in multi-modal large models like TourSynbio-7B are not sufficient for protein engineering tasks, as they need to address complex challenges like protein structure prediction [17], evolution analysis, and *de novo* protein design [18], [19]. AI agent frameworks offer a potential solution by allowing LLMs to address domain-specific tasks through relevant tools [20]. However, current implementations of AI agents typically limit themselves to common APIs and simple AI models, such as text-to-image generation [21]. Protein engineering involves more complex tasks and APIs, arising from the need to integrate various specialized tools and models, such as protein folding models [22], molecular dynamics simulations [23], structure-function prediction tools [24], [25] and *etc* [26], [27] that takes protein sequence as input. This highlights the need for AI agents that can effectively integrate complex, domain-specific protein engineering APIs or models.

Previous agent works like ProteinEngine attempt to build an AI agent framework for protein engineering based on GPT-3.5 by assigning three roles to LLMs, namely task delegation, specialized task resolution, and effective communication of results [28]. However, GPT-3.5 lacks specialized training in protein sequences, hence it struggles to accurately interpret protein motifs, predict mutation impacts, and understand sequence-structure relationships in the agent. Additionally, ProteinEngine requires calling the LLM multiple times, which reduces efficiency. To address these limitations, we propose TourSynbio-Agent, an innovative framework built upon TourSynbio-7B’s enhanced protein sequence understanding capabilities. TourSynbio-Agent integrates previously dis-

connected deep learning models in the protein engineering domain, offering a unified conversational user interface for improved usability while preserving efficiency with a keyword-matching scheme for calling different models.

The major contributions of this paper are four-fold. Firstly, we present TourSynbio-7B, a novel multi-modal large model that bridges the protein sequence and text understanding. It demonstrates LLMs can inherently understand proteins as a new language without relying on external protein encoders. Second, we present how the user can utilize TourSynbio-7B as an innovative agent framework to perform complex protein engineering tasks, including mutation analysis, inverse folding, and visualization. Third, we present a human-centered interface that combines TourSynbio-7B and TourSynbio-Agent, providing researchers and engineers with a unified conversational user interface. This interface simplifies the execution of complex protein engineering workflows, enhancing usability and accessibility for both experts and non-experts in the field. Finally, we demonstrate the practical efficacy of our approach through two wet lab case studies, showcasing how the combination of TourSynbio-7B and TourSynbio-Agent facilitates real-world protein engineering tasks, leading to higher positive rates, improved mutations, shorter delivery times, and increased automation in wet lab settings.

II. METHODS

A. TourSynbio-7B

TourSynbio-7B is built upon the InternLM2-7B [15], leveraging its language understanding capabilities for both English and Chinese as a foundation for protein sequence comprehension. InternLM2-7B utilizes a transformer-based architecture with 32 layers, 4096 hidden dimensions, and 8 attention heads [15]. For post-training and instruction fine-tuning, we utilized the ProteinLMDataset [16], a dataset specifically designed for protein sequence understanding. Specifically, it comprises 17.46 billion tokens for self-supervised learning and 893,000 instructions for supervised fine-tuning. The self-supervised portion is strategically categorized into three segments, namely Chinese-English text pairs in protein science (0.69%), protein sequence-English text pairs (41.51%), and protein-related English text (57.80%). This diverse composition of ProteinLMDataset enables TourSynbio-7B to develop a robust understanding of protein-related concepts across multiple languages and data formats. The post-training process involved two main phases, namely the initial training on 4K context texts, followed by a transition to high-quality 32K texts for further training, which allows TourSynbio-7B to effectively capture long-term dependencies in protein sequences and related scientific literature. The instruction fine-tuning phase utilized the 893,000 instructions from ProteinLMDataset, covering various protein engineering tasks and concepts, to enhance the TourSynbio-7B’s ability to follow complex protein-related instructions and perform specific protein engineering tasks. All the training was conducted using the XTuner framework¹,

¹<https://github.com/InternLM/xtuner>

TABLE I

OVERVIEW OF THE DIVERSE FUNCTIONALITIES INTEGRATED INTO TOURSYNBIO-AGENT. IT SHOWCASES THE RANGE OF SPECIALIZED MODELS AND TOOLS INCORPORATED INTO OUR SYSTEM, COVERING CRITICAL ASPECTS OF PROTEIN ENGINEERING SUCH AS MUTATION PREDICTION, PROTEIN FOLDING AND INVERSE FOLDING, PROTEIN DESIGN AND RECONSTITUTION, AND AUXILIARY TOOLS FOR SEQUENCE ANALYSIS AND VISUALIZATION. EACH MODEL IS DESCRIBED WITH ITS SPECIFIC FUNCTION, INPUT REQUIREMENTS, AND OUTPUT FORMAT, ILLUSTRATING THE BREADTH AND DEPTH OF CAPABILITIES OFFERED BY TOURSYNBIO-AGENT IN ADDRESSING COMPLEX PROTEIN ENGINEERING TASKS.

Functionality	Model	Description	Input	Output
Mutation Prediction	ESM-1v [25]	Zero-shot prediction of mutation effects on protein function	Protein sequence, mutation offset, and additional parameters	CSV file with detailed mutation effects
	SaProt [24]	Protein language model integrating sequence and structure information	Sequence and structure information in a letter string format	Quantitative mutation score and mutation dictionary
Protein Folding and Inverse Folding	ESMFold [22]	Structure prediction using masked language modeling without multiple sequence alignment	Raw protein sequence	Residue-level pLDDT and PDB file of predicted structure
	ESM-IF1 [29]	Transformer-based model for single-site mutation prediction	Original protein sequence	Novel protein sequence with predicted mutations
	GraDE-IF [30]	Graph-based model for inverse folding	Protein sequence	<i>De novo</i> protein sequence
Protein Design and Reconstitution	AntiFold [31]	Model for designing sequences fitting antibody variable domain structures	PDB code, heavy chain sequence, and additional parameters	CSV file with design metrics and FASTA file of designed sequences
	Chroma [32]	Generative model for programmatic protein design	Desired chain length and design constraints	PDB file of the generated protein
	LigandMPNN [33]	Enhanced version of ProteinMPNN for protein reconstruction considering ligand interactions	PDB code, temperature parameter, and additional constraints	Reconstructed PDB file with optimized protein structure
Codon Analysis	CaLM [27]	Generate high-dimensional embeddings for biological sequences	DNA or protein sequence	Numerical embedding vector representing sequence features
Visualization	PyMOL	3D visualization tool for protein molecules	PDB code or file	Interactive 3D visualization of the protein structure

which enables efficient scaling across multiple GPUs. Through this training process, TourSynbio-7B has developed the capability to understand and process protein sequences as a form of language, without the need for external protein encoders.

B. TourSynbio-Agent

Building upon TourSynbio-7B, we elaborate on TourSynbio-Agent, the AI agent framework designed to execute complex protein engineering models, as illustrated in Fig. 2. TourSynbio-Agent is built on the Lagent framework [34].

1) *Intent Classification*: In this stage, TourSynbio-7B analyzes the user’s input to determine whether a specific agent needs to be called. When certain words or phrases like “why” or “introduce” typically indicate that the user is seeking general information rather than requesting an agent-specific task. These inputs are directed to the standard dialogue process. For inputs lacking clear keywords, TourSynbio-7B utilizes a custom-designed prompt to classify the user’s intent.

2) *Keyword-Based Agent Selection with Fuzzy Matching*: Given the integration of 10 different functional agents in TourSynbio-7B as depicted in Table I, relying solely on the language model to determine agent calls could lead to inaccuracies. To enhance the accuracy of agent call request recognition, we implemented a keyword-matching mechanism, which utilizes independent keyword tables for each agent, containing specific terms relevant to their function. For instance, the inverse folding agent’s table includes keywords such as “inverse folding” and “reconstruction”. When TourSynbio-7B

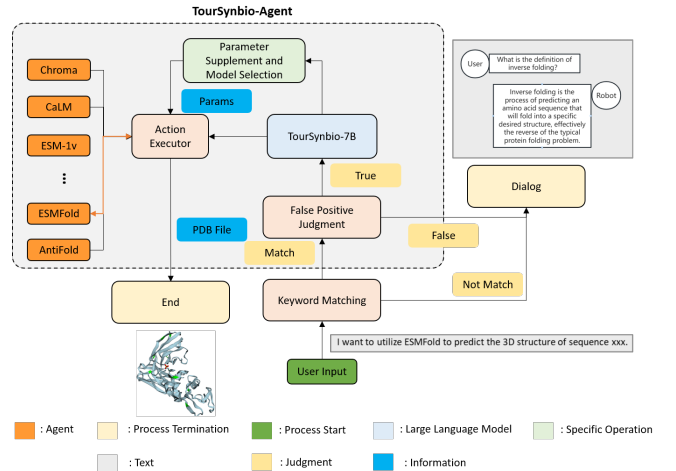


Fig. 2. The overall workflow of TourSynbio-Agent from initial user input to final task execution. It encompasses several stages: (1) Intent classification using TourSynbio-7B to determine if an agent call is required; (2) Keyword-based agent selection with fuzzy matching to identify the appropriate agent; (3) User-guided selection for functionally similar agents when necessary; (4) Parameter extraction and interactive validation to ensure accurate inputs; and (5) Agent execution. The process incorporates multiple decision points and feedback loops, allowing for efficient handling of various scenarios, including false positives and ambiguous user inputs. This design ensures robust and accurate execution of protein engineering tasks while maintaining a user-friendly interface.

receives user input, it systematically traverses the keyword table of each agent, checking for matches. If corresponding keywords are identified, the system activates the appropriate agent for processing. If no match is found, it proceeds to check the next agent’s keyword table. This design of independent keyword tables effectively prevents erroneous agent activation. To further improve the accuracy and robustness of the agent workflow and enhance user experience, we incorporated fuzzy matching capabilities, which accommodate potential spelling errors by users. It ensures that even if a keyword is input incorrectly, the system can still recognize and call the corresponding agent accurately. For example, minor misspellings like “inverse foding” or “recontruction” would still trigger the inverse folding agent.

TourSynbio-Agent Intent Classification Prompt:

You are an information extraction assistant. You need to determine whether the user’s input intends to call any agent and return *True* or *False*.

It should be noted that when a user inputs a sequence of letters or numbers in text, we tend to assume that the user is entering the parameters of the agent.

But you need to note that model names do not belong to the category of alphabetical sequences, such as Chroma, CaLM, SaProt, ESM-IF1. When these sequences appear, you need to carefully determine whether the user is in a normal conversation or wants to call the agent.

The example is as follows:

Q: Reverse fold PDB file 5YH2 with chain ID A, temperature set to 1.0, and sampling quantity of 1.

A: {*True*}

Q: What is inverse folding?

A: {*False*}

Please extract information from the following sentence:

Q: user_input

3) *User-Guided Selection for Functionally Similar Agents:* TourSynbio-Agent incorporates an optional user selection function to address scenarios where multiple agents possess similar functionalities. For example, in the case of mutation prediction, where both SaProt [24] and ESM-1v [25] agents offer highly correlated capabilities. In such instances, conventional keyword matching might not accurately direct users to the most suitable agent. To address this, we have implemented a selection interface for agents with overlapping functions. When the system identifies a specific keyword (such as “mutation prediction”) that could potentially trigger multiple agents, it prompts the user to manually select the desired agent. This interface presents users with a choice between the available options, such as SaProt and ESM-1v for mutation prediction tasks, where users can thus make informed decisions based on their specific requirements. This user-driven selection process not only mitigates the limitations of keyword-based routing but also empowers users with greater

control over their protein engineering workflows.

4) *Parameter Extraction and Interactive Validation:* After successful keyword matching, TourSynbio-7B employs a validation process to confirm the user’s intended agent call, thereby mitigating false positives that may arise during the keyword-matching phase. This stage operates on the assumption that users are cognizant of their desired agent and its associated parameter requirements. To facilitate accurate parameter extraction, we have developed tailored information extraction methods for each agent using prompt engineering by leveraging TourSynbio-7B’s natural language processing capabilities to parse user input effectively. For instance, consider the following prompt designed for the ESMFold [22] agent:

ESMFold Agent Parameter Extraction Prompt:

You are an information extraction assistant. Please extract the protein sequence from the following sentence and return it in the format of ‘{sequence: xxx}’.

Example:

Q: Perform structure prediction on the sequence QERLKSIVRILERSKEPVSG.

A: {sequence: QERLKSIVRILERSKEPVSG}

Please extract information from the following sentence:

Q: user_input

This prompt utilizes a few-shot prompting approach to enhance TourSynbio-7B’s ability to parse information accurately. It returns results in JSON format, facilitating efficient information extraction and subsequent processing. Upon receiving user input, TourSynbio-7B attempts to extract all required parameters. If any essential parameters are missing or ambiguous, TourSynbio-7B generates an interactive interface. This interface allows users to verify the accuracy of extracted parameters, ensuring that the agent operates with the correct inputs. It also provides an opportunity for users to supply any optional parameters that may enhance the agent’s performance or customize its output.

5) *Agent Execution:* After completing all the previous steps, our TourSynbio-7B calls the corresponding agent to execute the task and return the result. If keyword matching fails to identify a suitable agent but the user’s intention is clear through text analysis, TourSynbio-7B automatically determines which agent to call and parses the required parameters to complete the task.

C. Human-centered Conversational User Interface

Our human-centered conversational user interface further enhances the user experience and system usability of TourSynbio-7B and TourSynbio-Agent which provides an intuitive and user-friendly interface, as illustrated in Fig. 3. The interface design is based on the following key principles:

- **User-friendliness.** The interface adheres to principles of simplicity and intuitiveness, enabling effortless user interaction. Leveraging TourSynbio-7B to process nature language and protein sequence, users can input

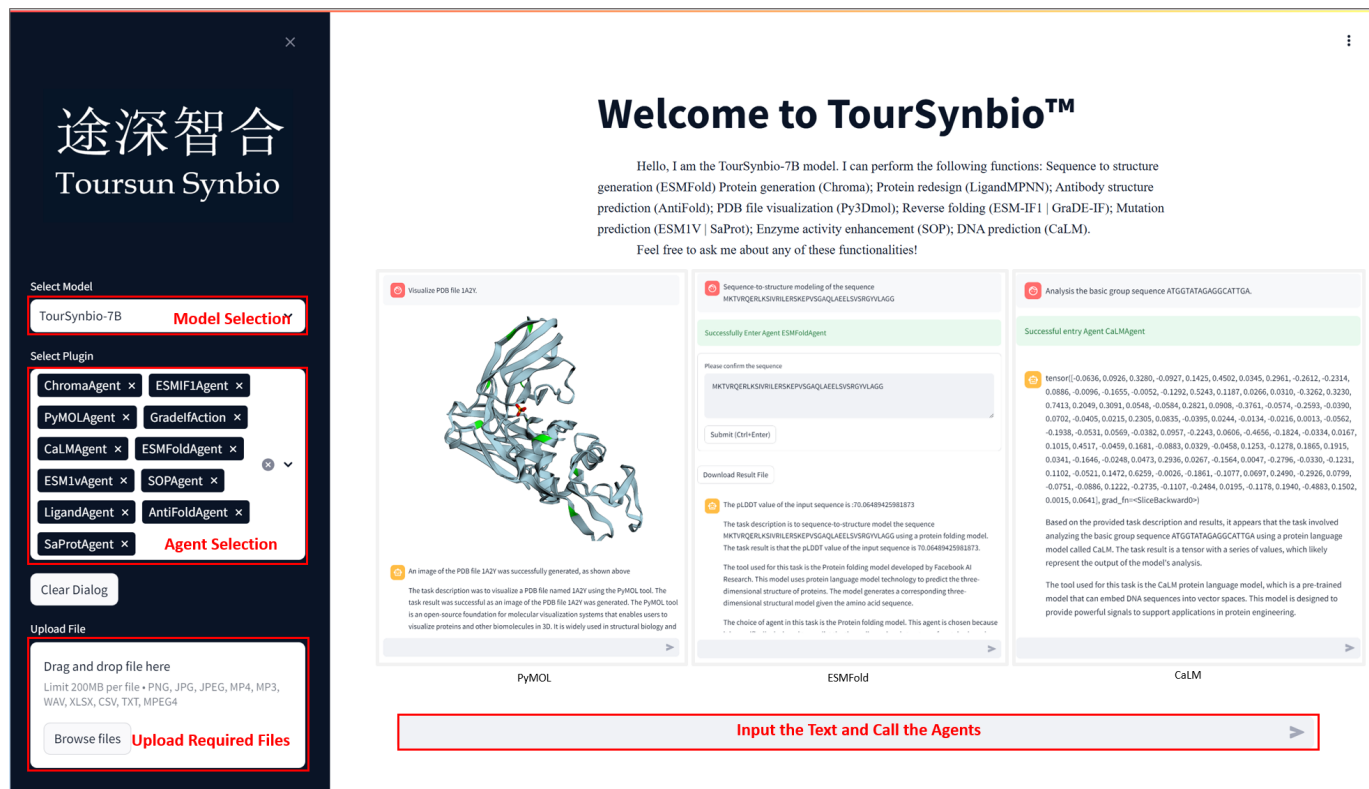


Fig. 3. The Human-centered conversational user interface of TourSynbio-Agent. It integrates four key components: (1) Model Selection, allowing users to choose between TourSynbio-7B and other language models; (2) Agent Selection, enabling customization of active protein engineering tools; (3) File Upload, supporting various file formats for data input; and (4) Text Input area for natural language interactions. The interface also showcases real-time visualizations and results from various protein engineering tasks, including the PyMOL, ESMFold [22], and CaLM [27] agents. This intuitive design facilitates seamless interaction between users and the AI-driven protein engineering models, enhancing workflow efficiency and accessibility for both experts and non-experts in the field. A video demo is available at [here on GitHub](#).

requests using everyday language without the need to learn complex command syntax of protein engineering. It can reduce the learning curve, making the interaction with protein engineering models accessible to users with diverse backgrounds and expertise levels.

- **Customizability.** It offers flexible interface options, allowing users to tailor the layout and functional modules to their specific needs and preferences. Users can customize the workflow by selecting specific agents, adjusting the interface theme, and modifying color schemes to enhance their comfort and productivity. This personalization capability enables users to create an optimal working environment that aligns with their individual work habits and preferences.

Our CUI includes three functional modules.

1) *Model Selection:* Our CUI offers users the flexibility to choose from a range of large language models, including TourSynbio-7B and GPT-4-turbo, to drive the agent framework. This feature provides several key advantages in the context of protein engineering tasks. Specifically, users can select the most appropriate model for different tasks, ensuring optimal processing and performance. For instance, TourSynbio-7B may be preferred for tasks requiring specialized protein sequence understanding, while GPT-4-turbo might be chosen

for more general language tasks. The interface is designed with extensibility in mind, accommodating the addition of other LLMs in the future.

2) *Agent Selection:* Our CUI also allows users to customize the set of agents activated for each task, enhancing the professionalism and accuracy of task execution. By selecting specific agents, users can ensure that the most suitable tools and functions are employed for their current protein engineering project. This flexibility enables users to fine-tune the system's behavior according to different scenarios, avoiding potential errors that might occur with automatic selection. Users can activate or deactivate multiple agents based on task requirements, optimizing the processing of each protein engineering task while enhancing their sense of control and trust in the system.

3) *File Upload:* The interactive interface supports file uploads to facilitate various agent tasks. Users can easily upload files through drag-and-drop or browsing functions, with support for multiple file types including PDB, FASTA, CSV, TXT. This feature enables users to seamlessly incorporate necessary data when handling complex protein engineering tasks, thereby improving work efficiency and accuracy in task completion.

III. EXPERIMENTS

A. Implementation Details

We used consistent hyperparameters across both phases. The learning rate was set to 0.0002, with a batch size of 16. We employed the AdamW optimizer and alternated between LinearLR and CosineAnnealingLR for parameter scheduling. The training method utilized LoRA with an α value of 16 and a dropout rate of 0.1. We trained the TourSynbio-7B on eight NVIDIA A100 GPUs, each with 80GB of memory.

B. Quantitative Evaluations

1) *ProteinLMBench and Evaluation Metric*: To assess the performance of TourSynbio-7B and compare it with other LLMs, we utilized ProteinLMBench [16]. ProteinLMBench consists of 944 manually verified multiple-choice questions, covering a wide range of protein-related topics including protein general knowledge, literature understanding, property prediction, function prediction, and protein design [16]. The evaluation metric for our experiments was accuracy, defined as the percentage of correctly answered questions on ProteinLMBench.

2) *Experimental Setups*: We compare with the TourSynbio-7B with various state-of-the-art LLMs in a zero-shot manner, including Falcon-7b [35], Qwen1.5-7B [36], Moonshot [37], Mistral-7B-Instruct-v0.2 [38], Baichuan2-7B-Chat [39], LLaMA-2-7B-Chat-hf [40], InternLM-Chat-20B, InternLM2-Chat-7B [15], ChatGLM3-6B [41], Yi-6B-Chat [42], GPT3.5-turbo, and GPT4-turbo [9]. For the ablation study, we compared TourSynbio-7B trained with instruction fine-tuning only (denoted as ‘w/o PT’) and the one with full settings, *i.e.*, trained with both instruction fine-tuning and post-training.

TABLE II
PERFORMANCE COMPARISON OF TOURSYNBIO-7B AGAINST STATE-OF-THE-ART LLMs ON PROTEINLMBENCH. CORRECT RATE INDICATES ACCURACY, AND INFERENCE TIME IS MEASURED IN MINUTES. TOURSYNBIO-7B (w/o PT) REFERS TO THE MODEL WITHOUT POST-TRAINING. **BOLD** INDICATES THE BEST PERFORMANCE.

Protein models score		
Model	Correct Rate (%)	Inference Time (min)
GPT4-turbo [9]	57.94	15.52
InternLM2-20B [15]	57.52	47.20
GPT3.5-turbo [9]	55.19	21.03
InternLM2-7B [15]	54.98	19.23
InternLM2-Chat-7B [15]	54.76	35.58
InternLM2-Chat-20B [15]	51.38	31.11
Yi-6B [42]	50.85	59.05
Mistral-7B-Instruct-v0.2 [38]	50.11	13.00
ChatGLM3-6B [41]	48.94	8.00
Baichuan2-7B [39]	44.49	16.37
InternLM-Chat-20B [15]	40.54	66.00
LLaMA2-7B [40]	39.64	64.00
Moonshot [37]	38.26	16.25
Qwen1.5-7B [36]	21.73	13.00
Falcon-7B-Instruct [35]	20.55	25.42
Falcon-7B [35]	19.17	15.55
TourSynbio-7B	62.18	22.34
TourSynbio-7B (w/o PT)	58.26	21.36

3) *Results*: Table II presents the performance of LLMs on the ProteinLMBench, comparing their accuracy scores and inference times. TourSynbio-7B demonstrates superior performance, achieving the highest accuracy of 62.18% among all models evaluated, including larger models and commercial APIs. Notably, TourSynbio-7B outperforms GPT4-turbo, by a margin of 4.24 percentage points (62.18% vs. 57.94%). This result is impressive given that GPT4-turbo is a much larger model with more extensive training data. The ablation study results show the effectiveness of our two-stage training approach. TourSynbio-7B with full settings (both post-training and instruction fine-tuning) achieves a 3.92 percentage point improvement over the version without post-training (62.18% vs. 58.26%). It underscores the importance of the post-training self-supervised learning phase in enhancing the model’s protein understanding capabilities. Among the 7B-parameter models, TourSynbio-7B significantly outperforms its counterparts. For instance, it surpasses InternLM2-7B by 7.2 percentage points (62.18% vs. 54.98%) and Mistral-7B-Instruct-v0.2 by 12.07 percentage points (62.18% vs. 50.11%). This demonstrates the effectiveness of our specialized training approach for protein-related tasks. Regarding inference time, TourSynbio-7B maintains competitive efficiency. Its inference time of 22.34 minutes on ProteinLMBench is comparable to or better than many other models, including some with lower accuracy scores. This indicates a good balance between performance and computational efficiency. It’s worth noting that TourSynbio-7B even outperforms larger models like InternLM2-20B (57.52%) and InternLM2-Chat-20B (51.38%), despite having fewer parameters. This suggests that our targeted training on protein-related data is more effective for protein understanding tasks than simply increasing model size.

C. Qualitative Comparison via Case Study

To illustrate the practical benefits of TourSynbio-Agent over traditional programming methods in protein engineering tasks, we conducted a case study comparing these two approaches for analyzing turnover numbers. Fig. 4 highlights this comparison, showcasing the key differences in user experience and workflow efficiency. The left panel of Fig. 4 depicts a conventional programming-based method, requiring users to write and execute Python code for data analysis. This method necessitates extensive programming expertise and familiarity with specific libraries such as PyTorch. Users must manually handle data loading, model initialization, and result processing, which can be time-consuming and error-prone, particularly for researchers without a strong programming background. In contrast, the right panel of Fig. 4 demonstrates TourSynbio-Agent’s approach, offering a more intuitive and user-friendly interface. Users interact with the system using natural language, exemplified by the query, “I want to predict the mutation of the sequence”. It abstracts the complexities of code implementation, enabling researchers to concentrate on their scientific inquiries rather than the technical details of programming. TourSynbio-Agent’s conversational interface democratizes access to advanced protein engineering tools, making

them accessible to a wider range of researchers, including those lacking extensive programming skills. Furthermore, TourSynbio-Agent adapts to various tasks without requiring users to write or modify code, enhancing research efficiency and reducing the likelihood of errors. While not explicitly shown in the figure, the interface likely includes options for downloading and visualizing results, further increasing its utility. This case study demonstrates how TourSynbio-Agent’s human-centered design significantly lowers the barrier to entry for complex protein engineering tasks, making sophisticated analyses more accessible and efficient.

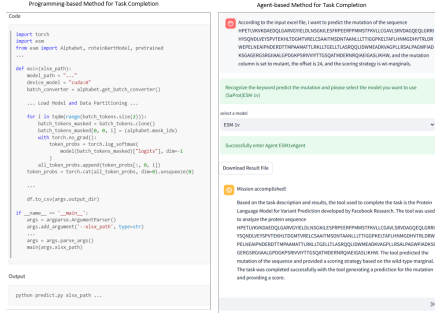


Fig. 4. Comparison between traditional programming-based and TourSynbio-Agent for protein engineering tasks. The left panel illustrates a code snippet for mutation prediction using a deep learning model, while the right panel showcases the TourSynbio-Agent’s user-friendly conversational interface for the same task, utilizing the ESM-1v model. This highlights the ease and efficiency of the TourSynbio-Agent, which abstracts complex code into intuitive interactions, thereby streamlining the protein engineering workflow.

D. Wet Lab Downstream Case Studies

Finally, to demonstrate the practical applicability of TourSynbio-7B and TourSynbio-Agent, we conducted two downstream case studies, namely vanilla key enzyme modification and steroid compound catalysis. These studies aim to showcase how our model can address real-world challenges.

1) *Vanilla Key Enzyme Modification*: Vanilla, renowned for its distinctive aroma, is important in various industries, including cosmetics, food, and fragrance [43]. Natural vanilla extract, priced at 1,000 RMB/kg, is significantly more expensive than its synthetic counterpart (200 RMB/kg), yet preferred for high-end products due to superior quality. Protein engineering offers a promising solution to this challenge through the synthesis of vanilla from biological waste. This process relies on two-step enzymatic catalysis, with the Aldehyde Deformylating Oxygenase (ADO) protein playing an important role [44]. The efficiency of ADO directly impacts the purity and yield of the produced vanilla, thereby influencing its market value. Traditional rational design approaches for ADO optimization have shown limited success. Hence, we employed TourSynbio-7B and TourSynbio-Agent to address these challenges, aiming to reduce dry lab delivery time, improve enzyme turnover number, and decrease production costs, as depicted in Fig. 5.

2) *Steroid Compound Catalysis*: Steroid compounds are important for maintaining cell membrane fluidity and stability. Our case study focuses on improving the selectivity of the

(a) TourSynbio™	
Dry lab delivery time	<1 day
Enzyme activity	Improve >4 times
Production cost	Lower 4 times

(b) No.	Mutant	Concentration
0	WT	45.8
1	E6W	7.7
2	E10W	173
3	I11G	58.5
4	S12D	7.5
5	Y14W	1.4
6	P73N	57.8

(c) No.	Protein Selectivity
WT	35.4
1	65.61
2	73.89
3	55.37
4	60.43
5	50.75
6	48.93

Fig. 5. Result on the vanilla key enzyme modification. (a) Summary of TourSynbio-Agent’s performance improvements in enzyme engineering: reduced dry lab delivery time to less than one day, enhanced enzyme activity by over four times, and lowered production costs by four times. (b) Table showing various mutants and their concentrations, highlighting E10W in red due to its significantly higher concentration. (c) Table displaying various protein mutants ranked by their selectivity, with the red mutant highlighted as the top performer.

P450 enzyme, which catalyzes steroid compounds. By utilizing TourSynbio-7B and TourSynbio-Agent, we aim to identify potential modifications to the P450 enzyme that could enhance its catalytic selectivity, potentially leading to more efficient and precise steroid synthesis processes.

IV. CONCLUSION

In this study, we introduced TourSynbio-7B, the first multi-modal large model specifically designed for protein engineering without relying on external protein encoders. TourSynbio-7B, leveraging the structural similarities between protein sequences and natural languages, demonstrated superior performance in protein understanding and engineering tasks, outperforming GPT-4 on the ProteinLMBench benchmark. We further developed TourSynbio-Agent, an agent framework integrating various deep learning models and tools, offering a unified conversational interface for the efficient execution of complex protein engineering tasks. Our framework’s capabilities were validated through wet lab case studies on enzyme modification and steroid compound catalysis, showcasing significant improvements in mutation accuracy, delivery time, and automation. Despite these advancements, TourSynbio-7B’s performance in predicting highly complex protein structures could be further improved, indicating the need for specialized post-training on diverse structural datasets. Future work will focus on integrating advanced structural prediction models to enhance the precision and applicability of TourSynbio-7B in more real-world protein engineering scenarios.

ACKNOWLEDGMENT

We gratefully acknowledge Karamay Carbon Neutrality Network Technology Co., Ltd for providing the GPU resources used in this research.

REFERENCES

- [1] M. Leisola and O. Turunen, "Protein engineering: opportunities and challenges," *Applied Microbiology and Biotechnology*, vol. 75, no. 6, pp. 1225–1232, 2007.
- [2] D. Listov, C. A. Goverde, B. E. Correia, and S. J. Fleishman, "Opportunities and challenges in design and optimization of protein function," *Nature Reviews Molecular Cell Biology*, pp. 1–15, 2024.
- [3] M. Defresne, S. Barbe, and T. Schiex, "Protein design with deep learning," *International Journal of Molecular Sciences*, vol. 22, no. 21, p. 11741, 2021.
- [4] J. Dauparas *et al.*, "Robust deep learning-based protein sequence design using proteinmpnn," *Science*, vol. 378, no. 6615, pp. 49–56, 2022.
- [5] J. Wang, H. Cao, J. Z. Zhang, and Y. Qi, "Computational protein design with deep learning neural networks," *Scientific reports*, vol. 8, no. 1, pp. 1–9, 2018.
- [6] A. Bairoch and R. Apweiler, "The swiss-prot protein sequence database and its supplement trembl in 2000," *Nucleic acids research*, vol. 28, no. 1, pp. 45–48, 2000.
- [7] M. Heinzinger, A. Elnaggar, Y. Wang, C. Dallago, D. Nechaev, F. Matthes, and B. Rost, "Modeling aspects of the language of life through transfer-learning protein sequences," *BMC bioinformatics*, vol. 20, pp. 1–17, 2019.
- [8] A. Rives *et al.*, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *PNAS*, 2019. [Online]. Available: <https://www.biorxiv.org/content/10.1101/622803v4>
- [9] OpenAI *et al.*, "GPT-4 Technical Report," *arXiv preprint arXiv:2303.08774*, 2024.
- [10] M. Blum *et al.*, "The InterPro protein families and domains database: 20 years on," *Nucleic Acids Research*, vol. 49, no. D1, pp. D344–D354, 11 2020. [Online]. Available: <https://doi.org/10.1093/nar/gkaa977>
- [11] L. Lv, Z. Lin, H. Li, Y. Liu, J. Cui, C. Y.-C. Chen, L. Yuan, and Y. Tian, "Prollama: A protein large language model for multi-task protein language processing," *arXiv preprint arXiv:2402.16445*, 2024.
- [12] C. Wang, H. Fan, R. Quan, and Y. Yang, "Protchatgpt: Towards understanding proteins with large language models," *arXiv preprint arXiv:2402.09649*, 2024.
- [13] M. Xu, X. Yuan, S. Miret, and J. Tang, "ProtST: Multi-modality learning of protein sequences and biomedical texts," in *ICML*, vol. 202, 2023, pp. 38 749–38 767. [Online]. Available: <https://proceedings.mlr.press/v202/xu23t.html>
- [14] H.-Y. Zhou, Y. Fu, Z. Zhang, B. Cheng, and Y. Yu, "Protein representation learning via knowledge enhanced primary structure reasoning," in *ICLR*, 2023. [Online]. Available: <https://openreview.net/forum?id=VbCMhg7MRnj>
- [15] Z. Cai *et al.*, "Internlm2 technical report," *arXiv preprint arXiv:2403.17297*, 2024.
- [16] Y. Shen, Z. Chen, M. Mamalakis, L. He, H. Xia, T. Li, Y. Su, J. He, and Y. G. Wang, "A fine-tuning dataset and benchmark for large language models for protein understanding," 2024. [Online]. Available: <https://arxiv.org/abs/2406.05540>
- [17] S. K. Burley *et al.*, "RCSB Protein Data Bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences," *Nucleic Acids Research*, vol. 49, no. D1, pp. D437–D451, 11 2020. [Online]. Available: <https://doi.org/10.1093/nar/gkaa1038>
- [18] P. Liang *et al.*, "Holistic Evaluation of Language Models," *Transactions on Machine Learning Research*, 2023, featured Certification, Expert Certification. [Online]. Available: <https://openreview.net/forum?id=iO4LZibEqW>
- [19] L. Kinch, R. Schaeffer, A. Kryshchuk, and N. Grishin, "Target classification in the 14th round of the critical assessment of protein structure prediction (CASP14)," *Proteins: Structure, Function and Bioinformatics*, vol. 89, no. 12, pp. 1618–1632, Dec. 2021.
- [20] D. Hendrycks *et al.*, "Measuring massive multitask language understanding," in *ICLR*, 2021. [Online]. Available: <https://openreview.net/forum?id=d7KBjmI3GmQ>
- [21] S. Kapoor, B. Stroebel, Z. S. Siegel, N. Nadgir, and A. Narayanan, "Ai agents that matter," *arXiv preprint arXiv:2407.01502*, 2024.
- [22] Z. Lin, H. Akin *et al.*, "Language models of protein sequences at the scale of evolution enable accurate structure prediction," *BioRxiv*, 2022. [Online]. Available: <https://www.biorxiv.org/content/early/2022/07/21/2022.07.20.500902>
- [23] S. A. Hollingsworth and R. O. Dror, "Molecular dynamics simulation for all," *Neuron*, vol. 99, no. 6, pp. 1129–1143, 2018.
- [24] J. Su, C. Han *et al.*, "Saprot: Protein language modeling with structure-aware vocabulary," *BioRxiv*, 2023. [Online]. Available: <https://www.biorxiv.org/content/early/2023/10/02/2023.10.01.560349.1>
- [25] J. Meier, R. Rao *et al.*, "Language models enable zero-shot prediction of the effects of mutations on protein function," *BioRxiv*, 2021. [Online]. Available: <https://www.biorxiv.org/content/early/2021/07/10/2021.07.09.450648>
- [26] J. Smirnovienė, L. Baranauskienė, A. Zubrienė, and D. Matulis, "A standard operating procedure for an enzymatic activity inhibition assay," *European Biophysics Journal*, vol. 50, no. 3, pp. 345–352, 2021. [Online]. Available: <https://doi.org/10.1007/s00249-021-01530-8>
- [27] C. Outeiral and C. M. Deane, "Codon language embeddings provide strong signals for use in protein engineering," *Nature Machine Intelligence*, vol. 6, no. 2, pp. 170–179, 2024.
- [28] Y. Shen, O. Lv, H. Zhu, and Y. G. Wang, "Proteinengine: Empower llm with domain knowledge for protein engineering," 2024. [Online]. Available: <https://arxiv.org/abs/2405.06658>
- [29] C. Hsu, R. Verkuil, J. Liu, Z. Lin, B. Hie, T. Sercu, A. Lerer, and A. Rives, "Learning inverse folding from millions of predicted structures," *BioRxiv*, 2022. [Online]. Available: <https://www.biorxiv.org/content/early/2022/09/06/2022.04.10.487779>
- [30] K. Yi, B. Zhou, Y. Shen, P. Liò, and Y. Wang, "Graph denoising diffusion for inverse protein folding," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [31] M. H. Høie *et al.*, "Antifold: Improved antibody structure-based design using inverse folding," 2024. [Online]. Available: <https://arxiv.org/abs/2405.03370>
- [32] J. B. Ingraham *et al.*, "Illuminating protein space with a programmable generative model," *Nature*, vol. 623, no. 7989, pp. 1070–1078, 2023. [Online]. Available: <https://doi.org/10.1038/s41586-023-06728-8>
- [33] J. Dauparas *et al.*, "Atomic context-conditioned protein sequence design using ligandmpnn," *Biorxiv*, pp. 2023–12, 2023.
- [34] L. D. Team, "Lagent: InternLM a lightweight open-source framework that allows users to efficiently build large language model (llm)-based agents," <https://github.com/InternLM/lagent>, 2023.
- [35] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, Étienne Goffinet, D. Hesslow, J. Launay, Q. Malartic, D. Mazzotta, B. Noune, B. Pannier, and G. Penedo, "The Falcon Series of Open Language Models," *arXiv preprint arXiv:2311.16867*, 2023.
- [36] J. Bai *et al.*, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023.
- [37] D. J. Zhang *et al.*, "Moonshot: Towards controllable video generation and editing with multimodal conditions," *arXiv preprint arXiv:2401.01827*, 2024.
- [38] A. Q. Jiang *et al.*, "Mistral 7B," *arXiv preprint arXiv:2310.06825*, 2023.
- [39] Baichuan, "Baichuan 2: Open large-scale language models," *arXiv preprint arXiv:2309.10305*, 2023. [Online]. Available: <https://arxiv.org/abs/2309.10305>
- [40] H. Touvron *et al.*, "Llama 2: Open Foundation and Fine-Tuned Chat Models," *arXiv preprint arXiv:2307.09288*, 2023.
- [41] Z. Du, Y. Qian, X. Liu, M. Ding, J. Qiu, Z. Yang, and J. Tang, "GLM: General Language Model Pretraining with Autoregressive Blank Infilling," in *ACL (Volume 1: Long Papers)*, 2022, pp. 320–335.
- [42] 01.AI, A. Y. : *et al.*, "Yi: Open Foundation Models by 01.AI," *arXiv preprint arXiv:2403.04652*, 2024.
- [43] F. Abbas *et al.*, "Aroma components in horticultural crops: chemical diversity and usage of metabolic engineering for industrial applications," *Plants*, vol. 12, no. 9, p. 1748, 2023.
- [44] C. Andre *et al.*, "Fusing catalase to an alkane-producing enzyme maintains enzymatic activity by converting the inhibitory byproduct h2o2 to the cosubstrate o2," *Proceedings of the National Academy of Sciences*, vol. 110, no. 8, pp. 3191–3196, 2013.