UNIVERSITY OF CALIFORNIA SAN DIEGO

Advancing inverse folding models: exploring diverse optimization strategies

A Thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Bioengineering

by

Zinnia Ma

Committee in charge:

Professor Wei Wang, Chair
Professor Benjamin Lee Smarr, Co-Chair
Professor Shankar Subramaniam

2024

The Thesis of Zinnia Ma is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

Thanks to my family and my fiancé's.

Thanks also to my advisor Dr. Wei Wang and my co-advisor Dr. Benjamin Smarr for their guidance and support. Appreciation is extended to Dr. Shankar Subramaniam for agreeing to be a committee member and for offering valuable insights into this research.

ABSTRACT OF THE THESIS

Advancing inverse folding models: exploring diverse optimization strategies

by

Zinnia Ma

Master of Science in Bioengineering

University of California San Diego, 2024

Professor Wei Wang, Chair
Professor Benjamin Lee Smarr, Co-Chair

Inverse folding is an important task in protein engineering. This problem was once a significant challenge, but recent developments in the field of deep learning have led to the emergence of many effective models, exemplified by ProteinMPNN. However, some deficiencies remain in the field of inverse folding. On one hand, while the relationship between sequence and structure is many-to-many, most existing models focus solely on predicting the reference sequence corresponding to the original structure as the optimization objective during model training. On the other hand, the sequence recovery rate has

long been the primary and often the only evaluation metric used to evaluate inverse folding models. Based on this situation, this work primarily explores two aspects: firstly, we designed and compared some strategies to improve the performance of the inverse folding model by using a modular approach with information beyond the reference sequence itself during the training process. Secondly, we introduced a very comprehensive series of assessments of model performance, which allows for a detailed comparison of the strengths and weaknesses of the models, and focuses on the practical applications of the models.

Chapter 1

Introduction

1.1 What is inverse folding

Inverse folding plays a pivotal role in structural biology and protein design, with its main goal being the prediction of amino acid sequences that can fold into a given protein 3D structure. The significance of inverse folding mainly lies in two aspects: sequence diversification[1] and complementing the protein generative model[2].

The rationale for utilizing inverse folding models for sequence diversification is as follows: during the process of protein design, there are instances where a protein with desirable structural-based properties, such as high binding affinity to a specific target, is identified. However, other properties of the protein may not be satisfactory, and optimization is desired. In such circumstances, inverse folding models can generate a large set of sequences that not only preserve the desired structural-based properties but also potentially offer a wide distribution over other properties, thereby assisting in the identification

of ideal candidates[3].

The necessity of utilizing inverse folding models to complement the protein generative model arises from the current workflows in protein design, where most generative models predominantly produce only the backbone structure. Consequently, there is a requirement for inverse folding models to translate the generated structures into protein sequences, thereby completing the protein's information.

1.2 Existing problems

Among the existing inverse folding models, it is a common approach to use sequence recovery as both the learning objective and the main evaluation metric of the model. However, this approach does not perfectly align with the overarching goal of inverse folding. The primary aim is to generate amino acid sequences that can fold back into a given structure, not necessarily to recover the original sequence itself, especially in the context where it is known that the relationship between protein sequences and structures is many-to-many. It is crucial to acknowledge that sequence similarity does not guarantee structural similarity, and vice versa. This understanding leads to the conclusion that an inverse folding model with high sequence recovery performance is not automatically a truly effective "inverse folding" model. From another perspective, if the objective is to use inverse folding models for sequence diversification, we might prefer a model capable of sampling sequences as distinct as possible from the original sequence, aiming for minimal sequence identity. In such a scenario, using the reproduction of the original sequence as the sole optimization

goal without considering other factors, is highly contradictory with our goal.

In this study, we emphasized the important of using the refoldability metrics, and exploring different methods to improve the optimization process correspondingly. Our contributions are as follows:

- We propose several straightforward and efficient strategies to improve the performance of inverse folding models. These enhancements are designed to be modular, leaving the core architecture of the models unchanged, and can be integrated into a wide range of existing inverse folding models. These optimization techniques honor prior knowledge, demonstrating pronounced benefits particularly at higher sampling temperatures.

- We adopt a comprehensive approach to model evaluation, considering the model's performance across various dimensions, including recovery, perplexity, diversity, and refoldability. Additionally, we take into account the changes in model performance as a function of sampling temperature.

- To showcase the practicality of the enhanced models, we utilized them to develop a dataset wherein each pair of sequences and their corresponding structures are characterized by high structural concordance and low sequence similarity.

Chapter 2

Related Works

2.1 Traditional Methods

Computational Protein Design (CPD) aims to create proteins that fulfill specific functional requirements as tools or therapeutics, which is crucial in the fields of biology and medicine. Predicting the sequence of a given protein structure is a significant challenge in CPD, traditionally known as protein fixed-backbone sequence design and currently also referred to as inverse folding[4]. Historically, this method has been dominated by energy-based approaches, notably exemplified by Rosetta, which utilizes a single parametric energy function to capture the sequence-structure relationship[5].

2.2 Deep learning models

The field of inverse folding has advanced notably with deep learning methods since 2019. StructTrans[6] employs graph-based self-attention modules in its framework, enhancing the efficiency of capturing complex dependencies in proteins and achieving performance far surpassing that of Rosetta. Subsequently, GVP[7] introduced a unifying network to perform both geometric and relational reasoning together, outperforming StructTrans in terms of both perplexity and recovery. These innovations have significantly improved protein design capabilities.

The field of inverse folding has seen rapid progress, benefiting from breakthroughs in deep learning models and protein design. This advancement has been significantly influenced by some impactful models, including well-known examples such as MIF[8], ESM-IF[9], and ProteinMPNN[3].

MIF and MIF-ST stand for the Masked Inverse Folding model and the Masked Inverse Folding with Sequence Transfer model, respectively[8]. As its name suggests, MIF is trained as a Masked Language Model, using the BERT corruption scheme to train the model to reconstruct the original amino acids based on the corrupted sequence and backbone structure. MIF-ST, on the other hand, enhances the model's capability by utilizing outputs from a pretrained sequence-only protein masked language model called CARP (the Convolutional Autoencoding Representations of Proteins protein masked language model) as input to the inverse folding model.

ESM-IF[9], recognized as one of the leading inverse folding models, achieves its success

by enhancing inverse folding with predicted structures. The authors predicted structures for 12 million UniRef50 protein sequences using AlphaFold2, addressing the quantity limitations of the CATH dataset and significantly enhancing the model's performance. Specifically, the authors experimented with three model architectures in this work: GVPGNN, GVP-GNN-large, and GVP-Transformer. Among these, GVP-Transformer performed the best, however, it also had the highest total number of parameters, reaching 142 million, with a total training time of 653 GPU days.

ProteinMPNN[3], developed by the Baker lab, is currently the most renowned and widely applied inverse folding model. Its strengths are evident in multiple aspects. Firstly, the network is relatively small and easy to train, with a total parameter count of approximately 1.7 million. Secondly, the network incorporates several ingenious design elements, such as random decoding, symmetry awareness, multi-chain awareness, etc. Among these elements, random decoding is most closely related to this work. Unlike the conventional left-to-right sequential decoding, ProteinMPNN decodes in a random order, allowing for greater flexibility in the decoding process. This can be particularly useful in specific tasks, such as when certain parts of a sequence are fixed and the design of other parts needs to be conditioned on the known sections. For ProteinMPNN, which employs random decoding, this is straightforward to achieve. However, it is also important to note that this approach increases the randomness of the generated sequences. Without constraints or modifications, ProteinMPNN can predict a variety of different sequences from a single reference protein structure, even when using the argmax function during sampling.

In recent developments following the previously mentioned models, some new inverse

folding models have emerged. Notable among them are PiFold[10] and KWDesign[11]. These models have claimed to significantly outperform ProteinMPNN in terms of sequence recovery. However, in current protein design literature, for various reasons, researchers are still predominantly using ProteinMPNN rather than these newer models.

Additionally, besides developing different inverse folding models for protein datasets, some researchers are focusing on creating inverse folding models for more specific tasks, such as antibody design. Existing antibody inverse folding models include AbMPNN and AntiFold[12]. Both are fine-tuned from existing protein inverse folding models, with AbMPNN being based on ProteinMPNN and AntiFold being based on the ESM-IF model.

2.3 Benchmark works

In the field of inverse folding, there are two relatively new benchmark studies, namely ProteinInvBench[13] and PDBStruct[14].

ProteinInvBench[13] emphasizes its comprehensive evaluation metrics to assess existing inverse folding models. However, we believe the article's shortcoming is that, although it mentions different sampling temperatures, it does not conduct in-depth evaluation based on them. In ProteinInvBench, the authors only compare the curves of recovery and diversity for each model with varying sampling temperatures. However, we know that different models may have different sensitivities to temperature changes, so simply relying on these curves does not provide us with much useful information. Therefore, in this work, we emphasize the use of different sampling temperatures and propose some new, more meaningful

evaluation methods that were not present in previous work.

In PDBStruct[14], the authors also mention some shortcomings in the existing metrics for evaluating inverse folding models, so they also place great emphasis on refoldability metrics, using TM and pLDDT to compare the refoldability of different models in the article. The authors compared various models, including StructTrans, GVP, ProteinMPNN, PiFold, etc., among which ProteinMPNN performed the best. This result also shows that some newer models, such as PiFold, can achieve high values in sequence recovery but do not necessarily improve the fundamental goal of inverse folding. Given the state-of-the-art performance of ProteinMPNN, this study will employ it as the base model.

Chapter 3

Methods

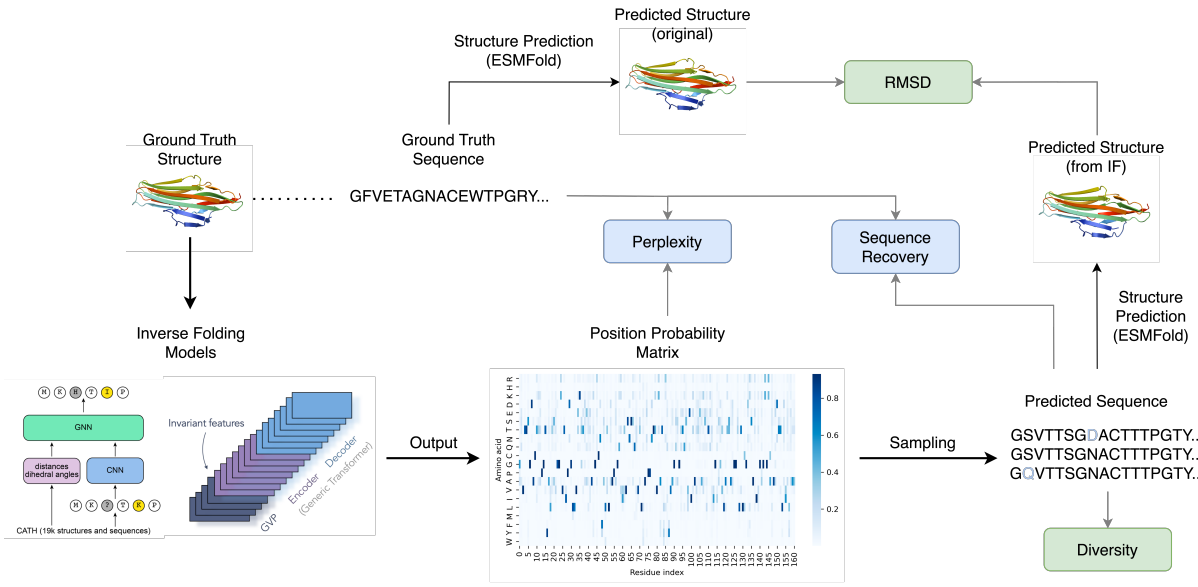

Figure 3.1: Workflow for testing inverse folding models. The ground truth structures are used as inputs to the model, with all amino acids masked so that the model does not see the reference sequence. The inverse folding models' final output is a position probability matrix, and the predicted sequences are sampled from this matrix. Based on the reference sequence and predicted sequence, we can use different metrics to evaluate the models.

To enhance the current inverse folding models, we have experimented with various

optimization techniques, primarily focusing on three different strategies: leveraging the substitution matrix, utilizing similarity between amino acid pairs, and incorporating MSA (Multiple Sequence Alignment) information.

The primary reason for employing these three methods is to explore how to set and optimize the training objectives of inverse folding models rationally. If our objective is to predict sequences that can fold into a given structure, rather than sequences identical to the reference sequence, using NLL loss for model optimization, as traditionally done, may not be completely reasonable. There are primarily two reasons for this:

Firstly, from the perspective of the reference sequence, biologically, the impact of substituting an amino acid at a certain position can vary. Some substitutions are common in evolutionary terms and have minimal impact on the structure, making such "incorrect predictions" relatively acceptable. However, other substitutions can critically affect the structure, potentially leading to disastrous outcomes for structural recovery. Therefore, it's clear that not all "incorrect predictions" should be treated equally, yet NLL Loss does not differentiate between these errors.

Secondly, given the fact that sequence similarity does not necessarily imply structural similarity, optimizing based solely on the reference sequence is also not rational. Ideally, the loss calculation for each iteration would be based on the structural recovery between the structures folded from the reference and predicted sequences. However, such a comprehensive model is unlikely to be feasible. Thus, we adopted the simplest approach: while direct structural prediction is not possible, we can utilize MSA information, allowing the inverse folding model to gain deeper insights into the structure.

3.1 Optimization strategies

3.1.1 Substitution matrix

An amino acid substitution matrix is a 20x20 grid quantifying how frequently each of the twenty standard amino acids in proteins is replaced by another over time[15]. In this work, we chose the BLOSUM62 matrix as our representative substitution matrix for sequence alignment and analysis. This decision was driven by BLOSUM62's widespread acceptance and utilization within the field of bioinformatics for its effectiveness in scoring alignments between evolutionary divergent protein sequences. In the BLOSUM62 matrix, each value is calculated using the equation[16]:

$$S_{ij} = 2\log_2 \frac{p_{ij}}{q_i q_j}$$

Where p_{ij} is the probability of two amino acids i and j replacing each other in a homologous sequence, and q_i and q_j are the respective probabilities of encountering the amino acids i and j in any protein sequence. λ is a scaling factor, adjusted to ensure the matrix contains integer values.

Therefore, the inferred equation for the ratio $\frac{p_{ij}}{q_i q_j}$ would be:

$$\frac{p_{ij}}{q_i q_j} = 2^{(S_{ij}/2)}$$

Based on the ratio $\frac{p_{ij}}{q_i q_j}$ derived, we can calculate the probability distribution of each original amino acid being replaced by other amino acids, as shown in the Figure 3.2. Consequently, an original sequence of length L will be converted into a position probability matrix with dimensions $L \times 21$. Subsequently, during the optimization process of the inverse folding

model, we can directly compare the model's output with this matrix, employing KL divergence loss to enhance their similarity. This approach enables the model to indirectly understand which substitutions are more conservative and less likely to occur.

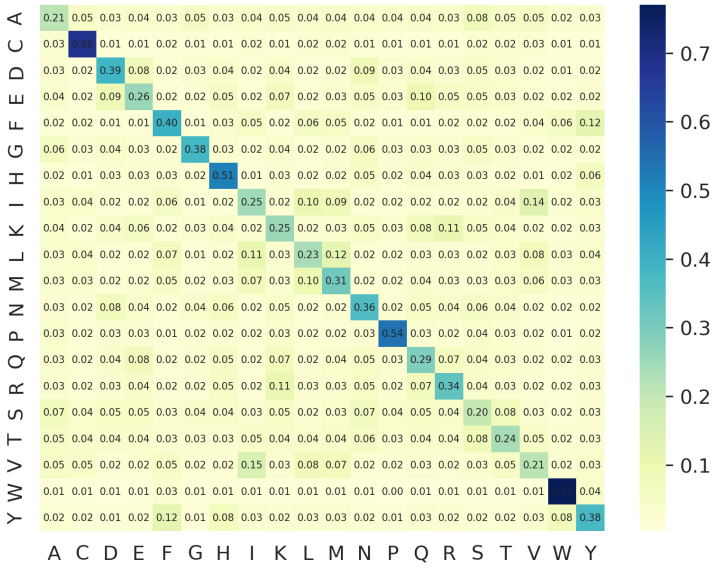

Figure 3.2: The probability distribution matrix calculated from the BLOSUM62 matrix.

3.1.2 Similarity

The reason we wish to utilize the similarity between residues is to enable the model during training to apply a lower penalty for "incorrect predictions" that involve similar residue substitutions, and a higher penalty for substitutions involving dissimilar residues, thus teaching the model to treat them differently. In order to let the model to reward correct predictions while penalizing incorrect ones with varying weights, we use the Mean Squared Error (MSE) loss as the loss function.

To quantitatively represent the similarity between two amino acid residues, we have

utilized existing large-scale language models. In the original ESM paper[17], the authors project the weight matrix of the network's final embedding layer into two dimensions using t-distributed stochastic neighbor embedding (t-SNE) and discover that the network has learned to encode physicochemical properties into its representations. Therefore, in this work, we use the weight matrix of the final embedding layer of the ESM2 model as the vector representation for each amino acid and calculate the similarity based on it. Figure 3.3 is a t-SNE visualization of these representations, presented in the same format as in the original ESM paper.

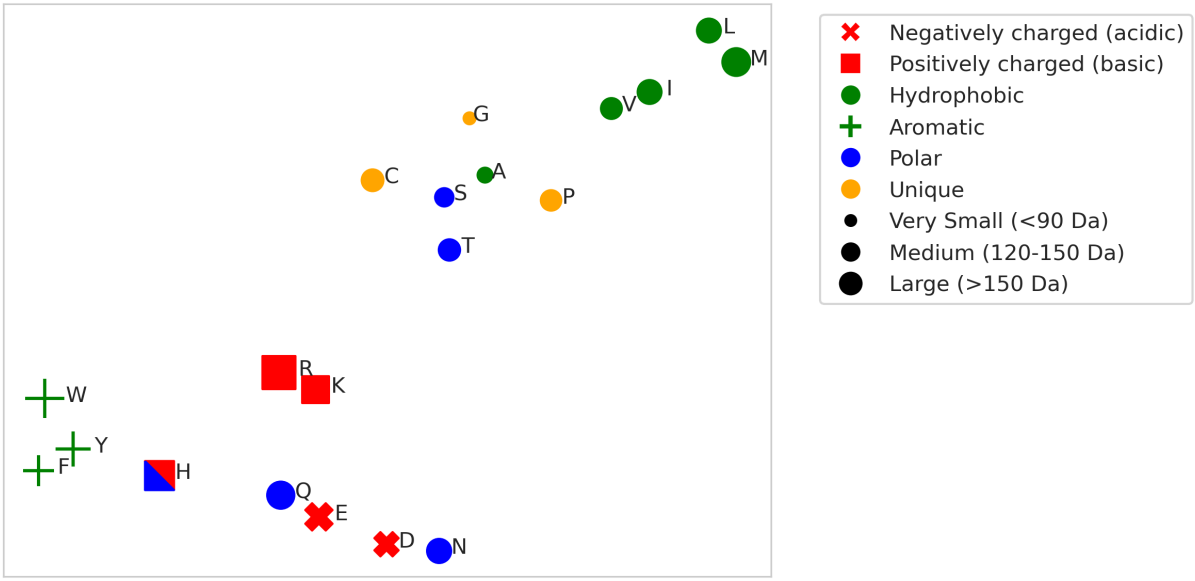

Figure 3.3: Biochemical properties of amino acids are represented in the Transformer model's output embeddings, visualized here with t-SNE.

Regarding the calculation of similarity between each pair of amino acids, we have adopted two distinctive and representative methods: cosine similarity and Euclidean distance. From these methods, we derive two separate 20x20 weight matrices that correspond to the degree of penalty to be applied for incorrect predictions between different

residues—one matrix for each calculation method. The specific steps from the representation to obtaining each weight matrix are as follows.

4 steps to obtain the cosine similarity weight matrix

Step1 - Calculate Cosine Similarity:

Calculate the cosine similarity S_{ij} between each pair of vectors i and j:

$$S_{ij} = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\|\|\mathbf{v}_j\|}$$

Step2 - Min-Max Normalize the Similarity Matrix:

Normalize the similarity matrix S to get S'_{ij}:

$$S'_{ij} = \frac{S_{ij} - S_{\min}}{S_{\max} - S_{\min}}$$

Step3 - Convert to Penalty Matrix:

Transform S' into a penalty matrix P:

$$P_{ij} = \begin{cases} 1 - S'_{ij} & \text{for } i \neq j \\ S'_{ij} & \text{for } i = j \end{cases}$$

Step4 - Exponentially Amplify Penalty Values:

Finally, apply the exponential function to the penalty matrix P to get the final penalty matrix P':

$$P'_{ij} = 10^{P_{ij}}$$

This step significantly increases the penalties, especially for higher penalty values, making the distinction between similarities more pronounced.

4 steps to obtain the Euclidean distance weight matrix

Step1 - Calculate Euclidean Distance:

Calculate the Euclidean distance D_{ij} between each pair of vectors i and j:

$$D_{ij} = \sqrt{\sum_{k} (v_{ik} - v_{jk})^2}$$

Step2 - Min-Max Normalize the Distance Matrix:

Normalize the distance matrix D to get D'_{ij}:

$$D'_{ij} = \frac{D_{ij} - D_{\min}}{D_{\max} - D_{\min}}$$

Step3 - Convert to Penalty Matrix:

Replace the diagonal of D' with 1s to form the penalty matrix P:

$$P_{ij} = \begin{cases} D'_{ij} & \text{for } i \neq j \\ 1 & \text{for } i = j \end{cases}$$

Step4 - Exponentially Amplify Penalty Values:

Finally, apply the exponential function to the penalty matrix P to get the final penalty matrix P':

$$P'_{ij} = 10^{P_{ij}}$$

This step significantly increases the penalties, especially for higher penalty values, making the distinction between distances more pronounced.

After obtaining these weight matrices, we applied MSE loss to effectively utilize them by assigning appropriate weights to each correct or incorrect substitution. The original

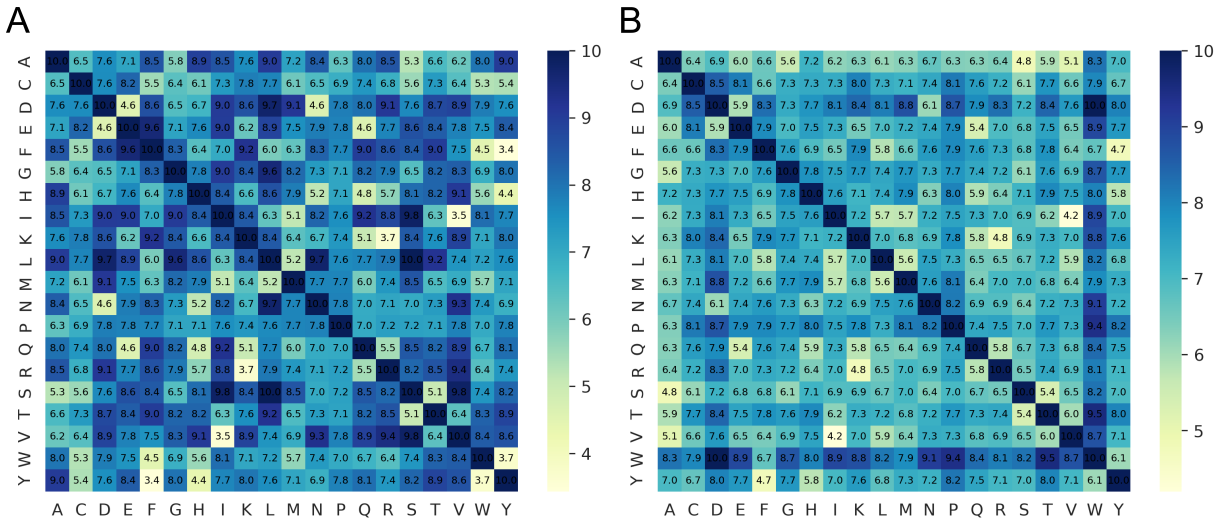

Figure 3.4: The two different weight matrices calculated from ESM2 model. A:Cosine similarity weight matrix, B: Euclidean distance weight matrix.

ProteinMPNN model uses NLL loss, which focuses solely on the correct class during optimization, whereas MSE loss considers the differences across all classes based on their expected probabilities. We tested the effect of replacing NLL loss with MSE loss without altering any other parts of the model or the soft label settings. The results demonstrated that this optimization approach did not significantly impact the model's performance. Therefore, we consider it reasonable to use MSE loss and apply our weight matrix to train the model, thereby assigning smaller penalties for substitutions between similar amino acids.

3.1.3 MSA information

For MSA information, we currently utilize it in a straightforward manner. After obtaining the sequences aligned to a given structure, we directly compute a position probability matrix from these sequences. This process is easy to implement because all the aligned

sequences have the same length. Subsequently, we use the KL divergence loss to optimize the model during the training process, with the objective of making the model's output position probability matrix closely approximate the matrix derived from the MSA for each sample.

3.2 Training

Since the improvements are modular, all basic settings employed in this study are identical to those described in the original ProteinMPNN paper. We trained each model for 150 epochs, incorporating 0.02 Gaussian noise during the training process and utilizing a single A6000 GPU. Each model was retrained three times to ensure consistency. Although not every model achieved full convergence by the 150th epoch, and it is probable that the performance at this checkpoint is not the optimal one observed across all epochs, we selected the 150th epoch as the checkpoint for each model in subsequent evaluations to maintain control over other variables. As a result, the performance metrics observed in the evaluations may not fully represent the maximum potential of the models.

3.3 Evaluation metrics

3.3.1 Sequence recovery

Sequence recovery is the primary metric used to measure the ability of a designed protein to recover its reference residues. Recovery is calculated by dividing the number of

identical residues between the predicted sequence and the reference sequence by the total sequence length.

3.3.2 Perplexity

For a sequence $s = s_1, s_2, \ldots, s_L$ where s_i is the character at position i and L is the length of the sequence, the perplexity of the sequence is defined as the exponential of the negative average (normalized) log probability of the sequence. It is given by:

$$\text{Perplexity}(s) = \exp\left(-\frac{1}{L}\sum_{i=1}^{L}\log P(s_i|\text{position } i)\right)$$

3.3.3 Diversity

Generating diverse sequences is crucial for exploring the reasonable protein sequence space.

$$D_{ij} = \frac{\sum_{l=1}^{n} \mathbb{1}_{r_{i,l} \neq r_{j,l}}}{n}$$
$$\text{Div} = \sum_{i=1}^{m-1}\sum_{j=i+1}^{m} \frac{D_{i,j}}{\frac{m(m-1)}{2}}$$

where $i, j \in \{1, 2, 3, \ldots, m\}$ and m is the number of totally designed sequences. The divisor $\frac{m(m-1)}{2}$ reflects the number of sequence pairs when $j > i$.

3.3.4 Refoldability

The term 'refoldability' refers to the ability of a generated sequence to fold into the reference structure. To calculate refoldability, we use ESMFold[18] to fold both the reference

and the predicted sequences into 3D structures, and then calculate the Ca-RMSD value between these two structures. To minimize errors due to the folding algorithm, we first compare the crystal structures with the predicted structures of the reference sequences, and exclude the sample if the RMSD between the pair of structures is too large or if the pLDDT score is too low.

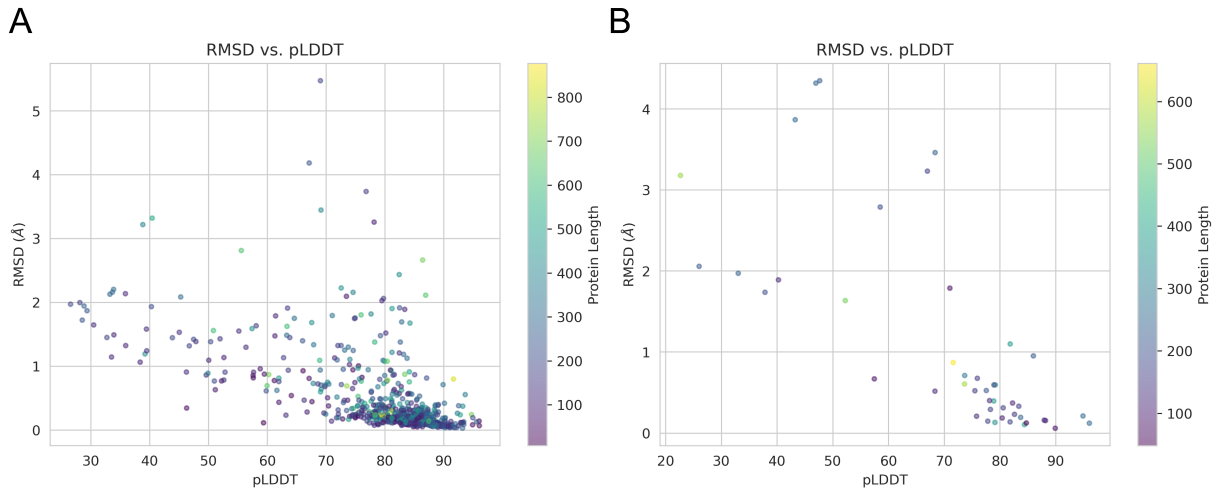

Figure 3.5: The scatter plot provides a visualization of the relationship between RMSD and pLDDT when comparing the crystal structures with the predicted structures: subfig A corresponds to the CAMEO dataset, and subfig B to the TS45 dataset. Each point's color denotes the sequence length, with exact values indicated on the color bar. There is a general negative correlation observed between RMSD and pLDDT. Concurrently, the sequence length appears to have minimal correlation with these two metrics, and sequences within the selected threshold retain a length distribution that is proportionate to the original dataset.

3.4 Dataset

3.4.1 PDBwithMSA

The OpenProteinSet[19] is an open-source corpus containing over 16 million MSAs, associated structural homologs from both the Protein Data Bank (PDB) and AlphaFold2 protein structure predictions. The original dataset used in the ProteinMPNN paper was curated from the PDB database and includes 676,452 single chains. For each chain, we search the OpenProteinSet to extract the corresponding MSA, merging this MSA information with the existing base data to form a new training dataset. We refer to this dataset as PDBwithMSA, which includes 645,363 single chains.

3.4.2 CAMEO

We downloaded protein 3D structures spanning one year (Oct 8, 2022 to Sep 30, 2023) from the Continuous Automated Model Evaluation (CAMEO) website, which we refer to as the CAMEO dataset throughout the paper. This CAMEO dataset originally contains 723 protein structures in total. However, to ensure the accuracy of the evaluation, we compared the crystal structures with the predicted structures of the reference sequences and retained only the proteins with RMSD < 0.5 and pLDDT > 75, resulting in a total of 515 proteins.

3.4.3 TS45

TS45 refers to the publicly available TS-domain structures from CASP15, consisting of 45 structures. In ProteinInvBench[13], the authors mention that this dataset can represent de-novo structures to some extent because the proteins in TS45 are significantly different from those in the training set, with low structural or sequence similarities.

Chapter 4

Experiments

4.1 Existing evaluation metrics

Table 4.1: Model Performance Comparison

Model	Recovery	Perplexity	RMSD
ProteinMPNN	**51.87%**	5.449	0.2245Å
KL_B62	49.14%	8.278	**0.2063Å**
MSE_ori	50.86%	**5.433**	0.2197Å
MSE_cos	50.88%	6.156	0.2142Å
MSE_ed	50.95%	6.126	0.2168Å
KL_MSA	33.98%	10.34	0.3755Å

In the evaluation of the models, we initially perform assessments on the CAMEO dataset, using the argmax function to directly obtain sequences for evaluation purposes. Given that the decoding process of ProteinMPNN is inherently random, to minimize the impact of this stochasticity, we generate 10 sequences for each structure using each model. As illustrated in Table 1, ProteinMPNN exhibits superior performance in terms of sequence

recovery, while the MSE_ori model has the lowest perplexity, and the KL_B62 model reaches the lowest RMSD. Notably, with the exception of the KL_MSA model, the other four models attain lower RMSD values than the original ProteinMPNN model, suggesting that the enhancements have yielded measurable improvements. Additionally, to confirm the significance of this improvement, we performed a paired t-test comparing the average RMSD values for ProteinMPNN and KL_B62 across these 515 proteins, obtaining a P-value < 0.05. This result suggests that the observed differences are statistically meaningful.

4.2 New evaluation related to model performance concerning sampling temperature

Relying solely on the argmax function for model evaluation is inadequate, as in real-world applications, it's common to employ various temperatures during the sampling process. To achieve a more thorough evaluation of the models and include the effects of sampling temperature on their performance, we analyzed the recovery, diversity, and RMSD of sequences produced by each model across a range of temperatures. The outcomes were depicted graphically in curve format.

Actually, in ProteinInvBench, the authors also compared the performance of models across various temperatures. However, they only specifically examining how the performance of different models varied with the sampling temperature for each metric. We conducted a similar analysis, and the results are presented in Figure B.1. From this figure, we can observe that as the sampling temperature increases, sequence recovery decreases,

perplexity remains largely stable, diversity rises, and RMSD also increases. A notable distinction is that, compared to other models, the KL_B62 and KL_MSA models exhibit significantly steeper slopes in sequence recovery, diversity, and RMSD as temperature changes, indicating greater sensitivity to variations in sampling temperature. Beyond this observation, it is difficult to extract additional meaningful information.

Given the known issue that different models have varying sensitivities to sampling temperature, it is challenging to derive useful evaluation insights from changes in a single metric. Inherent trade-offs in these metrics' performance include three closely linked yet distinct aspects: sequence recovery, diversity, and structural refoldability. Firstly, achieving higher diversity in sampled sequences often necessitates lower sequence recovery relative to the reference sequence, with numerous mutations influencing the ability of the generated sequences to fold into the reference structure. While these trade-offs—refoldability versus diversity and refoldability versus sequence recovery—are interconnected, they are not exactly the same. Each trade-off underlines a critical balance in experimental design: maximizing sequence diversity or minimizing sequence recovery without compromising the structural integrity required for accurate folding. Therefore, considering this alongside practical application needs, we have chosen to plot RMSD vs. Diversity and RMSD vs. Recovery curves to compare the performance of different models.

From Figure B.2, we can see that compared to the ProteinMPNN model, the other four models all show varying degrees of performance improvement. First, looking at Figure A, the RMSD vs. Diversity curves, considering that the goal is to maximize diversity while minimizing RMSD, a curve located more towards the lower right of the figure indicates

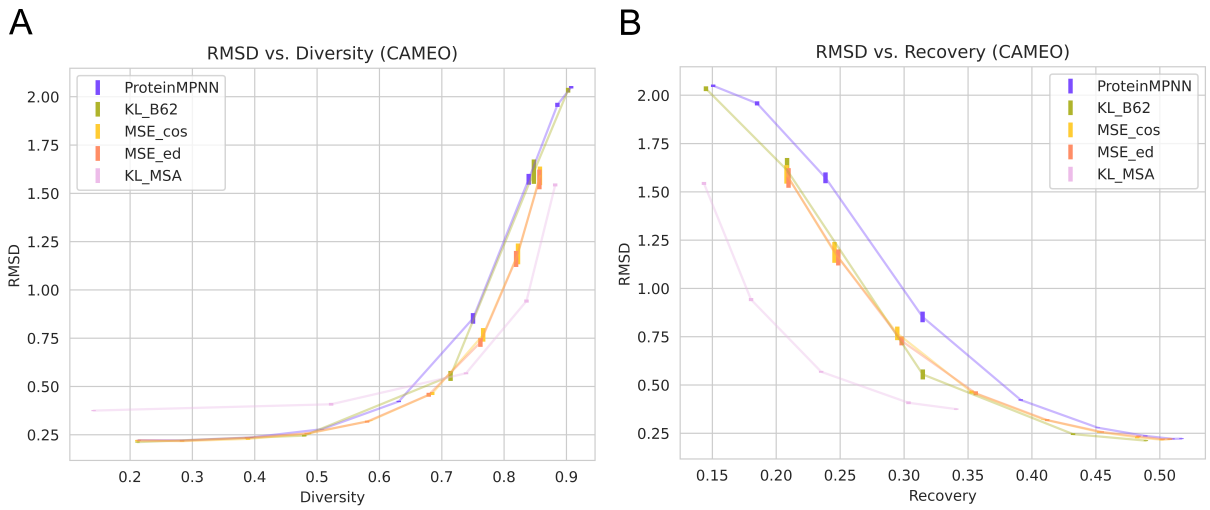

Figure 4.1: The curves on CAMEO dataset. A: RMSD vs. Diversity curve, B: RMSD vs. Sequence recovery curve.

better performance. Overall, the curves for the KL_B62, MSE_cos, and MSE_ed models are all positioned to the lower right of the ProteinMPNN, indicating that their improvements in this task are successful. The performance of KL_MSA is quite unique: within the range where diversity is below 0.65, it has the worst RMSD performance, but in the higher diversity range, it shows the most significant improvement. For instance, when generating sequences with a diversity of around 0.85, the RMSD for sequences generated by other models are about 1.6 Å, while for MSA it is lower than 1 Å.

This performance is understandable. By combining Figure B, we can see that due to the unique use of MSA information, when sampling is done with very low temperature, meaning diversity is still in a low range, its recovery already falls below 35%. Therefore, under low recovery, it is reasonable for RMSD performance to be worse. In Figure B, the goal is to minimize recovery while minimizing RMSD, so a curve located more towards the lower left indicates better performance. It can be seen that the KL_MSA model's curve

is much better than the others, while the curves for the KL_B62, MSE_cos, and MSE_ed models are positioned between KL_MSA and the ProteinMPNN model. This demonstrates that using MSA information is beneficial for sequence diversification, while the BLOSUM strategy and the similarity strategy are also effective.

When conducting this evaluation on the TS45 dataset, the results are shown in Figure B.2. It can be observed that the outcomes on the TS45 dataset are similar to those obtained using the CAMEO dataset. This indicates that even in a de-novo dataset, models other than ProteinMPNN still achieve better RMSD vs. Diversity curves and RMSD vs. Recovery curves. This demonstrates that the strategies we adopted continue to improve model performance in this task.

4.3 Sequence space expansion

Considering our previous discussion, the two primary implications of the inverse folding model currently are sequence diversification and complementing the protein generative mode, with our improvements mainly targeting the former. However, since differences in the curve alone are not sufficiently intuitive, we have designed the following task to more effectively assess the model's performance in practical applications.

Following the established settings for plotting the curve, we use a criterion to select novel sequences from those generated by each model and compare their success rates. The model with the higher success rate demonstrates a stronger ability for sequence diversification. To control variables, we choose to have each model generate the same number of

protein sequences, so we can directly compare the number of novel sequences that meet the criteria.

For this task, we continue to use the CAMEO dataset, which contains 515 protein structures. At each sampling temperature for every model, we sample 10 sequences for each reference structure, amounting to $515 \times 10 = 5,150$ in total. To select novel proteins from the generated sequences that fulfill our final goal, we designed the following criteria:

1. Firstly, the generated sequence should be able to fold into a structure in a relatively accurate way: pLDDT > 75;

2. Secondly, there should be a high similarity between the reference structure and the predicted structure, ensuring that the "inverse folding" is successful: RMSD < 0.5;

3. Lastly and most importantly, the sequence space of the reference structure should be successfully expanded, given the low identity between the reference sequence and the generated sequence: Sequence identity $< 20\%$.

In practice, considering that different models have different sensitivity to variations in sampling temperature, we need to select specific sampling temperatures for each model. To achieve this, we performed the selection process for each model at every temperature. The detailed results are shown in Appendix B.2, and we use the results corresponding to the best-performing temperature for each model to represent the model's capability in this task. The final results are presented in Table 4.2.

We can observe that, according to the number of selected sequences that meet the criteria, the KL_MSA model's ability to expand the sequence space is significantly superior to other models. Since each reference protein generated multiple sequences, we also

Table 4.2: Model Performance Comparison

Model	total_num	reference_num	average_length
ProteinMPNN	104	21	61.07
KL_B62	113	23	67.82
MSE_cos	115	22	65.38
MSE_ed	120	23	64.40
KL_MSA	**163**	**40**	**81.59**

counted the number of reference proteins for which each model successfully expanded the sequence space. The results show that KL_MSA expanded the sequence space for a far greater number of reference proteins than all other models, achieving nearly twice the number of successful expansions on this metric. Moreover, the average length of the selected proteins sampled by KL_MSA is the longest, reaching 81.59 amnio acids per sequence. Concurrently, KL_B62 and the two similarity-based models, MSE_cos and MSE_ed, also demonstrate strong performance than the ProteinMPNN model on selected proteins' number and average length.

To better observe and analyze the distribution of protein lengths generated by different models that meet the criteria, we visualized them in Figure 4.2. It is evident that the MSA_KL model samples novel proteins with the widest range of length distributions, spanning from 30 to 400. In contrast, other models can handle lengths only up to 200, and ProteinMPNN can manage lengths only less than 150. This fully demonstrates the advantages of training models based on MSA information.

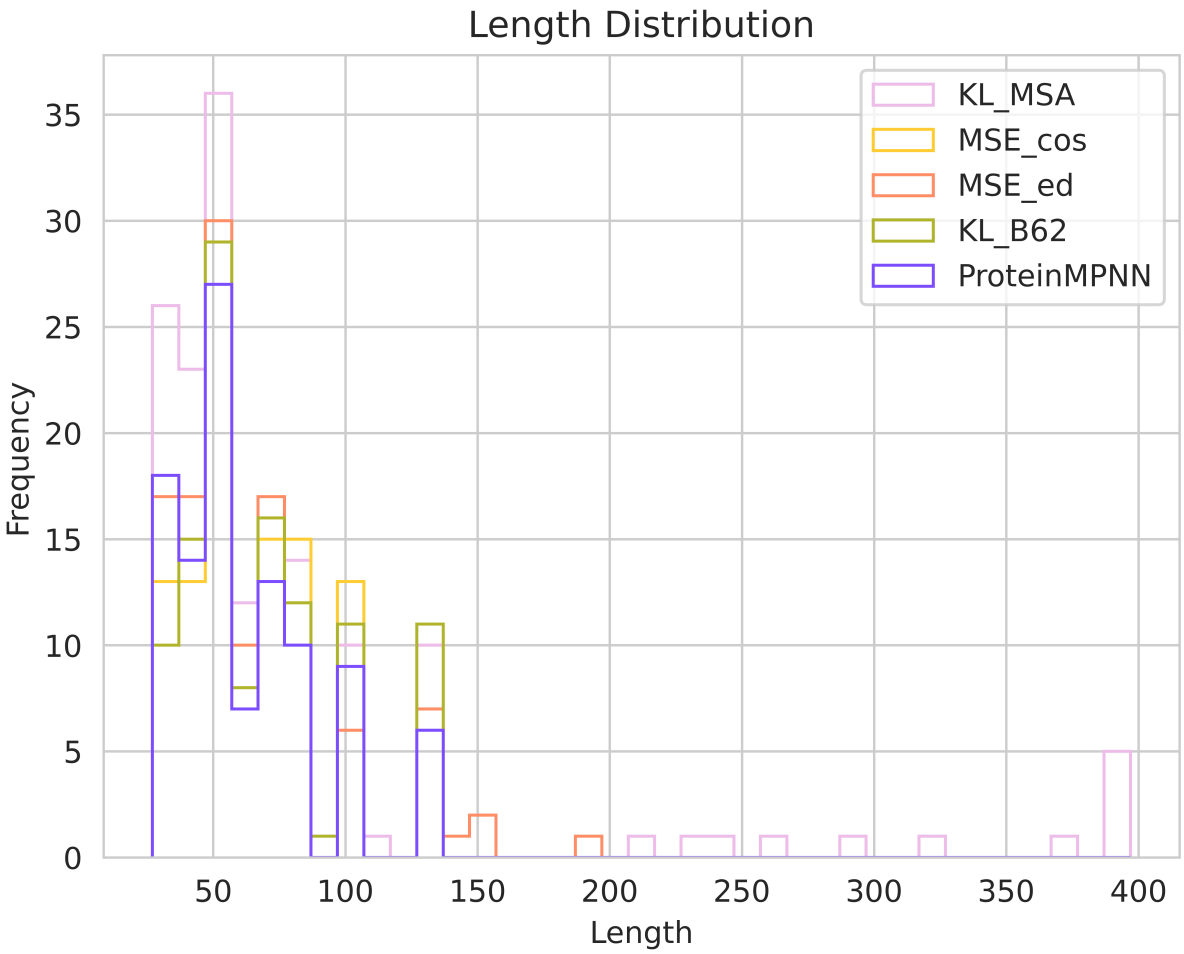

Figure 4.2: The length distribution of the selected protein designs

Chapter 5

Discussion and Conclusion

5.1 Effects and limitations of using different optimization strategies

Based on current results, the optimization strategies that utilize MSA information are the most effective, particularly evident in the experiment of expanding the sequence space. However, there is a significant flaw: the RMSD of sequences directly generated using the argmax function is quite poor, significantly trailing other models. Therefore, this strategy cannot be applied in tasks that do not involve sequence diversification, such as complementing the protein generative model.

The reason for this is not inherently due to the use of MSA, but because our method of utilizing MSA information is very basic and inadequate in the current KL_MSA model. We simply convert aligned sequences into position probability matrices, which has sev-

eral disadvantages. First, many useful co-evolutionary signals are lost—we do not know which mutations need to coexist or cannot coexist, and the MSA information is far from being fully utilized. Second, even if a sequence is directly extracted from these position probability matrices using the argmax function, it may not necessarily fold back to the reference structure. This is because it is not any of the sequences in the MSAs but is merely a combination of the most likely amino acids at each position. Such combinations are impractical and may contain various amino acid pairs that should not appear together.

However, the advantages of the MSA-based model become apparent when the sampling temperature is high. At high temperatures, every model tends to sample randomly from the predicted position probability matrix, and it becomes crucial whether the less probable amino acids at specific positions are reasonable. The original ProteinMPNN model can indeed learn a lot through NLL loss, but it is still not as effective as probability distributions predicted using methods based on similarity or BLOSUM. The MSA-based model is unique because using MSA is similar to using a structure-specific BLOSUM, ensuring that the less probable amino acids in the probability distribution appearing at specific positions are completely reasonable. Thus, the higher the sampling temperature, the more pronounced the advantages of the MSA model.

Unlike the MSA-based model, the other three models based on BLOSUM and similarity strategies display similar curves. Both ends of these curves are closely aligned with the positions of the curve of the ProteinMPNN model, and the middle temperature performances are notably distinct, more steeply curved downward to the right on the RMSD vs. Diversity curve and downward to the left on the RMSD vs. Recovery curve, indicating

better performance in these metrics. However, the enhancements in these three models are not as pronounced as those seen in the MSA-based model. Their advantage lies in their ability to improve performance without compromising the refoldability of sequences generated at low temperatures or those derived using the argmax function. Despite this, the refoldability at these endpoints has not shown significant improvement. Fundamentally, whether through BLOSUM or similarity strategies, the reference sequence remains unchanged. Therefore, any improvements are confined within a certain range and do not represent radical changes. Nevertheless, these improvements are still significant, as evidenced by the noticeable effects in the middle sections of the mentioned curves, indicating that the amino acids with lower probabilities in the model's output probability distribution are more logically positioned. This additional logic is derived from learning through BLOSUM or similarity methods.

Considering the limitations of Blosum and similarity strategies, improvements should be made in how MSA information is utilized if we aim to fundamentally enhance model performance. Currently, we identify two viable methods for effective enhancement. The first method involves using higher-dimensional information, such as two-dimensional MSA information. This approach extends beyond merely predicting the potential amino acid probability distributions at each sequence position. It involves understanding the possible amino acid distributions at other positions, given the amino acid type at any specific position, thereby capturing the fundamental co-evolutionary information. The second method entails optimizing the output position probability matrix using both KL divergence loss and NLL loss during the training of MSA-based models. This dual approach allows

the model to maximize the probability of the reference sequence at each sequence position while utilizing MSA information, enabling the model to perform well at low temperatures and conduct reliable sampling at high temperatures.

5.2 The importance of conducting a comprehensive evaluation

The results of this work clearly demonstrate that relying solely on sequence recovery to evaluate the performance of inverse folding models is unreasonable. Although we used RMSD as one of the metrics in Section 4.1 "existing evaluation metrics", in reality, metrics related to refoldability only began to be proposed and utilized with initiatives like ProteinInvBench and PDBStruct. Currently, most articles on inverse folding continue to prioritize models' performance based on recovery while overlooking other aspects, a practice that is not advisable. Recovery at the sequence level does not equate to refold at the structural level, this statement is already well substantiated by the results of our experiments. From the RMSD vs. Recovery curves, it is evident that, even for identical recovery values, RMSD can vary significantly. Further, in Section 4.3, the proteins we selected all show very low RMSD in comparison to the reference protein, yet exhibit substantial divergence from the reference sequence.

Additionally, we can also discover from the results of this work that even when incorporating metrics such as diversity and refoldability into the evaluation, without conducting further evaluations related to sampling temperature, the assessment of model performance

remains incomplete. Through the experiments in Chapter 4, we now fully understand that the various strategies we tested proved to be useful. However, if we rely solely on the results in Section 4.1, we cannot discern any significant advantages of my models over the original ProteinMPNN model. In particular, the KL_MSA model, if evaluated only using sequences generated by the argmax function, turns out to be the worst performer among all the models. Therefore, this also underscores the value of conducting a comprehensive evaluation, including plotting RMSD vs. Diversity and RMSD vs. Sequence recovery curves, as well as comparing the models' abilities to expand the sequence space. Without these comprehensive assessments, many advantages and disadvantages related to the practical applications of the models would remain hidden.

Appendix A

Implementation Details

A.1 Training and validation loss visualization

From Figure A.1, we can observe that although not every model has fully converged by 150 epochs, and some models' performances are still improving, each model has entered a plateau phase where any improvements are relatively slow. Therefore, choosing 150 epochs for subsequent evaluation is reasonably justified.

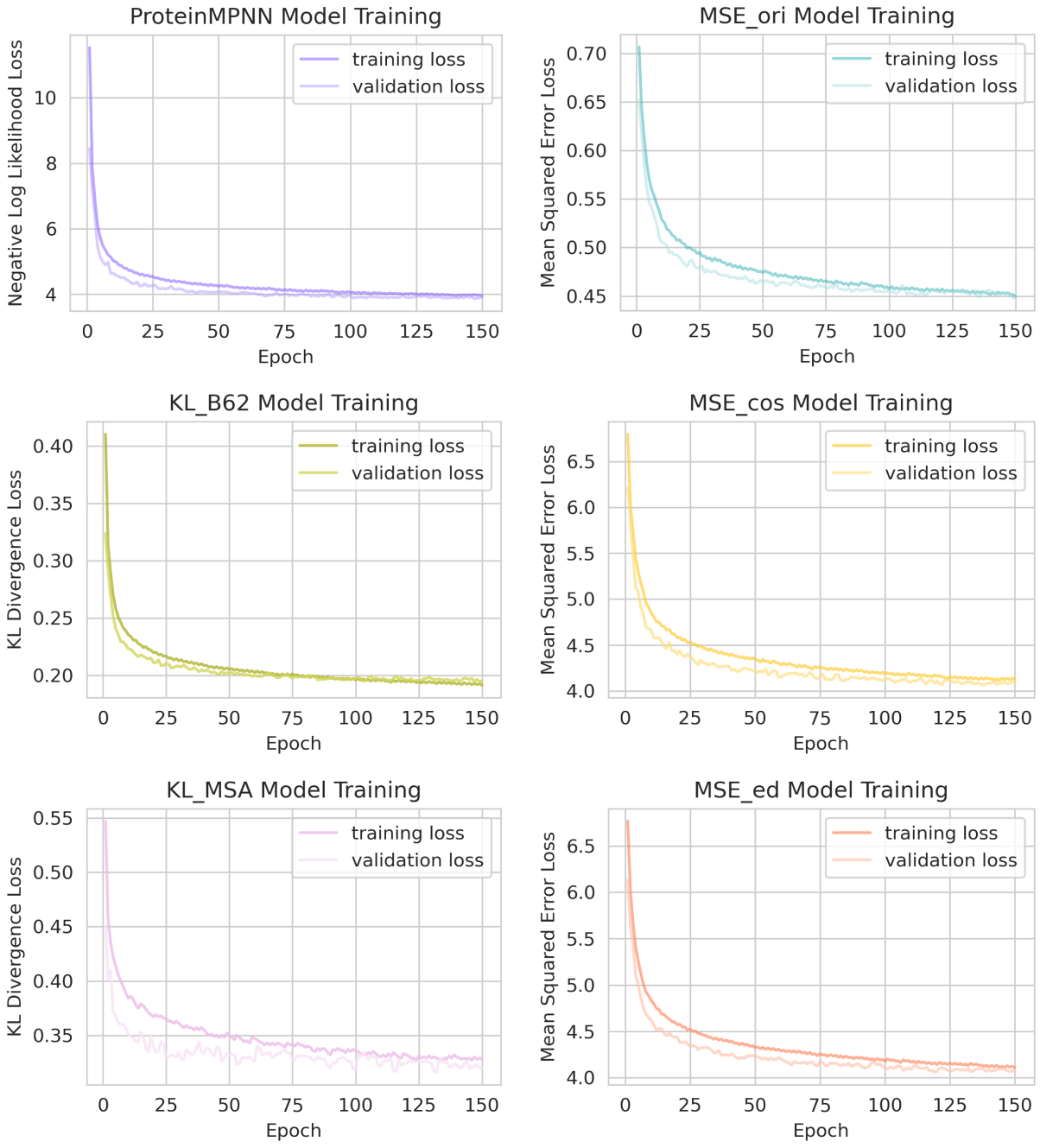

Figure A.1: Training and validation loss of different models

A.2 Retraining the ProteinMPNN model

Compared to the original ProteinMPNN model, my retrained ProteinMPNN model demonstrates comparable performance.

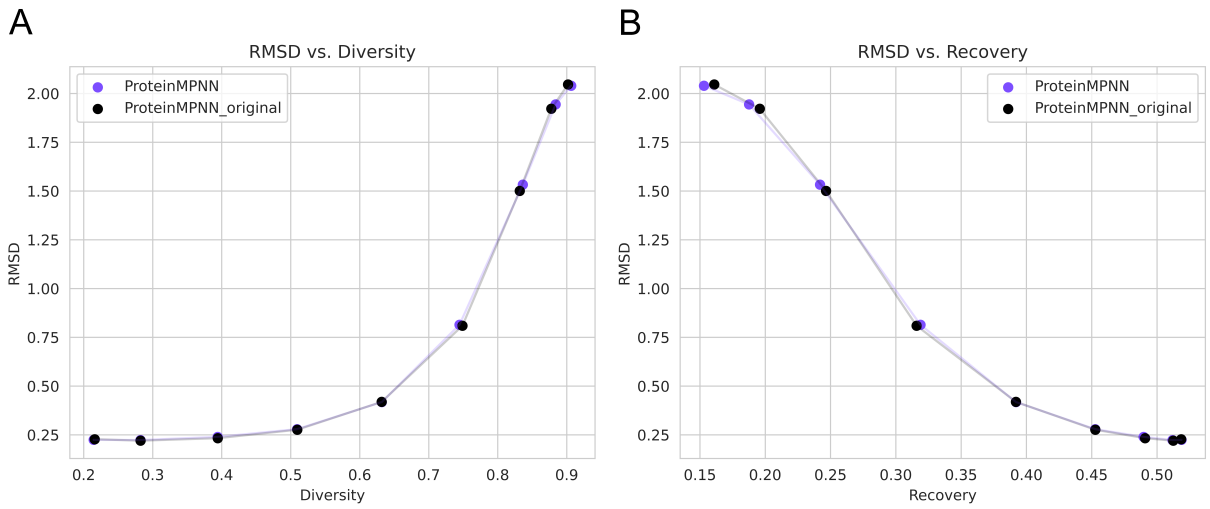

Figure A.2: The curves on CAMEO dataset. A: RMSD vs. Diversity curve, B: RMSD vs. Sequence recovery curve.

Appendix B

Extended Experimental Results

B.1 Supplementary results visualization

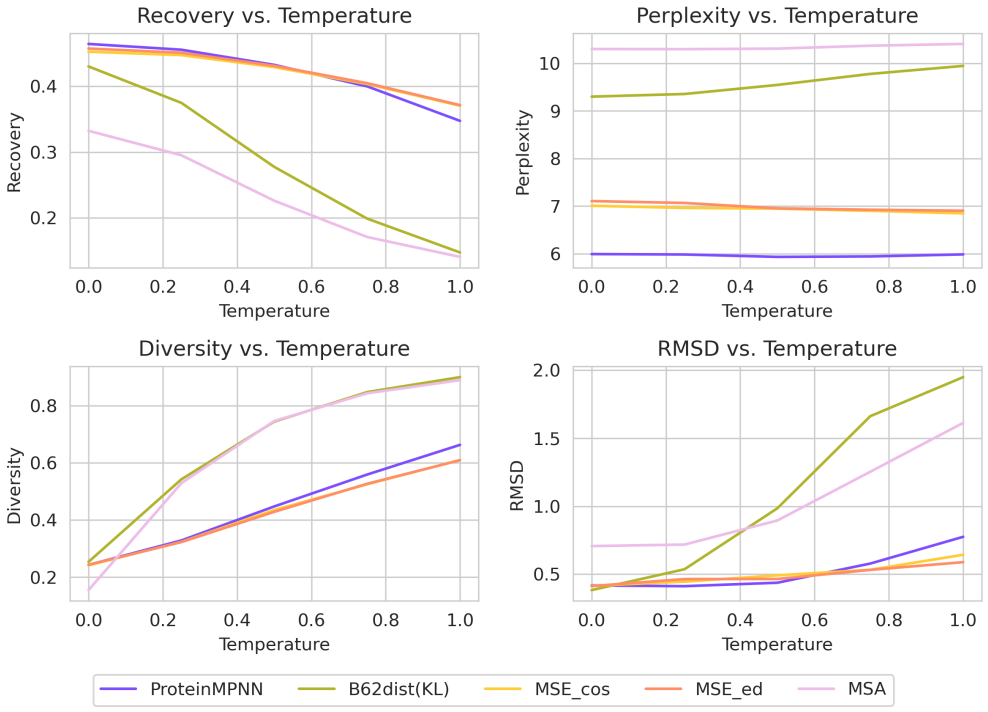

Figure B.1: The curves depict the performance of the models across four different metrics as a function of sampling temperature.

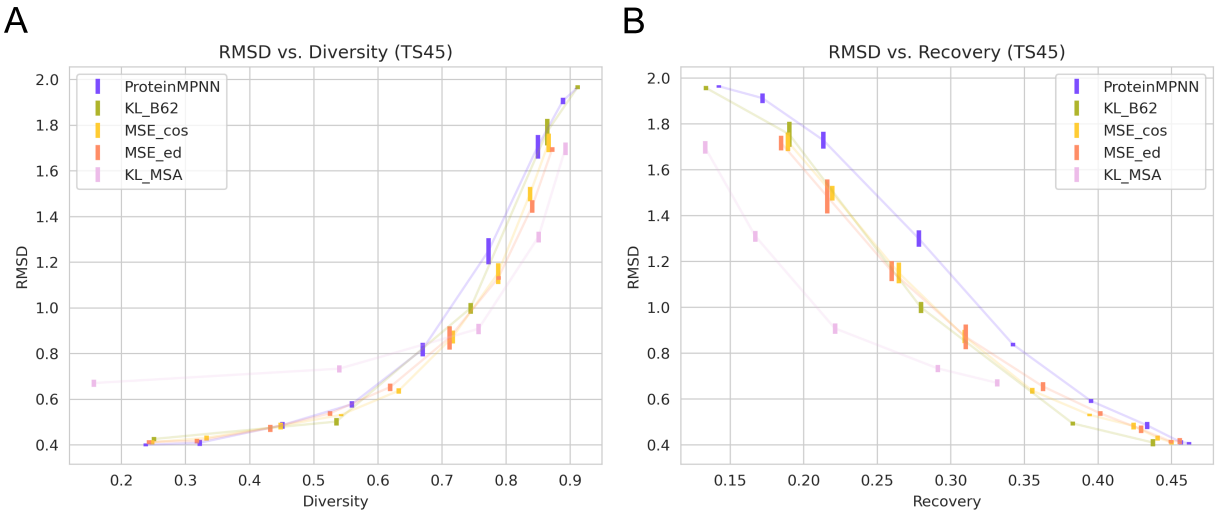

Figure B.2: The curves on TS45 dataset. A: RMSD vs. Diversity curve, B: RMSD vs. Sequence recovery curve.

B.2 Detailed results of sequence space expansion

Table B.1: ProteinMPNN

Temperature	total_num	reference_num	average_length
0	88	16	60.15
0.25	91	18	61.63
0.5	104	20	62.97
0.75	104	21	61.07
1	71	17	61.49
1.25	43	16	50.70
1.5	33	13	55.03
1.75	18	10	43.89
2	8	6	53.63

Table B.2: KL_B62

Temperature	total_num	reference_num	average_length
0	91	16	62.98
0.25	113	23	67.82
0.5	27	13	49.22
0.75	15	10	42.20
1	6	4	43.17

Table B.3: MSE_cos

Temperature	total_num	reference_num	average_length
0	82	15	63.17
0.25	86	19	64.65
0.5	102	17	64.50
0.75	115	22	65.38
1	109	24	60.76
1.25	97	22	56.89
1.5	85	20	56.87
1.75	52	17	52.17
2	52	14	52.54

Table B.4: MSE_ed

Temperature	total_num	reference_num	average_length
0	104	16	62.92
0.25	101	17	62.18
0.5	106	20	63.26
0.75	116	19	62.07
1	109	23	61.40
1.25	120	23	64.40
1.5	78	21	55.44
1.75	57	19	51.70
2	52	13	50.96

Table B.5: KL_MSA

Temperature	total_num	reference_num	average_length
0	116	19	67.09
0.25	128	28	67.74
0.5	163	40	81.59
0.75	112	31	62.59
1	35	12	68.87

Bibliography

[1] D. Akpinaroglu, K. Seki, E. Zhu, and T. Kortemme, "Frame2seq: structure-conditioned masked language modeling for protein sequence design,"

[2] J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, N. Hanikel, S. J. Pellock, A. Courbet, W. Sheffler, J. Wang, P. Venkatesh, I. Sappington, S. V. Torres, A. Lauko, V. De Bortoli, E. Mathieu, S. Ovchinnikov, R. Barzilay, T. S. Jaakkola, F. DiMaio, M. Baek, and D. Baker, "De novo design of protein structure and function with RFdiffusion," vol. 620, no. 7976, pp. 1089–1100.

[3] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. De Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker, "Science - ProteinMPNN," vol. 378, no. 6615, pp. 49–56.

[4] Y. Liu and H. Liu, "Protein sequence design on given backbones with deep learning," vol. 37, p. gzad024.

[5] X. Zhang, H. Yin, F. Ling, J. Zhan, and Y. Zhou, "SPIN-CGNN: Improved fixed backbone protein design with contact map-based graph construction and contact graph neural network," vol. 19, no. 12, p. e1011330.

[6] J. Ingraham, V. Garg, R. Barzilay, and T. Jaakkola, "Generative models for graph-based protein design,"

[7] B. Jing, S. Eismann, P. Suriana, R. J. L. Townshend, and R. Dror, "Learning from protein structure with geometric vector perceptrons." Number: arXiv:2009.01411.

[8] K. K. Yang, H. Yeh, and N. Zanichelli, "Masked inverse folding with sequence transfer for protein representation learning."

[9] 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."

[10] Z. Gao, C. Tan, P. Chacón, and S. Z. Li, "PiFold: Toward effective and efficient protein inverse folding."

[11] Z. Gao, C. Tan, and S. Z. Li, "Knowledge-design: Pushing the limit of protein design via knowledge refinement."

[12] M. H. Høie, A. M. Hummer, T. H. Olsen, M. Nielsen, and C. M. Deane, "AntiFold: Improved antibody structure design using inverse folding,"

[13] Z. Gao, C. Tan, Y. Zhang, X. Chen, L. Wu, and S. Z. Li, "ProteinInvBench: Benchmarking protein inverse folding on diverse tasks, models, and metrics,"

[14] C. Wang, B. Zhong, Z. Zhang, N. Chaudhary, S. Misra, and J. Tang, "PDB-struct: A comprehensive benchmark for structure-based protein design."

[15] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks.," vol. 89, no. 22, pp. 10915–10919.

[16] S. R. Eddy, "Where did the BLOSUM62 alignment score matrix come from?," vol. 22, no. 8, pp. 1035–1036.

[17] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, and R. Fergus, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," vol. 118, no. 15, p. e2016239118.

[18] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, M. Fazel-Zarandi, T. Sercu, S. Candido, and A. Rives, "Language models of protein sequences at the scale of evolution enable accurate structure prediction,"

[19] G. Ahdritz, N. Bouatta, S. Kadyan, L. Jarosch, D. Berenberg, I. Fisk, A. M. Watkins, S. Ra, R. Bonneau, and M. AlQuraishi, "OpenProteinSet: Training data for structural biology at scale,"