ROSETTA ENERGY APPROXIMATION USING A MACHINE LEARNING APPROACH

by
Sen Wei

A thesis submitted to The Johns Hopkins University in conformity with the requirements for the degree of Master of Science

Baltimore, Maryland
May 2024

Abstract

The advent of AlphaFold2 has significantly accelerated advancements in protein structure prediction using deep learning. Despite its monumental success, the AlphaFold2-like learning-based methods lack explainability and generalization, which has limited further understanding and application. Traditionally, the minimizing free energy approach, exemplified by Rosetta, has been cornerstone in protein structure prediction, embedding extensive biophysical insights into its methodology. Notably, when researchers encounter improbable structures, Rosetta is used to refine them, enhancing their physical plausibility. The critical interplay between structure prediction and energy optimization highlights a gap in current deep learning approaches, which overlook the integration of energy information. Addressing this, my project aims to incorporate energy-based metrics into deep learning models, enhancing both their predictive performance, generalization and explainability, alleviating AlphaFold2-like models' heavy reliance on Multiple Sequence Alignments (MSAs) and extensive data sets. By employing equivariant graph neural networks, I have begun to approximate Rosetta's one-body and two-body energy terms, achieving Pearson correlations with Roseta's energy metrics above 0.7 for most terms. My work has prepared machinery to integrate the energy model into some deep learning models like IgFold, an antibody structure prediction method developed by our lab. This integration aims to enhance IgFold's performance and its ability to generalize across diverse antibody structures.

Primary reader and thesis advisor:

Dr. Jeffrey J. Gray
Professor
Department of Chemical and Biomolecular Engineering
Johns Hopkins University, Baltimore MD

Secondary readers:

Dr. Brandon C. Bukowski
Assistant Professor
Department of Chemical and Biomolecular Engineering
Johns Hopkins University, Baltimore, MD

Dr. Jeremias Sulam
Assistant Professor
Department of Biomedical Engineering
Johns Hopkins University, Baltimore, MD

Acknowledgement

I would like to express my deepest appreciation to Professor Jeffrey J. Gray, who not only served as my advisor but also encouraged and challenged me throughout my research, making this thesis possible.

I am profoundly grateful to Laurent Ludwig for his diligence in creating the dataset. My appreciation extends to Lee-Shin Chu and Britnie Carpentier, whose insightful discussions were invaluable in shaping the direction of my work.

Special thanks to Ameya Harmalkar, Rituparna Samanta, and Professor Rocco Moretti for their expert assistance with the PyRosetta energy calculations. I am equally thankful to Jeffrey Ruffolo, AJ Vincelli, Teresa Huang, and Erik Henning Thiede for their insightful and meaningful discussions that significantly contributed to the development of my thesis.

My sincere thanks also go to the team at Gray Lab for their insights and camaraderie. Additionally, I am grateful to the National Institutes of Health (R35 GM141881) for their financial support which was essential in facilitating my research.

Table of Contents

Abstract . . . . . **ii**

Acknowledgement . . . . . **iv**

List of Tables . . . . . **vii**

List of Figures . . . . . **viii**

Chapter 1 Introduction **1**

Chapter 2 One-body and two-body residue-based energy Dataset . **7**

2.1 Energy data 7

2.2 Data Distributions 8

Chapter 3 Deep Learning Methods **12**

3.1 Data Representation 12

3.2 Model Architecture 12

3.3 Loss Functions 14

3.3.1 Weighted MSE Loss 14

3.3.2 Correlation Loss 15

3.4 Optimizer 16

3.5 Energy Clamp and Mask 16

3.5.1 Clamp One-Body Energy 16

3.5.2 Distance Mask for Two-Body Energy 17

Chapter 4 Results **20**

4.1 Training is stable 20

4.2 Energy Approximation Network captures Rosetta energy 21

4.3 Visualization 21

4.4 Ablation Studies 25

4.4.1 The input of my model 25

4.4.2 Test on AFDB5000 26
4.4.3 Test on AFDB10000 28

Chapter 5 Discussion **31**

Bibliographic references **34**

Appendix A Result Mining **38**
A.1 Before and after incorporate orientation 38
A.2 Energy Difference vs Amino Acid Type or Distance 39
A.3 Error vs residue length 41
A.4 Predicted energy vs Rosetta energy of 16 randomly selected test files 43

List of Tables

Table 1.1 One-Body Energies, adapted from [21] 5

Table 1.2 Two-Body Energies, adapted from [21] 6

Table 1.3 Ignored energy terms 6

Table 4.2 Ablation Study on AFDB5000 26

Table 4.3 Ablation Study vs AFDB5000 Baseline 27

Table 4.4 Ablation Study on AFDB10000(The model highlighted in bold is the baseline used by next part, values greater than baseline is shown in bold) 29

Table 4.1 Energy Approximation Network performance on one-body (top) and two-body (bottom) energy prediction 30

List of Figures

Figure 2.1 Distribution of Protein Lengths 8

Figure 2.2 One-Body Energies vs Amino Acid Type, for the various score terms defined in Table 1.1. 10

Figure 2.3 Two-Body Energies vs Distance of Alpha Carbon(zero values are ignored) 11

Figure 3.1 Illustration of the frame from the AlphaFold paper[1]. 13

Figure 3.2 The architecture of the Energy Approximation Network. 15

Figure 3.3 One-Body Energy Term Clamped at $5\% \sim 95\%$ vs Amino Acid Type 18

Figure 3.4 Clamped Two-Body Energy Term vs Distance of Alpha Carbon (zero values are ignored) 19

Figure 4.1 Loss Plot 20

Figure 4.2 One-Body Comparison: Predicted Energy vs Rosetta Energy . . 22

Figure 4.3 Two-Body Comparison: Predicted Energy vs Rosetta Energy . . 24

Figure 5.1 Loss Weight Decay Plot 33

Figure A.1 Result example w/o and w orientation. 38

Figure A.2 One-Body Energy Difference vs Amino Acid Type 39

Figure A.3 Two-Body Energy Prediction Difference vs $C_\alpha - C_\alpha$ Distance. . 40

Figure A.4 One-Body Energy Correlation vs Sequence Length 41

Figure A.5 Two-Body Energy Correlation vs Sequence Length 41

Figure A.6 One-Body Energy MSE vs Sequence Length 42

Figure A.7 Two-Body Energy MSE vs Sequence Length 42

Figure A.8 Predicted One-Body Energy vs Rosetta Energy of 16 randomly selected test files 43

Figure A.9 Predicted two-body Energy vs Rosetta Energy of 16 randomly selected test files 44

Chapter 1

Introduction

In recent years, deep learning based methods have achieved significant success in predicting protein structure[1–4]. However, as with machine learning applications in other fields such as computer vision and natural language processing, explainability and generalizability remain problematic due to limitations in data quantity and quality. Additionally, obtaining protein structures through experiments is more complex compared to obtaining pictures or texts. The experimental determination of protein structures is not only time-consuming but also expensive. Additionally, explainability in this field is a greater concern compared to machine learning's application in other fields[5].

Before the dominance of machine learning in the field of protein structure prediction, Rosetta was considered one of the best tools to predict and design protein structures[6]. It is based on minimizing the free energy of the protein, as the native structure is typically the structure with the lowest energy[7]. However, this important information, energy, is completely ignored in most machine learning based methods. Although energy minimizing methods alone cannot achieve results comparable to machine learning based methods, partly due to inaccurate energy calculations, I hypothesize that incorporating this domain knowledge into machine learning model will be helpful. Energy information may not only improve the generalization but also the explainability of the learning-based methods. In this way, I can alleviate

the AlphaFold2-like models' dependence on multiple sequence alignments and also improve the unrealistic predicted structures[8]. Notably, thermodynamic integration has been shown to enhance the performance and generalizability of machine learning-based RNA structure predictions[9], suggesting that similar benefits could extend to protein structure predictions. However, obtaining the Rosetta energy with its gradient is an expensive and cumbersome calculation, which complicates its integration into the existing learning based model. Therefore, the first step is to capture the energy properties, and this forms the cornerstone of my thesis.

The pursuit of a learned energy function for molecular and protein structures can be approached mainly through two methodologies. The first is the supervised method, which relies on experimentally determined energy values. However, the datasets available for such experiments are limited. For instance, databases like GDB[10] and FreeSolv[11] cater predominantly to small molecules, whereas others such as Megascale[12] and FireProt[13] focus on smaller proteins with point mutations. And PDBBind[14] is specialized in binding affinity. There is unclear how well these data will generalize to larger proteins.

The second type of approach is unsupervised learning. For example, DSMBind[15] and Yang et al's deep neural network energy function[16] do not require experimentally determined energy labels. while correlations being drawn between these pseudo-energies and experimental energies or Root Mean Squared Deviation (RMSD) between given structures and natural structures, the resulting pseudo-energies lack direct biophysical explanations and such comparisons may not directly demonstrate causality. Thus, these models are less interpretable, as vital information is only implicitly con-

sidered within the protein structure prediction models.

More importantly, these energy terms are represented by a single scalar indicating the stability of the entire structure or the binding energy, not the breakdown on a residue-by-residue basis. The research conducted by Zhou et al. [17] demonstrates that evaluating energy at the residue level is more effective for fine-tuning models than considering the energy of the entire structure.

The representations of data in computational models varies significantly. For example, all-atom representations are utilized by ANI [18], GraphomerMapper [19], and DSM-Bind [15]. However, these all-atom representations are computationally expensive, limiting their application primarily to small molecules or specific regions of proteins. To facilitate the modeling of larger proteins, some methods, such as ThermoMPNN [13] and ProteinMPNN [2], employ protein backbone coordinates. Others may leverage residue-level representations and invariant features based on distance, direction, and orientation [20].

In my research, I have followed the IgFold and AlphaFold2, employing embeddings and frames of each residue as inputs[1, 4]. To enhance the detail of energy calculations for each residue and expand training datasets, I utilized the AlphaFold DataBase (AFDB), with energy terms from the Rosetta Energy Function 2015 (REF15)[21] as the ground truth.

Geometric Graph Neural Networks (GNN) are particularly effective in terms of molecular representations, [22], and many models mentioned above leverage GNN architectures. To enhance the accuracy of my energy calculations, I selected an Equivariant

Graph Neural Network (EGNN)[23] as my foundational model. This choice was motivated by the EGNN's ability to maintain equivariance across rotations, translations, reflections, and permutations. For practical implementation, I trained the network to approximate the Rosetta energy terms accurately. These terms were calculated using PyRosetta[24], and the term meanings are detailed in Tables 1.1 and 1.2. While I included most energy terms from the REF15, I omitted three terms pro_close, yhh_planarity and dslf_fa13, because they are used in Rosetta to correct fine details (like the closing of the proline group), that are not needed in DL structure predictions like IgFold. The exclusion of pro_close and yhh_planarity is based on the premise that their associated penalty terms can be implicitly learned by the neural network. The term dslf_fa13 was excluded because it is not directly calculated by PyRosetta and does not correlate with the other energy terms.

One-body energy refers to the energy intrinsic to a single residue, which depends only on its residue type and coordinates. Two-body energy arises from interactions between two amino acids, and depends on amino acid types and coordinates of both. Rosetta energy terms are typically sums over atom-atom pairs, but here I seek to estimate the total for the residue or residue pair. Since I'm not utilizing the all-atom models, I have to figure out a way to represent residues using C_α coordinates and orientations just like AlphaFold and IgFold [1, 4], and the model must infer multibody interactions over all residues and pairs to suit the energetic conformations this way.

Table 1.1: One-Body Energies, adapted from [21]

Energy term	Description
fa_dun	probability that a chosen rotamer is native-like given backbone ϕ, ψ torsion angles
fa_intra_rep	repulsive energy between atoms within the same residue
fa_intra_sol_xover4	Gaussian exclusion implicit solvation energy between atoms in the same residue
ref	reference energies for amino acid types (for design)
p_aa_pp	probability of amino acid identity given backbone ϕ, ψ torsion angles
rama_prepro	probability of backbone ϕ, ψ torsion angles given the amino acid type (inbody correctness for adjcent proline)
omega	backbone-dependent penalty for cis ω dihedrals that deviate from 0° and trans ω dihedrals that deviate from 180°
total_score_1b	A linear combination of one-body energy terms

Table 1.2: Two-Body Energies, adapted from [21]

Energy term	Description
fa_atr	attractive van der waals energy between two atoms on different residues
fa_rep	repulsive van der waals energy between two atoms on different residues
fa_sol	Gaussian exclusion implicit solvation energy between protein atoms in different residues
fa_elec	energy of interaction between two nonbonded charged atoms
lk_ball_wtd	orientation-dependent solvation of polar atoms assuming ideal water geometry
hbond_sc	energy of side-chain-side-chain hydrogen bonds
hbond_bb_sc	energy of backbone-side-chain hydrogen bonds
hbond_sr_bb	energy of short-range backbone-backbone hydrogen bonds
hbond_lr_bb	energy of long-range backbone-backbone hydrogen bonds
total_score_2b	A linear combination of two-body energy terms

Table 1.3: Ignored energy terms

Energy term	Description
pro_close	penalty for an open proline ring and proline ω bonding energy
yhh_planarity	sinusoidal penalty for nonplanar tyrosine χ_3 dihedral angle
dslf_fa13	energy of disulfide bridges

Chapter 2

One-body and two-body residue-based energy Dataset

2.1 Energy data

I chose the AlphaFold DataBase (AFDB) [25] as my data source for structures because it contains over 200 million high confidence protein structures. Barrio et al.[26] clustered these structures within AFDB and identifies representative ones, facilitating the filtration process based on characteristics such as representative structure, length, and predicted Local Distance Difference Test (pLDDT). The pLDDT score serves as an indicator of local accuracy.

My collaborator, Laurent Ludwig, downloaded and filtered 10,000 structures from the AlphaFold Database, selecting those with a pLDDT greater than 90 and a length of less than 500 amino acids. I selected these thresholds because a pLDDT score above 90 is considered to reflect very high confidence[1] and the length is feasible with the computational resources available to us. I randomly split this whole dataset into trainset (80%) and testset(20%). Since each structure represents a cluster center of Barrio et al, we can expect the test set to be independent of the training set.

Figure 2.1 illustrates the distribution of protein lengths within the dataset of 10,000 structures. This dataset includes 1,555,172 residues and 334,179,740 residue pairs.

As mentioned ealier, Rosetta energies were calculated using PyRosetta[24]. Each one-

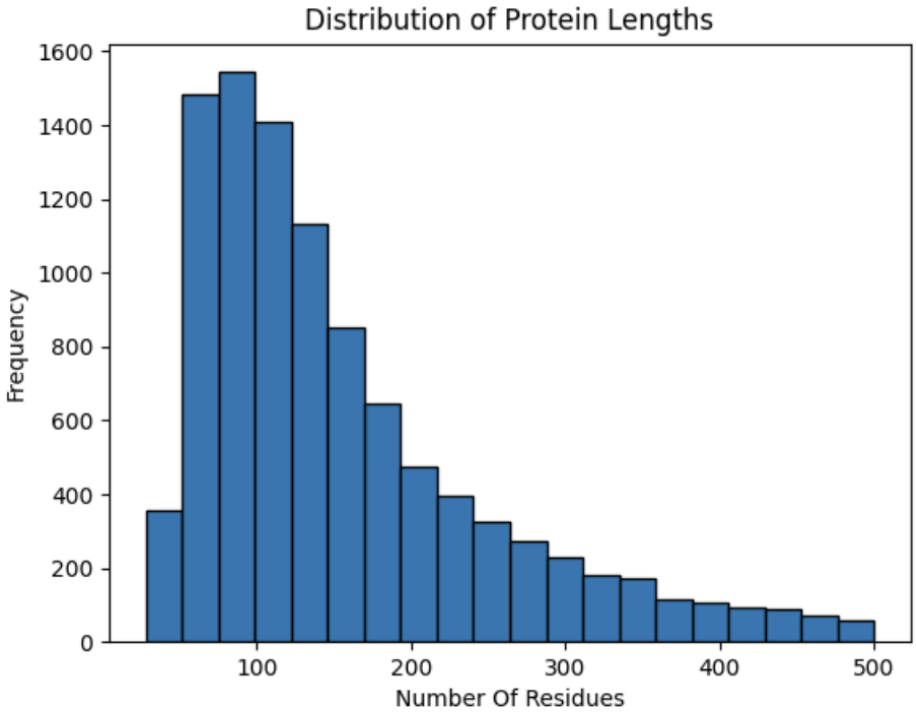


Figure 2.1: Distribution of Protein Lengths

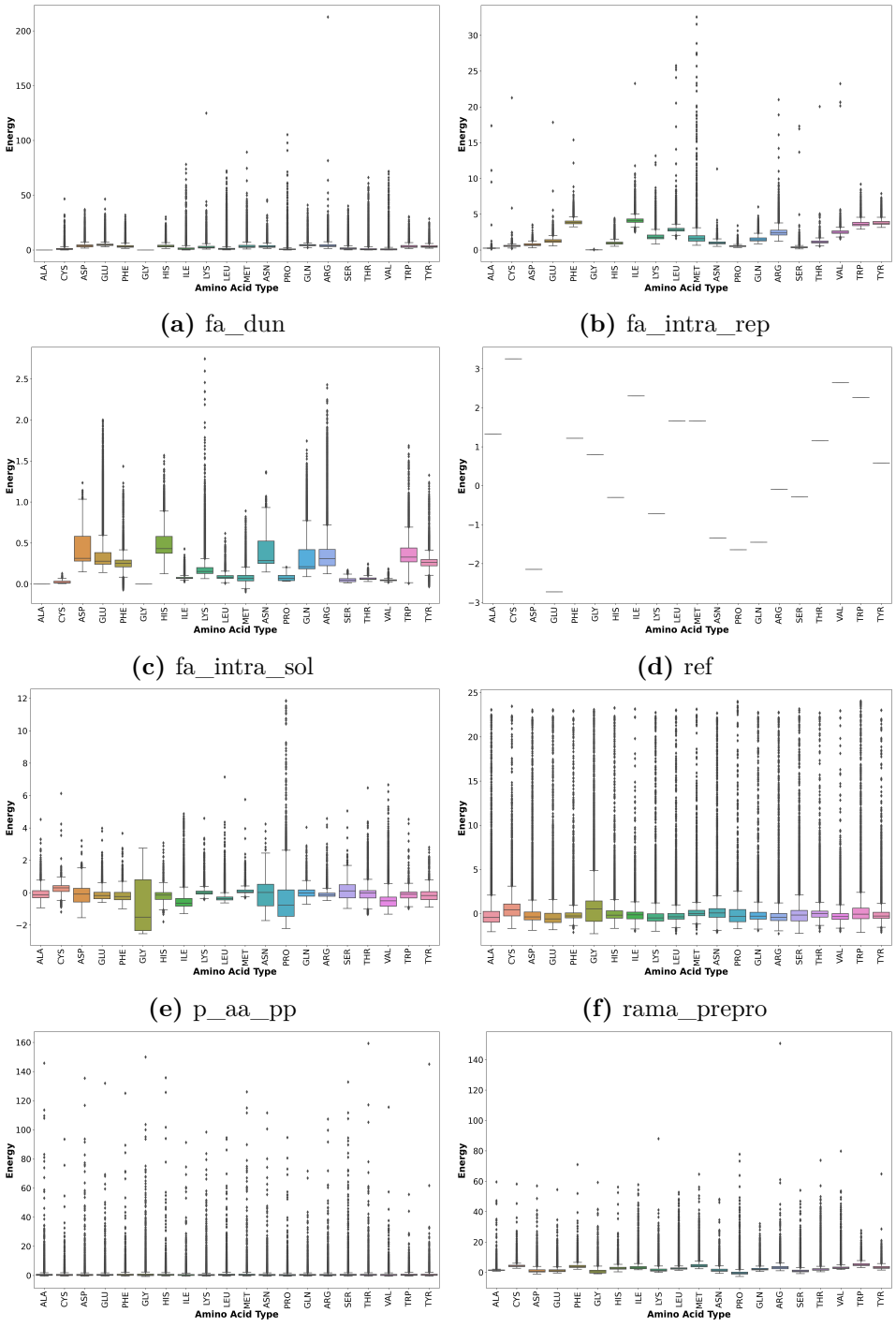
body energy term is represented as a vector of length N, where N is the length of the protein sequence. Each two-body energy term can be represented as a symmetric matrix of shape $N \times N$. Since the upper triangular part contains all the necessary information, we transform it into a vector of length $N \times (N-1)/2$.

2.2 Data Distributions

To enhance my understanding of their relationships, I first created box plots of the one-body energy terms against each amino acid type to enhance my understanding of their relationships. As shown in figure 2.2, most of the one-body energy values are concentrated in a small range, but there are many outliers. The amino acid reference energy (ref, Figure 2.2d) is constant for each residue type, making it the easiest to

learn.

From the scatter plots of two-body energy versus residue-residue $C_\alpha - C_\alpha$ distance, we can recognize patterns for fa_atr, fa_sol, fa_elec, hbond_sr_bb and hbond_lr_bb. However, identifying patterns for fa_rep, hbond_sc, hbond_bb_sc presents more challenges, suggesting these may be more difficult to learn. For all of these terms, the presence of multiple values for a given distance (due to hidden information about side chain positions) complicates the data distribution. Consequently, it is unclear whether a neural network will be able to accurately approximate these terms.



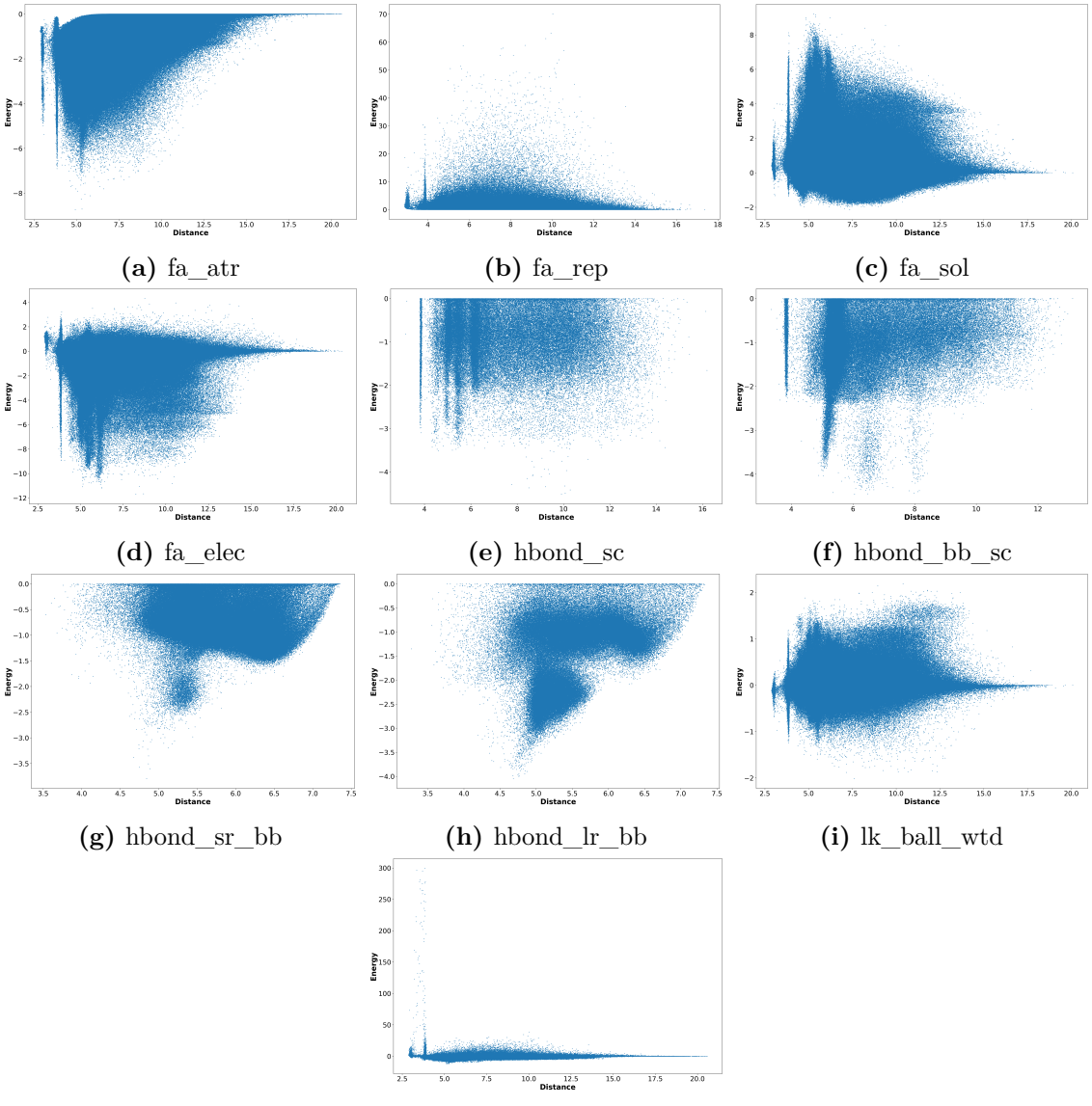
(a) fa_dun **(b)** fa_intra_rep

(c) fa_intra_sol **(d)** ref

(e) p_aa_pp **(f)** rama_prepro

(g) omega **(h)** total_score_1b

Figure 2.2: One-Body Energies vs Amino Acid Type, for the various score terms defined in Table 1.1.



(a) fa_atr **(b)** fa_rep **(c)** fa_sol

(d) fa_elec **(e)** hbond_sc **(f)** hbond_bb_sc

(g) hbond_sr_bb **(h)** hbond_lr_bb **(i)** lk_ball_wtd

(j) total_score_2b

Figure 2.3: Two-Body Energies vs Distance of Alpha Carbon(zero values are ignored)

Chapter 3

Deep Learning Methods

3.1 Data Representation

In my approach, I employ a 20-dimensional one-hot encoding for the protein sequence and use C_α coordinates as input. Furthermore, I calculate the orientations as follows:

$$
\begin{aligned}
\vec{v}_1 &= \vec{x}_3 - \vec{x}_2 \\
\vec{v}_2 &= \vec{x}_1 - \vec{x}_2 \\
\vec{e}_1 &= \vec{v}_1 / \|\vec{v}_1\| \\
\vec{u}_2 &= \vec{v}_2 - \vec{e}_1(\vec{e}_1^T \vec{v}_2) \\
\vec{e}_2 &= \vec{u}_2 / \|\vec{u}_2\| \\
\vec{e}_3 &= \vec{e}_1 \times \vec{e}_2 \\
R &= \text{concat}(\vec{e}_1, \vec{e}_2, \vec{e}_3)
\end{aligned}
\tag{3.1}
$$

Here, N is denoted as $\vec{x}_1$, C_α as $\vec{x}_2$, and C as $\vec{x}_3$. The frames, centered at $\vec{x}_2$ and using orientations, are depicted in Figure 3.1.

3.2 Model Architecture

In this study, I selected the Equivariant Graph Neural Network (EGNN)[23] as the foundational model. EGNN is designed to maintain equivariance under rotations, translations, reflections, and permutations. The natural aptitude of graph-based mod-

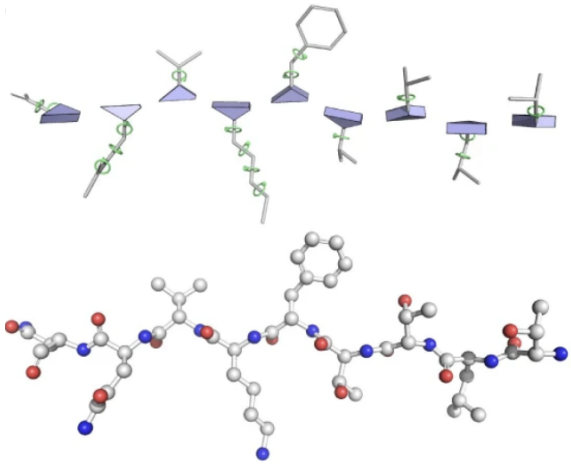


Figure 3.1: Illustration of the frame from the AlphaFold paper[1].
The blue triangles are the representation of each residue as one free-floating rigid body for the backbone. The corresponding atomic structure is shown below.

els to represent molecular structures, coupled with the inherent properties of EGNN, renders it highly appropriate for my research context. This equivariant property is ensured by these layer propogation equations:

$$\begin{aligned} \mathbf{m}_{ij} &= \phi_e(\mathbf{h}_i^l, \mathbf{h}_j^l, \|\mathbf{x}_i^l - \mathbf{x}_j^l\|^2, a_{ij}) \\ \mathbf{x}_i^{l+1} &= \mathbf{x}_i^l + C\sum_{j\neq i}(\mathbf{x}_i^l - \mathbf{x}_j^l)\phi_x(\mathbf{m}_{ij}) \\ \mathbf{m}_i &= \sum_{j\neq i}\mathbf{m}_{ij} \\ \mathbf{h}_i^{l+1} &= \phi_h(\mathbf{h}_i^l, \mathbf{m}_i) \end{aligned} \tag{3.2}$$

Here, $\mathbf{h}^l = \{\mathbf{h}_0^l, \ldots, \mathbf{h}_{M-1}^l\}$ are node embeddings, $\mathbf{x}^l = \{\mathbf{x}_0^l, \ldots, \mathbf{x}_{M-1}^l\}$ are coordinate embeddings and a are edge attributes. ϕ_e, ϕ_x and ϕ_h are the edge, coordinate and node operations respectively which are commonly approximated by Multilayer Perceptrons (MLPs).

Additionally, the computational demands of EGNN are reasonable compared to other models. I conducted comparative evaluations with alternative architectures, including Graph Convolutional Networks (GCN) [27], Graph Isomorphism Network (GIN) [28], and the a Transformer model [29]. In my earlier test, EGNN demonstrated superior performance with few modifications, further substantiating its suitability for the task at hand.

The architecture of the model is illustrated in Figure 3.2. Four EGNN layers are employed to derive node attributes. These attributes are subsequently fed into six linear layers to calculate one-body energies, followed by two additional linear layers to obtain the total one-body score. For the computation of two-body energies, I concatenate the node attributes of i and j along with their pairwise distance. Early experiments indicated that this method yields superior performance compared to techniques such as outer concatenation and outer sum[30].

3.3 Loss Functions

3.3.1 Weighted MSE Loss

During the experiment, I observed that for poorly performing energy terms such as omega (Figure 2.2g), the majority of values were zero. To address this, a Weighted MSE Loss was designed to assign lower weight to these zero values when calculating the loss. The function is presented in Equation 3.3.

$$\textbf{WeightedMSELoss}(X, Y) = \sum_{(x,y)\in(X,Y)} \left(\mathbf{1}_{\{y\neq 0\}} \cdot (1-\epsilon) + \epsilon\right) \cdot \text{MSELoss}(x, y) \quad (3.3)$$

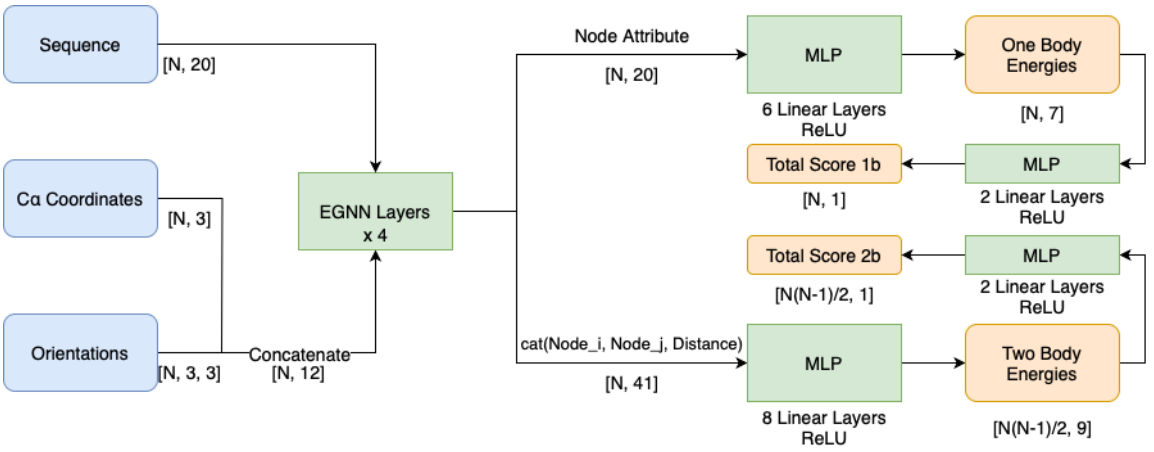


Figure 3.2: The architecture of the Energy Approximation Network.
Schematic representation of the neural network architecture for energy prediction in protein structures. The protein length is denoted by N. Initially, the C_α coordinates, orientations and one-hot encoding of the sequence are inputs. These coordinates and orientations are concatenated and then fed into four Equivariant Graph Neural Network (EGNN) layers. Subsequent to the EGNN layers, node attributes are processed to compute the energy. One-body energies are derived from six linear layers, whereas two-body energies are determined via eight linear layers. Finally, two linear layers are employed to calculate the total scores for one-body (1b) and two-body (2b) energies.

For a specific energy term, X represents the predicted energy, Y denotes the ground truth energy, ϵ is the weight parameter I chose, and MSELoss is the normal Mean Squared Error Loss. In this paper, I use $\epsilon = 0.1$.

3.3.2 Correlation Loss

To improve the correlation of predicted energies to Rosetta energies, I introduced a correlation loss by substracting the sum of the Pearson correlation coefficient from the number of predicted energy terms:

$$\textbf{CorrelationLoss}(X, Y) = 1 - \rho_{X,Y} = 1 - \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{3.4}$$

For a specific energy term, X represents the predicted energy while Y denotes the ground truth energy, and $\rho_{X,Y}$ is the Pearson correlation of X and Y.

3.4 Optimizer

I use the AdamW Optimizer[31] instead of Adam[32], Stochastic Gradient Descent, or Adam optimizer with warm up[29]. A detailed comparison can be found in Table 4.3 and 4.4.

3.5 Energy Clamp and Mask

3.5.1 Clamp One-Body Energy

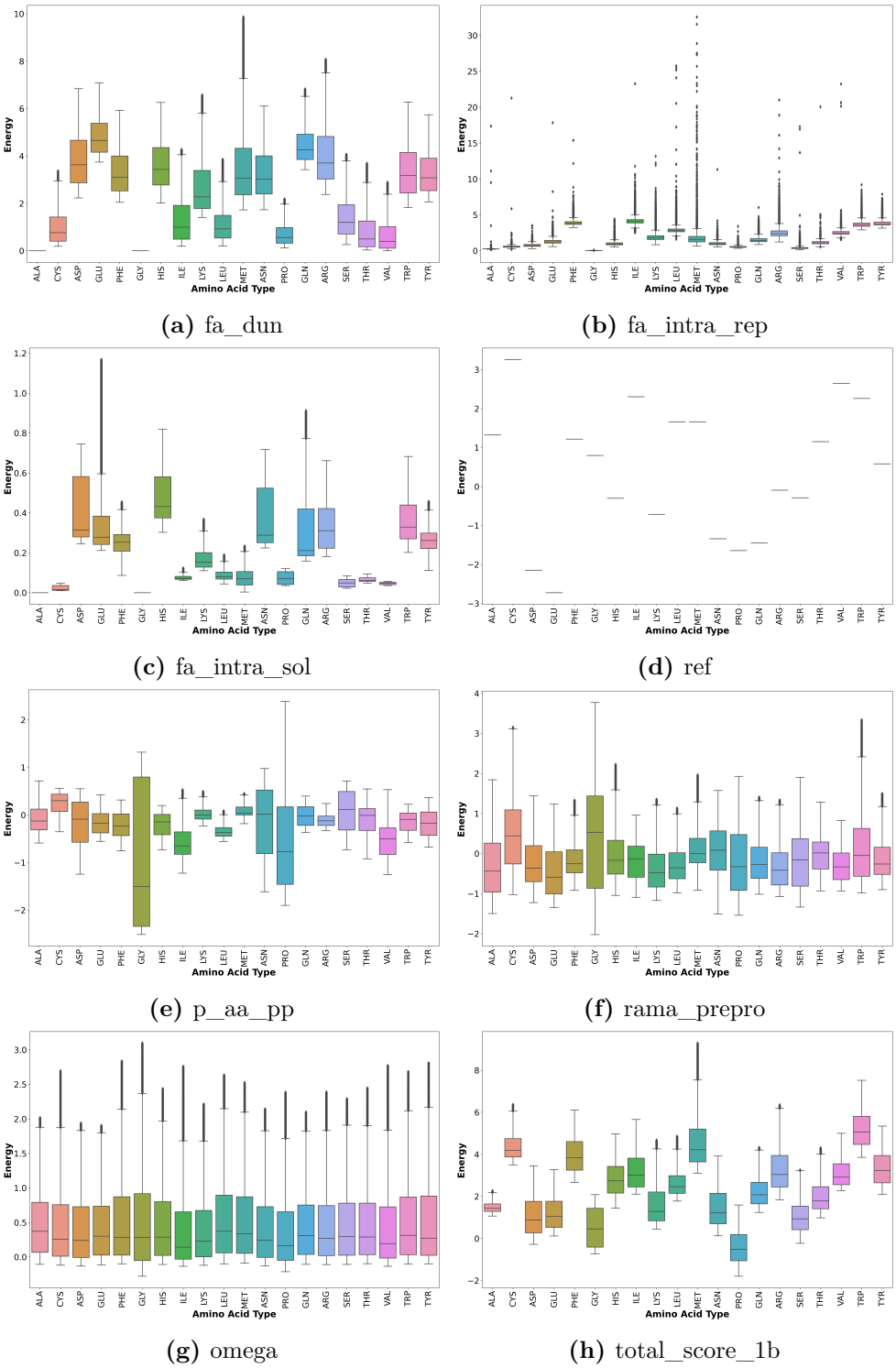
Due to the presence of many outliers of the one-body energies (Figure 2.2), I apply clamping. One-body energy is constrained to 0% and 95%. Any values above 95% are set to 95%. The resulting one-body energy data distribution is shown in Figure 3.3. A comparison with Figure 2.2 reveals the changes. After clamping, many outliers were removed, resulting in a clearer distribution.

I did not clamp the two-body energies because there are few extreme outliers and clamp would remove implicitly patterns in the energy distribution like shown in Figure 3.4, which could negatively impact my model.

An exception is the fa_rep term, which can become very large when atoms clash. Thus, I clamp the fa_rep term maximum at 10.

3.5.2 Distance Mask for Two-Body Energy

Figure 2.3 reveals that Rosetta pairwise energies are zero beyond a certain distance. Therefore, I have applied a mask to all residue pairs that are farther than 15 Å.



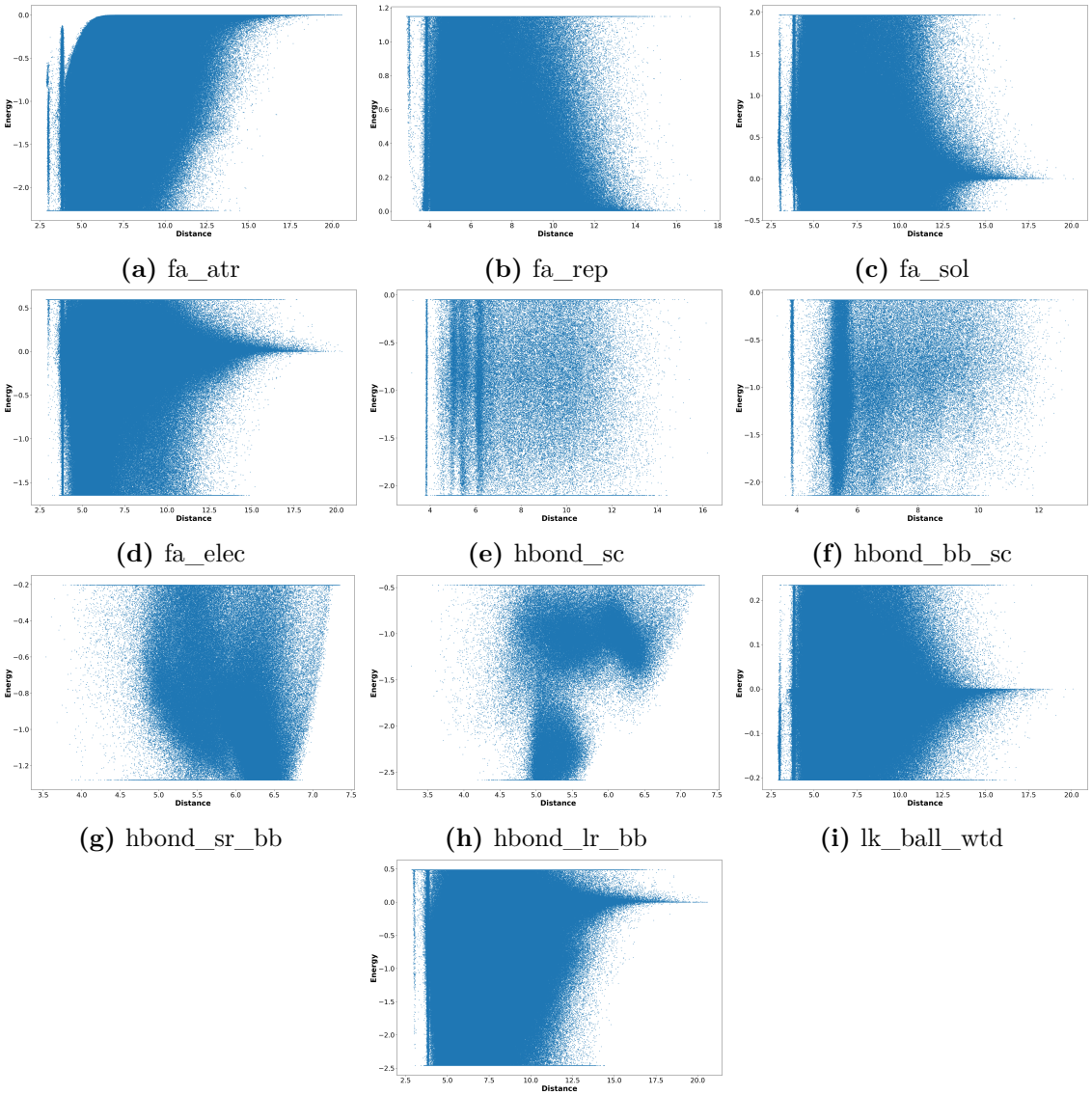
(a) fa_dun **(b)** fa_intra_rep

(c) fa_intra_sol **(d)** ref

(e) p_aa_pp **(f)** rama_prepro

(g) omega **(h)** total_score_1b

Figure 3.3: One-Body Energy Term Clamped at 5% $\sim$ 95% vs Amino Acid Type



(a) fa_atr **(b)** fa_rep **(c)** fa_sol

(d) fa_elec **(e)** hbond_sc **(f)** hbond_bb_sc

(g) hbond_sr_bb **(h)** hbond_lr_bb **(i)** lk_ball_wtd

(j) total_score_2b

Figure 3.4: Clamped Two-Body Energy Term vs Distance of Alpha Carbon (zero values are ignored)

Chapter 4

Results

4.1 Training is stable

As Figure 4.1 shows, the training is quite stable.

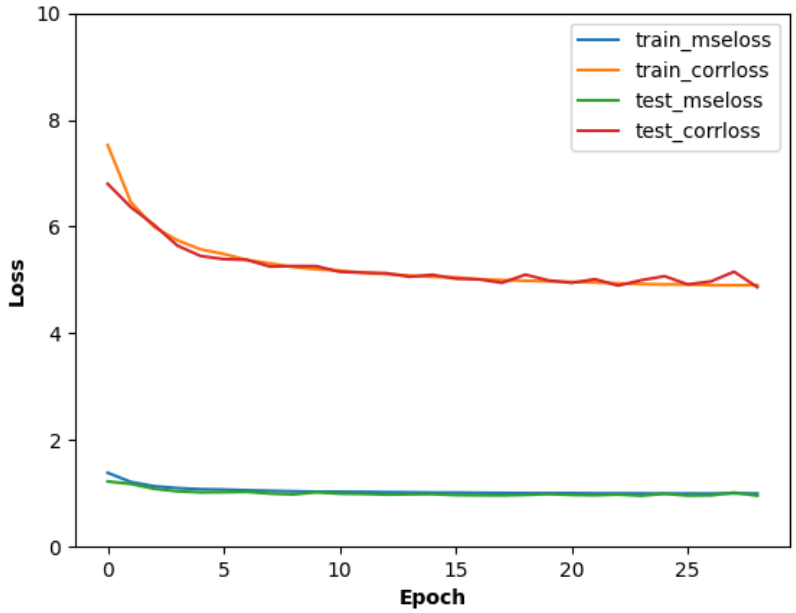


Figure 4.1: Loss Plot

4.2 Energy Approximation Network captures Rosetta energy

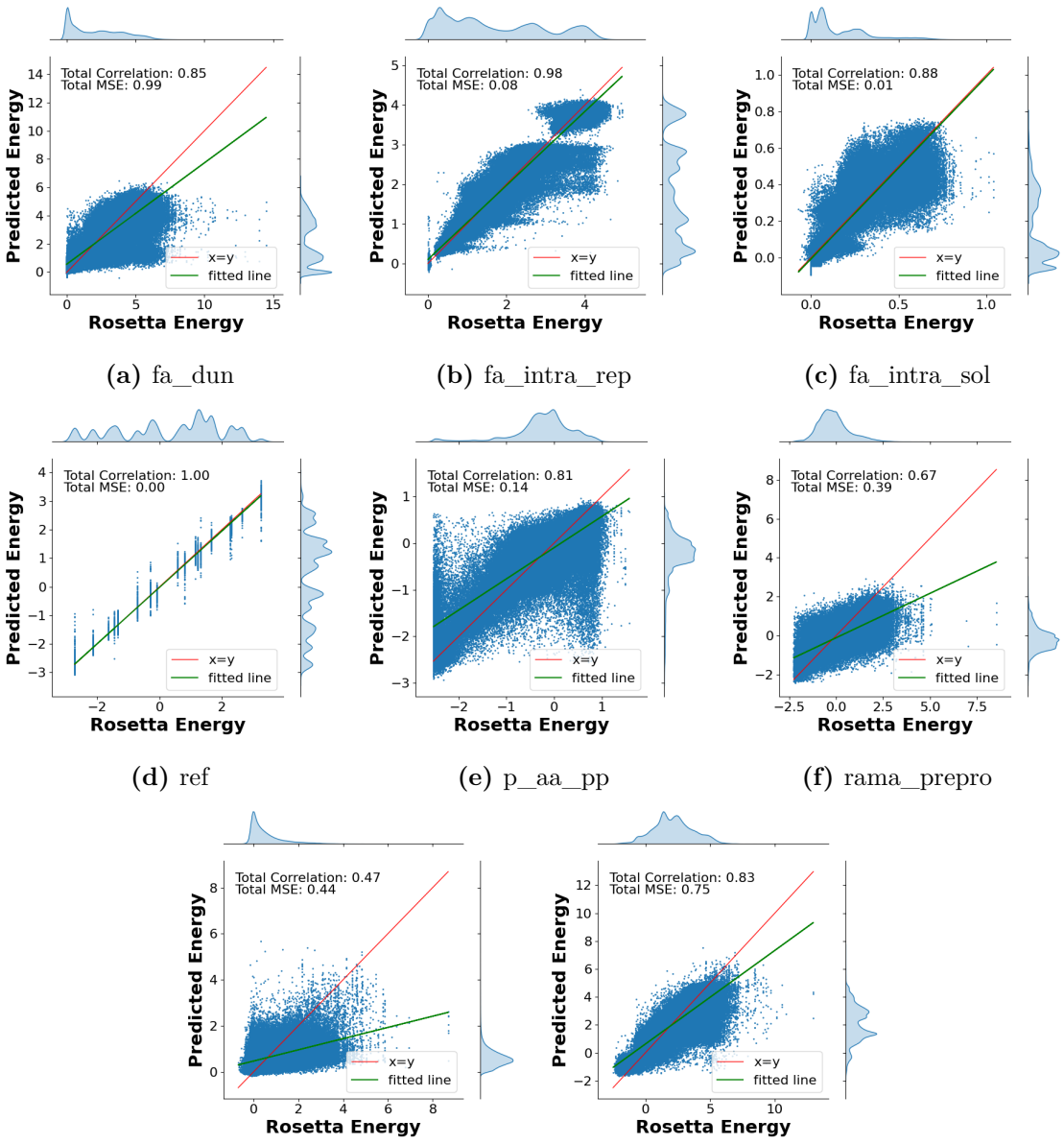
To evaluate the model performance, I calculated the Pearson correlation and Mean Squared Error (MSE) between predicted energy and ground truth energy (Table 4.1). To illustrate the efficiency, I also compared the computing time.

The model obtains energy from a pdb in 40ms, compared to 900ms for PyRosetta, which is more than 22 times faster. I achieved a high correlation (>0.7) for most energy terms.

4.3 Visualization

To better interpret these correlation and MSE numbers, I plot predicted energy versus ground truth energy in the testset (Figure 4.2 and 4.3). In each plot the red line represents the ideal $y = x$ relationship, while the green line shows the best fit derived from all data points within the plot. The accompanying marginal density plots reveal the distribution of these points.[1]

[1]The correlation and MSE values displayed in each plot may slightly diverge from those listed in the tables. The discrepancy arises because the plot values are computed across the entire test set, whereas the table values represent an average computed for each protein within the test set.



(a) fa_dun **(b)** fa_intra_rep **(c)** fa_intra_sol

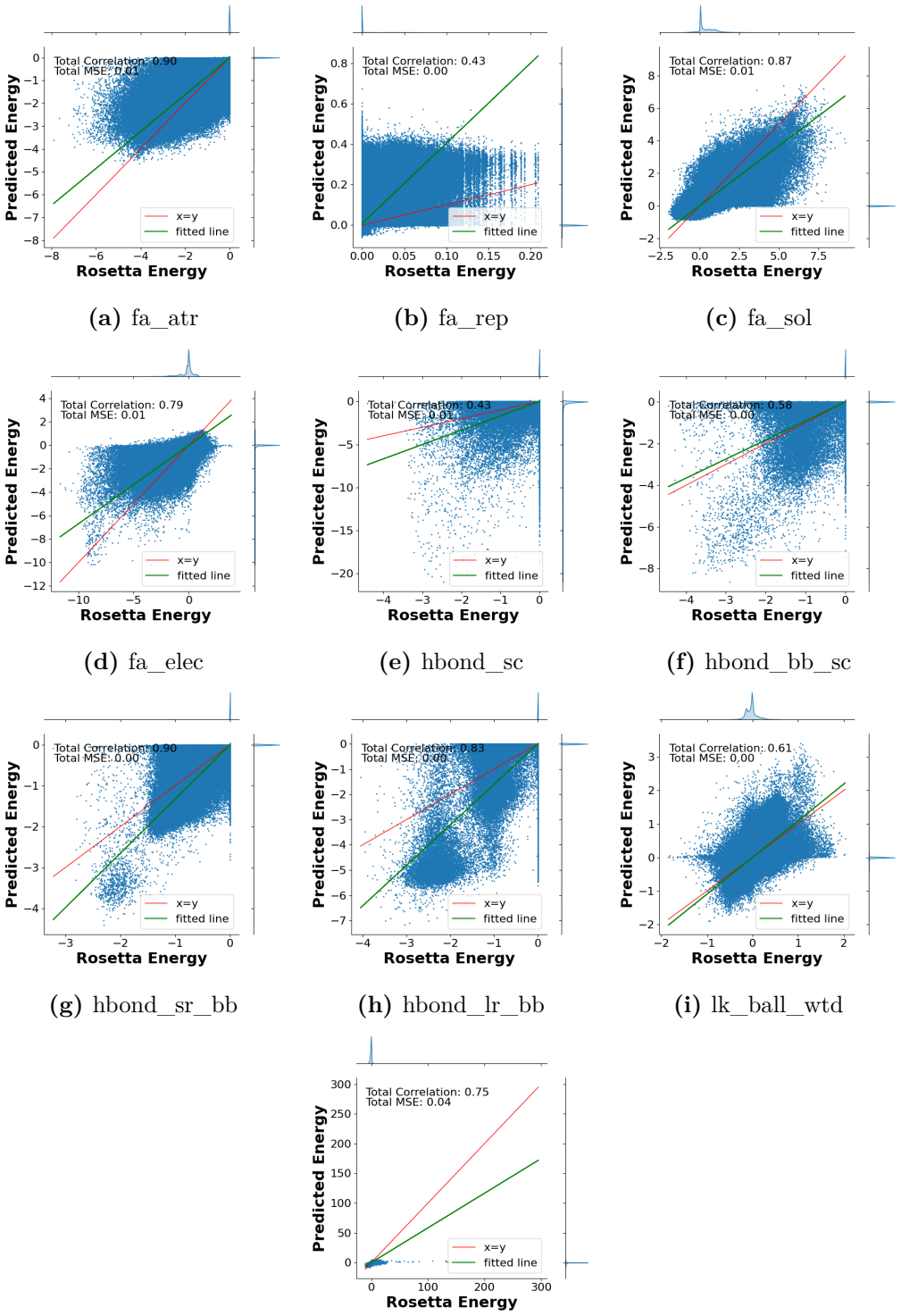
(d) ref **(e)** p_aa_pp **(f)** rama_prepro

(g) omega **(h)** total_score_1b

Figure 4.2: One-Body Comparison: Predicted Energy vs Rosetta Energy

In the one-body energy comparisons, a strong correlation is evident for terms like fa_dun, fa_intra_rep, fa_intra_sol, total_score_1b, and ref. An interesting obser-

vation for the ref energy term is that despite a high correlation score, there exists a variation in predicted energy for a specific Rosetta energy value, suggesting that factors beyond residue type are influencing this estimate. The effect of clamping is noticeable in terms like p_aa_pp. For energy terms that initially performed poorly, such as rama_prepro and omega, the small MSE did not translate into strong correlation, indicating a need for alternative loss functions. Indeed, after switching to WeightedMSE, I improved these terms.



(a) fa_atr

(b) fa_rep

(c) fa_sol

(d) fa_elec

(e) hbond_sc

(f) hbond_bb_sc

(g) hbond_sr_bb

(h) hbond_lr_bb

(i) lk_ball_wtd

(j) total_score_2b

Figure 4.3: Two-Body Comparison: Predicted Energy vs Rosetta Energy

In the two-body energy comparisons, terms such as fa_atr, fa_sol, fa_elec, hbond_sr_bb, hbond_lr_bb, and total_score_2b achieved good correlations ($\rho > 0.7$). The total_score_2b was impacted by outliers, which I suspect might be miscalculations by Rosetta. Terms like fa_rep demonstrate weak learning, as indicated by several vertical clusters suggesting inaccurate approximations. For the hbond terms, the dominance of zero values—implying that most residue pairs do not form hydrogen bonds—signifies that the model struggles to learn this particular pattern. For lk_ball_wtd, the broader distribution of Rosetta Energy compared to the predicted energy's narrower range is an anomaly that remains unexplained.

In summary, the model excels in capturing the trends of energy terms with distinct distributions. However, for energy terms predominantly characterized by zeros, learning proves more challenging. For a more granular view, detailed plots for randomly selected proteins are presented in Appendix Figures A.8 and A.9.

4.4 Ablation Studies

4.4.1 The input of my model

Initially, the model utilized only C_α coordinates; however, this approach yielded unsatisfactory results due to the occurrence of multiple energy values corresponding to a single distance. The integration of orientations into the model presented a challenge. After careful consideration, I adopted a method that involved reshaping and concatenating the orientation data with the coordinates. This modification led to an improvement in the model's performance, as evidenced by the results displayed in Figure A.1.

4.4.2 Test on AFDB5000

Initially, I tested the model on a dataset comprising 5,000 structures from the AlphaFold Database (AFDB). Through these experimental trials, I discovered that implementing a correlation loss and increasing the number of linear layers contributed to enhanced performance (Table 4.2). Therefore, I established this configuration as my baseline. Table 4.2 summaries performance of additional network variations that I explored in my study.

In those early studies, I obtained the total_score_1b and total_score_2b directly, alongside other energy terms, rather than calculating them through an MLP subsequent to deriving the other energy terms. Subsequent to conducting an ablation study, I determined that setting the batch size to 2 was the only modification that enhanced my model's performance. Alternative configurations, such as employing Mean Absolute Error (MAE) or Huber loss, utilizing stochastic gradient descent (SGD) or Adam optimization with a warm-up phase, adjusting batch sizes, or modifying the settings of the EGNN, or incorporating generated bad decoys of the same sequence, did not yield any performance improvements. Here, I only show results of the total_score_1b and total_score_2b. Certain energy terms may have different trends.

Table 4.2: Ablation Study on AFDB5000

Model	total_score_1b	total_score_2b
MSELoss	**0.6806**	0.5302
MSELoss+CorrLoss	0.6615	**0.5732**
MSE+Corr_deeper_EGNN	0.6334	-0.0008
MSE+Corr_deeper_linear	**0.7402**	**0.7575**

Table 4.3: Ablation Study vs AFDB5000 Baseline

Model	total_score_1b	total_score_2b
baseline	**0.7402**	0.7575
Different Loss Functions		
MAE+CorrLoss	0.7110	0.6364
Huber+CorrLoss	0.7029	0.6240
Different Optimizers		
SGD	0.6858	0.6046
AdamwithWarmUp	0.6835	0.5864
Different Batch sizes		
batchsize2	0.7396	**0.7665**
batchsize3	0.6922	0.6246
batchsize4	0.7315	0.7278
batchsize5	0.6441	0.4705
Different Settings of EGNN		
batch2+nc	0.7396	**0.7665**
batch2+nf	0.7289	0.7345
batch2+nfnc	0.7289	0.7345
batch2+uc	0.6889	0.6739
batch2+ucnc	0.7357	0.7576
batch2+nfucnc	0.7297	0.7457
batch2+validRadius12	0.6888	0.6414
Add generated bad decoys of the same sequences		
+1timeBadDecoys	0.6663	0.4097
+5timesBadDecoys	0.6246	0.2771
OnlyBadDecoys	0.7054	0.2784

4.4.3 Test on AFDB10000

Upon expanding my dataset to 10,000 entries and incorporating previously omitted energy terms such as rama_prepro, hbond_sr_bb, and hbond_lr_bb, I observed significant improvements. The inclusion of additional Rosetta energy terms, the application of Weighted Mean Squared Error Loss (WeightedMSELoss), the use of AdamW optimization, the independent calculation of total energy, clamping of one-body energy, and the utilization of a distance mask were all beneficial strategies that contributed to the enhanced performance of my model (Table 4.4).

Table 4.4: Ablation Study on AFDB10000(The model highlighted in bold is the baseline used by next part, values greater than baseline is shown in bold)

Model	total_score_1b	total_score_2b
10000	0.6922	0.6246
10000_Addrama_srlrhbond	**0.7484**	**0.8117**
AdamW	**0.7494**	**0.8171**
dropout0.1	0.7183	0.6946
Add1timeBadDecoys	0.7088	0.3424
Add1timeBadDecoys_sameTestset	**0.8258**	0.3456
RosettaRelaxedStructures	**0.8904**	**0.8123**
clamp1b	**0.8074**	0.7845
clampAll	**0.8251**	nan
maskDist15	0.7415	0.7952
maskDistandClamp	**0.8226**	**0.8226**
WeightedMSE	**0.7508**	**0.8196**
WeightedMSE+Mask	**0.7536**	**0.8211**
WeightedMSE+Mask+Clamp1b	**0.8232**	**0.8272**
SmallerWeightedMSE+Mask+Clamp1b	**0.8229**	**0.8271**
separateCalculateOneandTwo	0.6990	0.7123
separateCalculateTotal	**0.8257**	**0.8326**
separateCalculateTotal_smallerWeightedMSE	**0.8297**	**0.8326**

Table 4.1: Energy Approximation Network performance on one-body (top) and two-body (bottom) energy prediction

Energy term	Correlation	MSE
fa_dun	0.8546	1.0033
fa_intra_rep	0.9765	0.0766
fa_intra_sol_xover4	0.8918	0.0078
ref	0.9984	0.0087
p_aa_pp	0.8074	0.1077
rama_prepro	0.6909	0.3417
omega	0.4984	0.3921
total_score_1b	0.8257	0.7144
fa_atr	0.9092	0.0218
fa_rep	0.4926	0.0342
fa_sol	0.8822	0.0194
fa_elec	0.8156	0.0201
lk_ball_wtd	0.6563	0.0050
hbond_sc	0.4250	0.0107
hbond_bb_sc	0.6218	0.0056
hbond_sr_bb	0.9058	0.0024
hbond_lr_bb	0.7911	0.0049
total_score_2b	0.8326	0.0671

Chapter 5

Discussion

This thesis introduces a model based on Equivariant Graph Neural Network (EGNN) that effectively approximates Rosetta Energy with high accuracy, with 12 out of 18 energy terms achieving Pearson correlation coefficients above 0.7. Compared with the methods mentioned in the introductory chapter (Chapter 1), this method is fast and offers intricate details about residue energies by breaking out various physics-based terms. Such granular information holds the potential to enhance protein structure prediction models, enabling targeted focus on specific energy terms or residues.

Nevertheless, the model exhibits suboptimal performance for certain energy terms, such as omega and hbond, where a predominant number of values are zeros. To rectify this, I recommend refining the loss function parameters and exploring alternative data representations.

Extensive experimentation with various configurations revealed that certain approaches, such as the inclusion of orientation information, the use of Weighted Mean Squared Error Loss, Correlation Loss, AdamW optimizer, and strategies like clamping one-body energy, applying a distance mask for two-body energy, optimizing batch size, adding more linear layers, and separating total score calculation, were advantageous. On the other hand, some tactics did not yield positive results, such as clamping two-body energy or normalizing features within EGNN layers. Fine-tuning of parameters—including dropout rate, layer count, WeightedMSELoss weighting, learning

rate, hidden dimension, and the strategic inclusion of 'bad' decoys—is likely required.

As an alternative to one-hot encoding, future work could explore the use of protein language model embedding such as from Evolutionary Scale Modeling (ESM) [33] to potentially boost model performance. Incorporating edge updates analogous to those used in ProteinMPNN [2] and adopting a protein representation that includes distance, direction, and orientation [20] could also prove beneficial. Additionally, I suggest a weight decay strategy for the combined use of WeightedMSELoss and CorrelationLoss, with an initial focus on the former, gradually shifting towards the latter, as described by the following equations:

$$\omega_{\text{epoch}} = \max(\omega_0 \times (1 - r_{\text{decay}})^{\text{epoch}}, 0) \tag{5.1}$$

$$\text{CombinedLoss} = \omega_{\text{epoch}} \times \text{WeightedMSELoss} + (1 - \omega_{\text{epoch}}) \times \text{CorrelationLoss}$$

The trajectory of this weight change is graphically depicted in Figure 5.1.

Having established the Energy Approximation Network, the next logical step is to integrate it with the IgFold system to demonstrate its utility. The network may serve not only as a specialized loss function but also as a feature in the training of other machine learning models related to proteins. By embedding knowledge from the energy domain, it holds the promise of refining the performance and generalizability of these models.

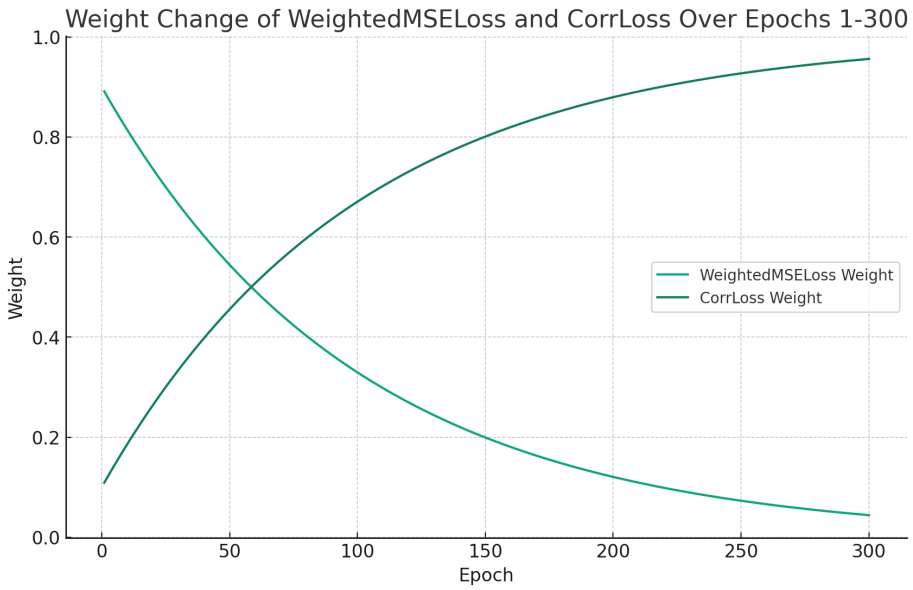


Figure 5.1: Loss Weight Decay Plot

Bibliographic references

1. Jumper, J. *et al.* Highly accurate protein structure prediction with alphafold. *Nature* **596,** 583–589 (2021).

2. Dauparas, J. *et al.* Robust deep learning–based protein sequence design using proteinmpnn. *Science* **378,** 49–56 (2022).

3. Watson, J. L. *et al.* De novo design of protein structure and function with rfdiffusion. *Nature* **620,** 1089–1100 (2023).

4. Ruffolo, J. A., Chu, L.-S., Mahajan, S. P. & Gray, J. J. Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies. *Nature communications* **14,** 2389 (2023).

5. Holzinger, A., Langs, G., Denk, H., Zatloukal, K. & Müller, H. Causability and explainability of artificial intelligence in medicine. *Wiley interdisciplinary reviews: data mining and knowledge discovery* **9,** e1312 (2019).

6. Ovchinnikov, S., Park, H., Kim, D. E., DiMaio, F. & Baker, D. Protein structure prediction using rosetta in casp12. *Proteins: structure, function, and bioinformatics* **86,** 113–121 (2018).

7. Anfinsen, C. B. Principles that govern the folding of protein chains. *Science* **181,** 223–230 (1973).

8. Park, H., Lee, G. R., Kim, D. E., Anishchenko, I., Cong, Q. & Baker, D. High-accuracy refinement using rosetta in casp13. *Proteins: structure, function, and bioinformatics* **87,** 1276–1282 (2019).

9. Sato, K., Akiyama, M. & Sakakibara, Y. Rna secondary structure prediction using deep learning with thermodynamic integration. *Nature communications* **12,** 941 (2021).

10. Fink, T. & Reymond, J.-L. Virtual exploration of the chemical universe up to 11 atoms of c, n, o, f: assembly of 26.4 million structures (110.9 million stereoiso-

mers) and analysis for new ring systems, stereochemistry, physicochemical properties, compound classes, and drug discovery. *Journal of chemical information and modeling* **47,** 342–353 (2007).

11. Mobley, D. L. & Guthrie, J. P. Freesolv: a database of experimental and calculated hydration free energies, with input files. *Journal of computer-aided molecular design* **28,** 711–720 (2014).

12. Tsuboyama, K., Dauparas, J., Chen, J., Laine, E., Mohseni Behbahani, Y., Weinstein, J. J., Mangan, N. M., Ovchinnikov, S. & Rocklin, G. J. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature* **620,** 434–444 (2023).

13. Dieckhaus, H., Brocidiacono, M., Randolph, N. Z. & Kuhlman, B. Transfer learning to leverage larger datasets for improved prediction of protein stability changes. *Proceedings of the national academy of sciences* **121,** e2314853121 (2024).

14. Wang, R., Fang, X., Lu, Y. & Wang, S. The pdbbind database: collection of binding affinities for protein- ligand complexes with known three-dimensional structures. *Journal of medicinal chemistry* **47,** 2977–2980 (2004).

15. Jin, W., Chen, X., Vetticaden, A., Sarzikova, S., Raychowdhury, R., Uhler, C. & Hacohen, N. Dsmbind: se (3) denoising score matching for unsupervised binding energy prediction and nanobody design. *Biorxiv,* 2023–12 (2023).

16. Yang, H., Xiong, Z. & Zonta, F. Construction of a deep neural network energy function for protein physics. *Journal of chemical theory and computation* **18,** 5649–5658 (2022).

17. Zhou, X., Xue, D., Chen, R., Zheng, Z., Wang, L. & Gu, Q. Antigen-specific antibody design via direct energy-based preference optimization. *Arxiv preprint arxiv:2403.16576* (2024).

18. Smith, J. S., Isayev, O. & Roitberg, A. E. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chemical science* **8,** 3192–3203 (2017).

19. Nugmanov, R., Dyubankova, N., Gedich, A. & Wegner, J. K. Bidirectional graphormer for reactivity understanding: neural network trained to reaction atom-to-atom mapping task. *Journal of chemical information and modeling* **62,** 3307–3315 (2022).

20. Ingraham, J., Garg, V., Barzilay, R. & Jaakkola, T. Generative models for graph-based protein design. *Advances in neural information processing systems* **32** (2019).

21. Alford, R. F. *et al.* The rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation* **13,** 3031–3048 (2017).

22. Duval, A. *et al.* A hitchhiker's guide to geometric gnns for 3d atomic systems. *Arxiv preprint arxiv:2312.07511* (2023).

23. Satorras, V. G., Hoogeboom, E. & Welling, M. *E (n) equivariant graph neural networks* in *International conference on machine learning* (2021), 9323–9332.

24. Chaudhury, S., Lyskov, S. & Gray, J. J. Pyrosetta: a script-based interface for implementing molecular modeling algorithms using rosetta. *Bioinformatics* **26,** 689–691 (2010).

25. Varadi, M. *et al.* Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research* **50,** D439–D444 (2022).

26. Barrio-Hernandez, I. *et al.* Clustering predicted structures at the scale of the known protein universe. *Nature* **622,** 637–645 (2023).

27. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. *Arxiv preprint arxiv:1609.02907* (2016).

28. Xu, K., Hu, W., Leskovec, J. & Jegelka, S. How powerful are graph neural networks? *Arxiv preprint arxiv:1810.00826* (2018).

29. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **30** (2017).

30. Wang, S., Sun, S., Li, Z., Zhang, R. & Xu, J. Accurate de novo prediction of protein contact map by ultra-deep learning model. *Plos computational biology* **13,** e1005324 (2017).

31. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. *Arxiv preprint arxiv:1711.05101* (2017).

32. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. *Arxiv preprint arxiv:1412.6980* (2014).

33. Lin, Z. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379,** 1123–1130 (2023).

Appendix A

Result Mining

A.1 Before and after incorporate orientation

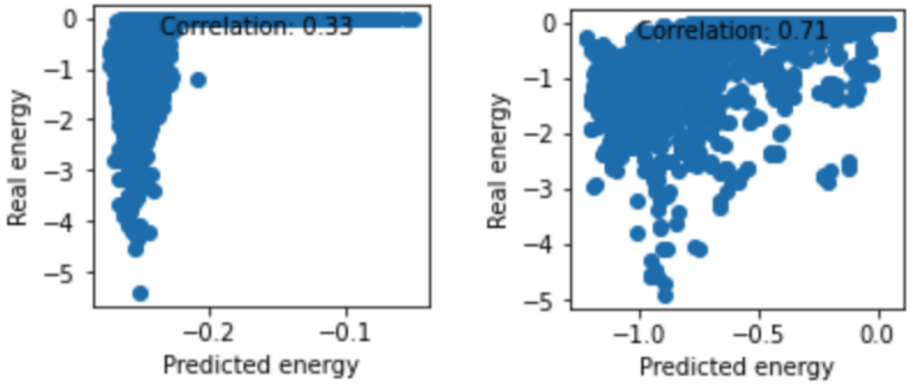
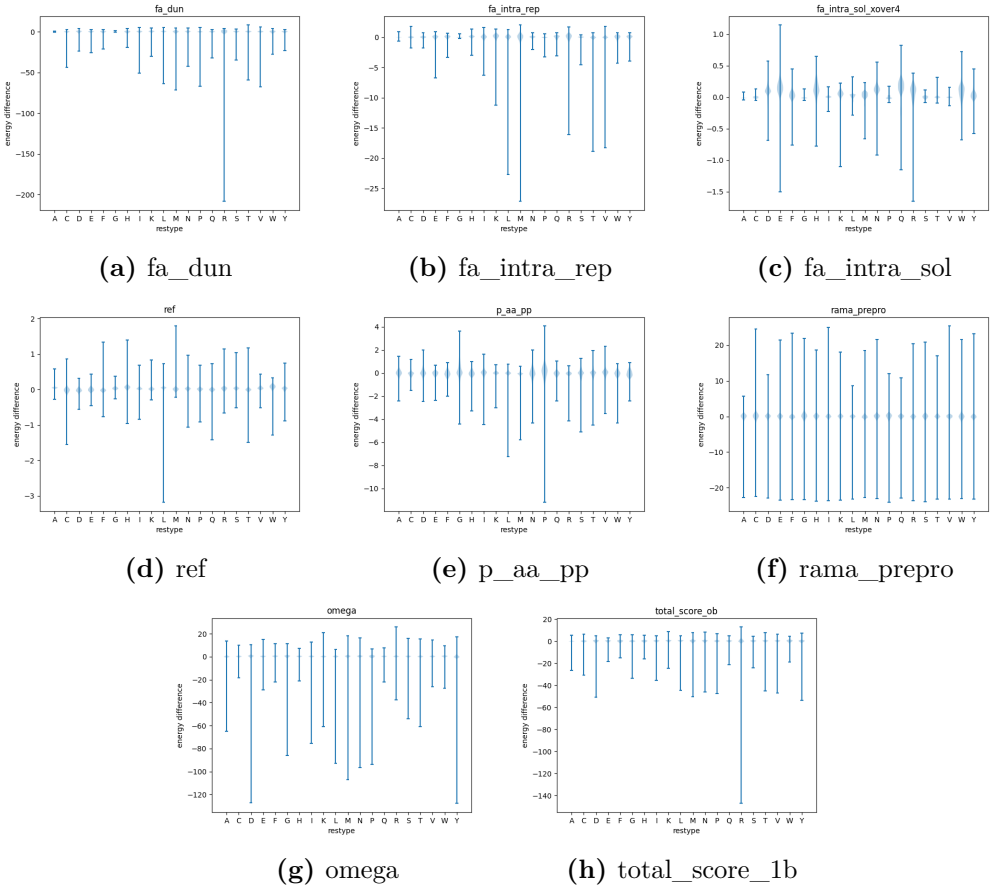


Figure A.1: Result example w/o and w orientation.

A.2 Energy Difference vs Amino Acid Type or Distance

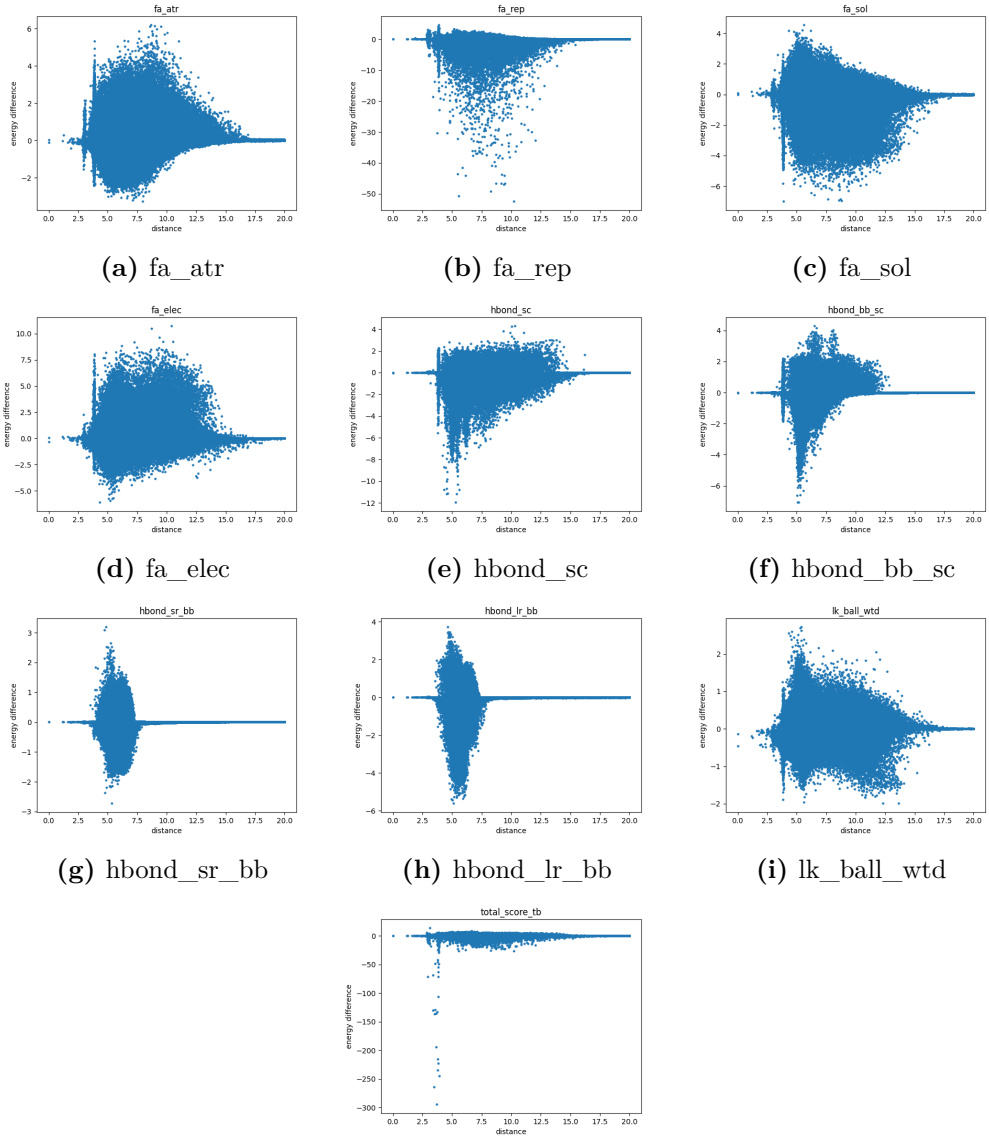


(a) fa_dun **(b)** fa_intra_rep **(c)** fa_intra_sol

(d) ref **(e)** p_aa_pp **(f)** rama_prepro

(g) omega **(h)** total_score_1b

Figure A.2: One-Body Energy Difference vs Amino Acid Type



(a) fa_atr **(b)** fa_rep **(c)** fa_sol

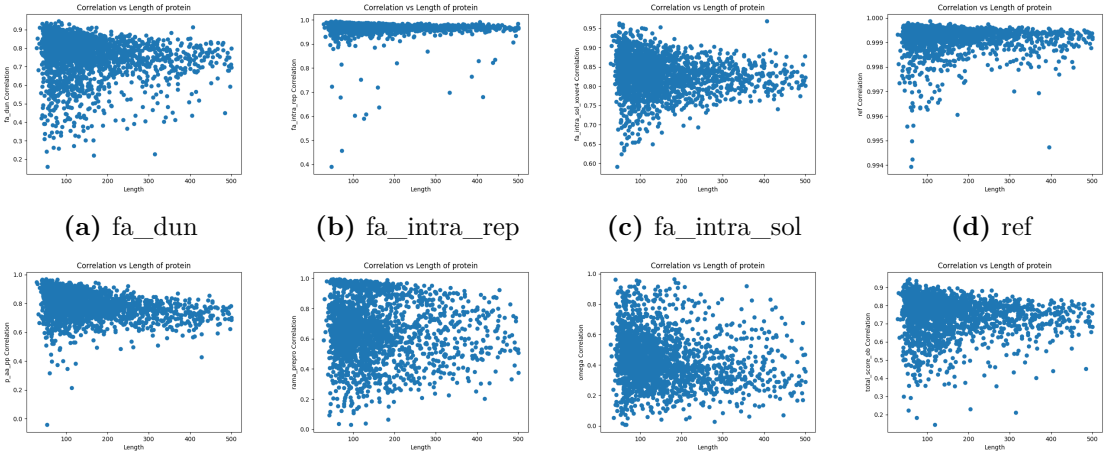
(d) fa_elec **(e)** hbond_sc **(f)** hbond_bb_sc

(g) hbond_sr_bb **(h)** hbond_lr_bb **(i)** lk_ball_wtd

(j) total_score_2b

Figure A.3: Two-Body Energy Prediction Difference vs $C_\alpha - C_\alpha$ Distance.

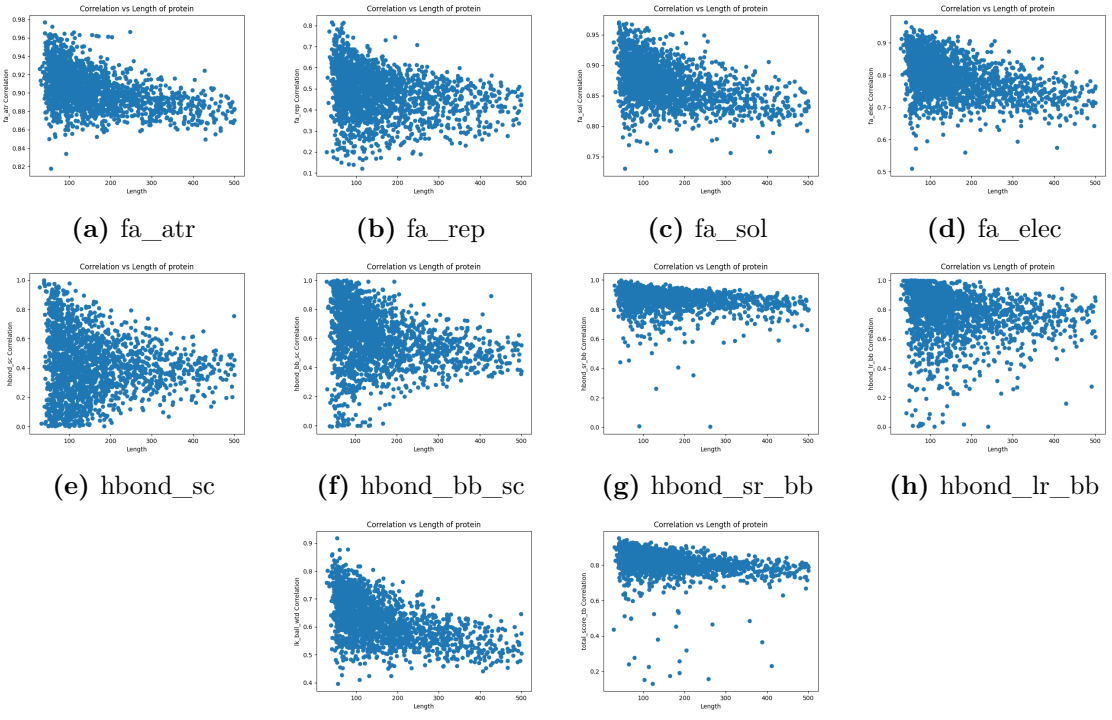
A.3 Error vs residue length



(a) fa_dun **(b)** fa_intra_rep **(c)** fa_intra_sol **(d)** ref

(e) p_aa_pp **(f)** rama_prepro **(g)** omega **(h)** total_score_1b

Figure A.4: One-Body Energy Correlation vs Sequence Length

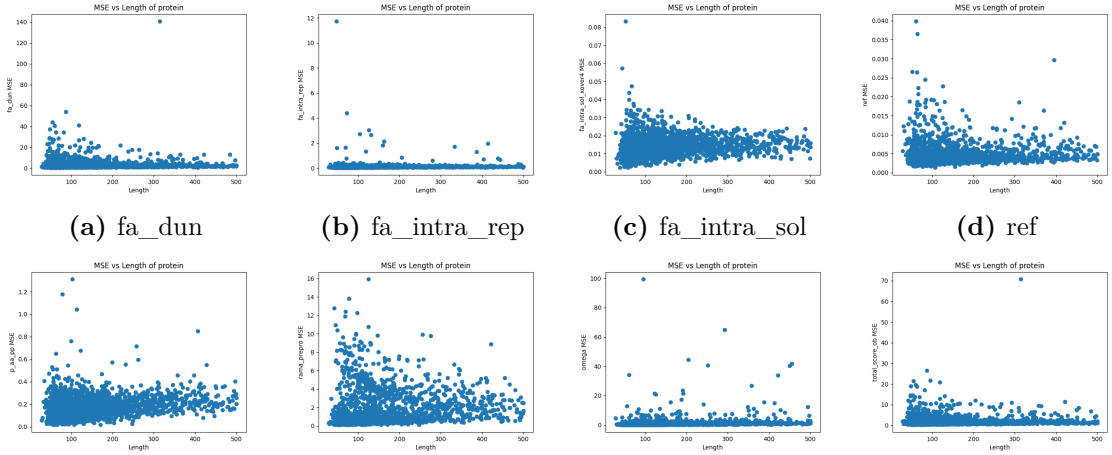


(a) fa_atr **(b)** fa_rep **(c)** fa_sol **(d)** fa_elec

(e) hbond_sc **(f)** hbond_bb_sc **(g)** hbond_sr_bb **(h)** hbond_lr_bb

(i) lk_ball_wtd **(j)** total_score_2b

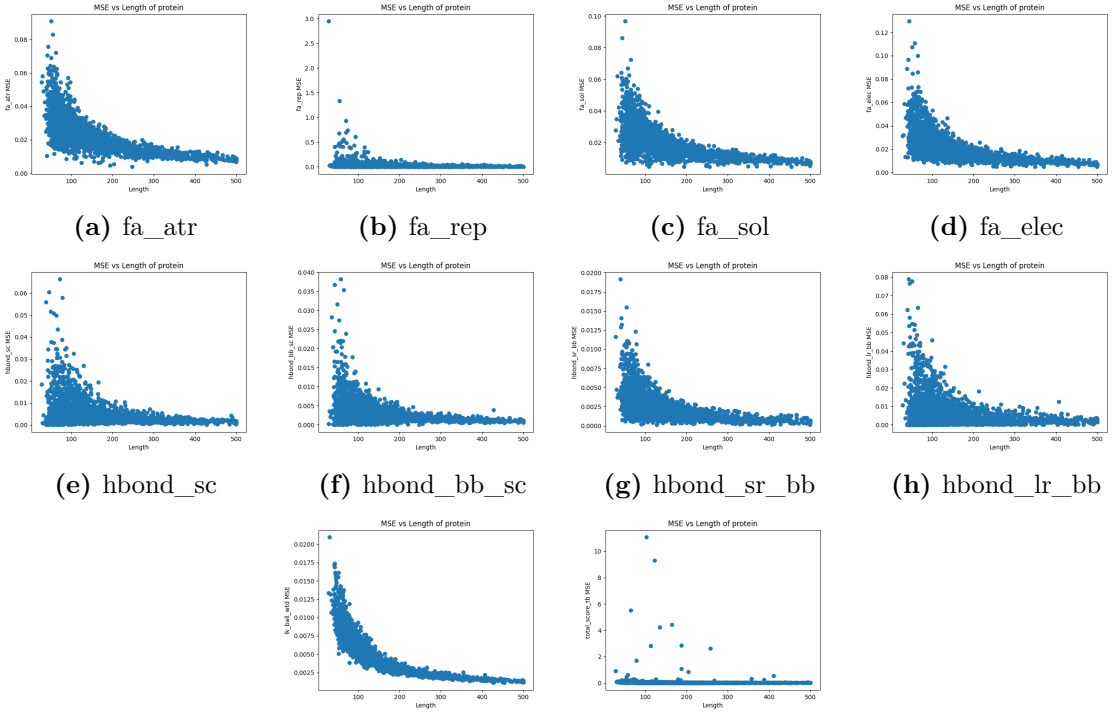
Figure A.5: Two-Body Energy Correlation vs Sequence Length



(a) fa_dun **(b)** fa_intra_rep **(c)** fa_intra_sol **(d)** ref

(e) p_aa_pp **(f)** rama_prepro **(g)** omega **(h)** total_score_1b

Figure A.6: One-Body Energy MSE vs Sequence Length



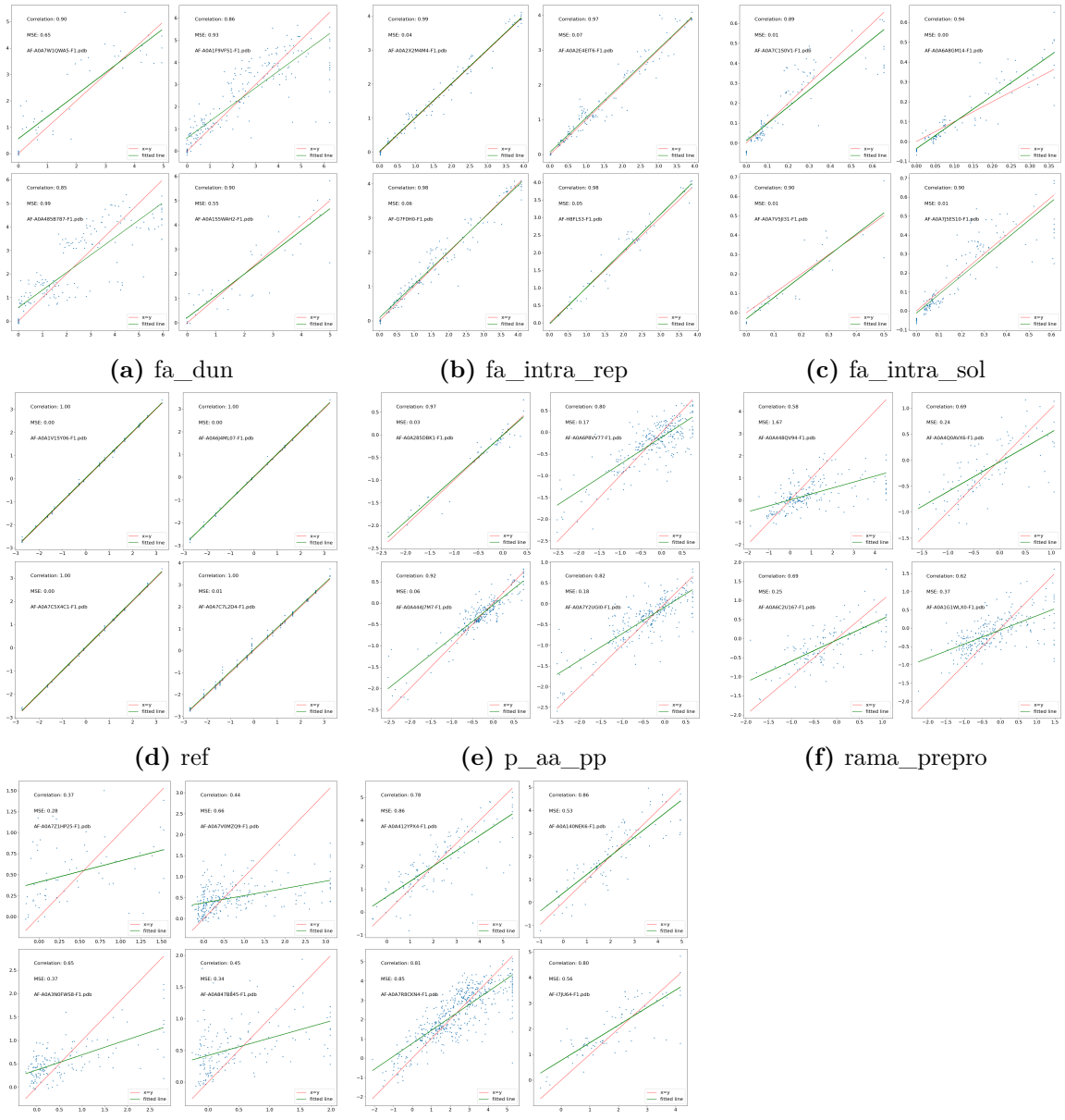
(a) fa_atr **(b)** fa_rep **(c)** fa_sol **(d)** fa_elec

(e) hbond_sc **(f)** hbond_bb_sc **(g)** hbond_sr_bb **(h)** hbond_lr_bb

(i) lk_ball_wtd **(j)** total_score_2b

Figure A.7: Two-Body Energy MSE vs Sequence Length

A.4 Predicted energy vs Rosetta energy of 16 randomly selected test files

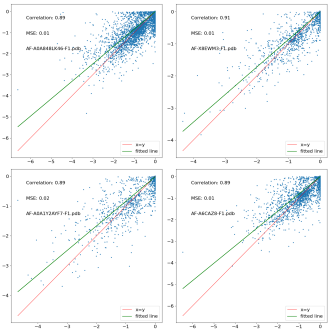


(a) fa_dun **(b)** fa_intra_rep **(c)** fa_intra_sol

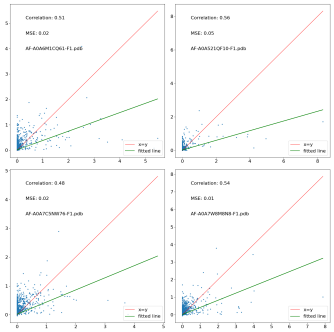
(d) ref **(e)** p_aa_pp **(f)** rama_prepro

(g) omega **(h)** total_score_1b

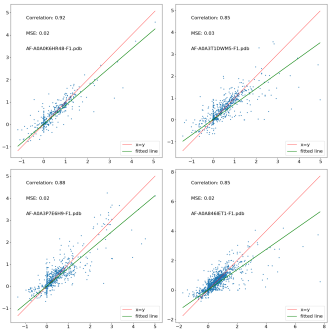
Figure A.8: Predicted One-Body Energy vs Rosetta Energy of 16 randomly selected test files



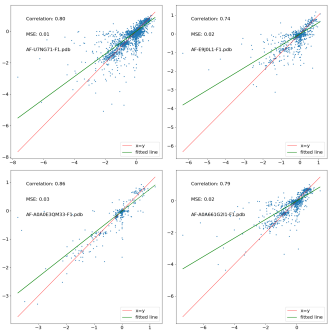
(a) fa_atr



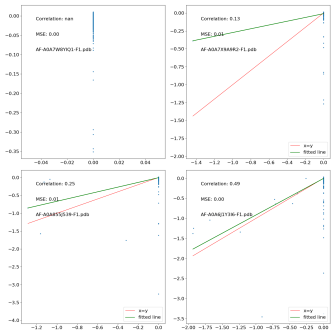
(b) fa_rep



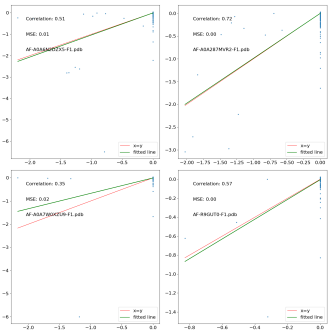
(c) fa_sol



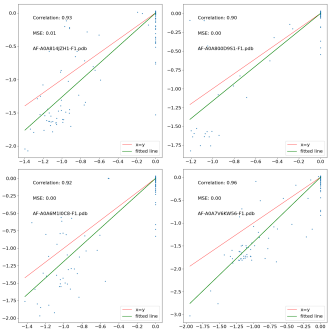
(d) fa_elec



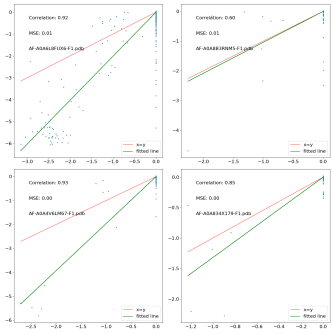
(e) hbond_sc



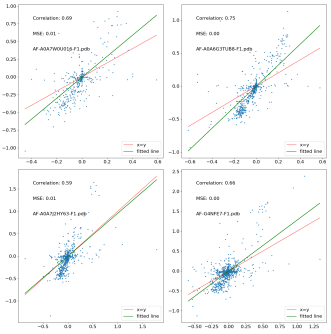
(f) hbond_bb_sc



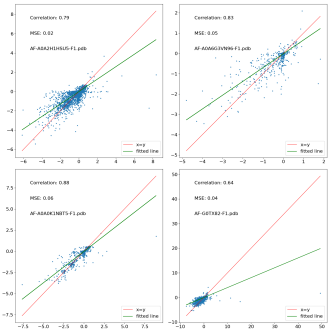
(g) hbond_sr_bb



(h) hbond_lr_bb



(i) lk_ball_wtd



(j) total_score_2b

Figure A.9: Predicted two-body Energy vs Rosetta Energy of 16 randomly selected test files