Article

EGG: Accuracy Estimation of Individual Multimeric Protein Models Using Deep Energy-Based Models and Graph Neural Networks

Andrew Jordan Siciliano 1,†, Chenguang Zhao 2,†, Tong Liu 1,‡ and Zheng Wang 1,*

1 Department of Computer Science, University of Miami, 1365 Memorial Drive, Coral Gables, FL 33124, USA; ajs550@miami.edu (A.J.S.); tong.liu@miami.edu (T.L.)
2 Computer Information Sciences Department, St. Ambrose University, 518 W. Locust Street, Davenport, IA 52803, USA; zhaochenguang@sau.edu
* Correspondence: zheng.wang@miami.edu
† These authors contributed equally to this work.

Abstract: Reliable and accurate methods of estimating the accuracy of predicted protein models are vital to understanding their respective utility. Discerning how the quaternary structure conforms can significantly improve our collective understanding of cell biology, systems biology, disease formation, and disease treatment. Accurately determining the quality of multimeric protein models is still computationally challenging, as the space of possible conformations is significantly larger when proteins form in complex with one another. Here, we present EGG (energy and graph-based architectures) to assess the accuracy of predicted multimeric protein models. We implemented message-passing and transformer layers to infer the overall fold and interface accuracy scores of predicted multimeric protein models. When evaluated with CASP15 targets, our methods achieved promising results against single model predictors: fourth and third place for determining the highest-quality model when estimating overall fold accuracy and overall interface accuracy, respectively, and first place for determining the top three highest quality models when estimating both overall fold accuracy and overall interface accuracy.

Keywords: structural bioinformatics; bioinformatics; neural networks

1. Introduction

The process by which a sequence of amino acids (primary structure) converges to form a three-dimensional (tertiary) structure, known as protein folding, has been of great interest across scientific communities. The three-dimensional structure of proteins is crucial to understanding how they function, interact with, and impact biological systems [1]. Experimental methods (e.g., X-ray Crystallography and Nuclear Magnetic Resonance) of obtaining the native structure of a protein molecule are time-consuming and costly [1,2]. Predicting the tertiary structure from a given primary structure has the potential to positively influence biological and medical research significantly [1,3]. Multiple chains can merge to form a protein complex over inter-subunit interfaces, known as the quaternary structure. Predicting the quaternary structure is becoming increasingly important, as vital cellular processes heavily involve protein complexes [4].

The accuracy estimation of predicted protein models (EMA) is a crucial step for determining their respective utilities. Many types of quality scores, with diverse purposes, can be generated for a given model. Accuracy is defined as the deviance of the predicted model from its native structure. EMA can produce a variety of these quality scores, which help us to determine the strengths and weaknesses of each model, indicating their most effective use case or cases. The Critical Assessment of Protein Structure Prediction 15 (CASP15) experiment exclusively performed EMA upon multimeric protein models.
Deep learning has been used to perform accuracy estimation of both single and multiple-chain (multimers) proteins. Processing not only the individual properties of each residue, but also the interactions between them, is vital to a successful EMA algorithm. Graph neural networks (GNNs) are a pioneering subset of deep learning that takes a graph as input and directly account for the relations between nodes (edges). GNNs allow for arbitrary-sized input graphs, and traditionally output a topologically equivalent graph to the input. Cutting-edge architectures like graph transformers [5,6] process the relationships between nodes (edges) using attention mechanisms [7]. Incorporating attention into message-passing frameworks improves the GNN’s ability to determine the importance of nodes and or edges, given their context, through learned attention coefficients. GNNs in recent years have excelled in regression problems, and have been used in EMA for both single and multiple-chain proteins [8–11].

The success of a deep learning algorithm depends largely upon the training data. Nodes and edges must contain meaningful and indicative feature vectors for GNNs to perform their best. Energy scores, an indicator of molecular stability, are commonly used as input features to deep learning EMA models. It is widely accepted that the native conformations of proteins are associated with points of minimum free energy [12]. Energy functions, which produce energy scores, aim to resemble the free energy quantitatively (e.g., the Rosetta [13] energy function). Energy scores have been used to rank protein models (or complexes), assuming the most accurate models are the most molecularly stable.

Deep neural networks can be trained to emulate an energy function, denoted as energy-based models (EBM). EBMs produce associated energy scores, where smaller values indicate a higher likelihood of the input being correct and larger values indicate a lower likelihood of the input being correct. Correctness is arbitrary, and can have different meanings given different contexts. For example, the correctness of a protein could indicate a high molecular stability, whereas the correctness of an image could indicate the likelihood the image matches a certain label or falls within a specific distribution of images.

Energy-based graph convolutional networks (EGCN) [14] have been deployed for predicting the qualities of protein docking models, a similar task to that of EMA for protein-complex models. EGCN [14] was trained using the absolute difference between the predicted and target (binding energy) values as the loss function. Typically, energy-based models are trained without using the absolute difference or mean squared error loss functions, and instead utilize objective functions that minimize and maximize the output energy given true and adversarial inputs, respectively. Protein-EBM [15] used this approach to deploy and learn state-of-the-art energy-based models directly from protein structures (monomers), using the log-likelihood objective function.

Despite this, to the best of our knowledge, energy-based models have not been used to directly infer the qualities of multimeric protein models. In this paper, we present two approaches that use cutting-edge graph neural networks and novel energy-based models. Graphs are constructed based upon the geometry and topology of the input protein complex structure, where each node represents a residue and edges represent the spatial interactions between them. The EBMs were used to predict the quality of protein complex structures from the perspective of free-energy minimization. Our evaluation results demonstrate significant performance for discerning high- from low-quality computationally predicted multimeric protein models.

2. Results and Discussion

To compare our EBM and GNN models, we computed their respective mean absolute error (L1) and mean squared error (MSE) losses between the predicted and true accuracy scores of the given multimeric protein models, for both overall fold (TM-score) and interface (QS-score) accuracy scores. Both methods achieved comparable results, with only slight differences (see Table 1).
Table 1. Average loss comparison on CASP15 targets.

<table>
<thead>
<tr>
<th>Type</th>
<th>Backbone</th>
<th>TM-L1</th>
<th>TM-MSE</th>
<th>QS-L1</th>
<th>QS-MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>Transformer</td>
<td>0.243</td>
<td>0.094</td>
<td>0.314</td>
<td>0.127</td>
</tr>
<tr>
<td>EBM</td>
<td>Transformer</td>
<td>0.245</td>
<td>0.096</td>
<td>0.341</td>
<td>0.156</td>
</tr>
<tr>
<td>Regression</td>
<td>MetaLayer</td>
<td>0.263</td>
<td>0.098</td>
<td>0.300</td>
<td>0.124</td>
</tr>
<tr>
<td>EBM</td>
<td>MetaLayer</td>
<td>0.242</td>
<td>0.088</td>
<td>0.299</td>
<td>0.126</td>
</tr>
</tbody>
</table>

CASP15 ranking loss [16] is defined as the difference between the true top-ranked model and the top-ranked model determined by a predictor’s method. The group name APOLLO was the original model submitted to CASP15. The group names associated with our presented methods are denoted as EBM–[layer] and Regression–[layer], with the layer being either “Transformer” or “MetaLayer”. Our models were trained as single-model predictors [8,17–19] and, therefore, we filtered all consensus-model predictors.

We found that our methods, on rare occasions, hit a plateau in the energy function, meaning the same prediction is made for all models of a given target. This is not the case for APOLLO, for which this issue was present in many of the CASP15 targets. We slightly modified the CASP15 ranking loss criteria in our blind test of CASP15 targets to address this issue. We only accept a top-ranked model from a given predictor if the predicted quality score is unique for all models in the given target. When these cases occur, we ignore this predictor’s prediction and treat it as if the ranking loss was not less than 0.1. This stricter criteria agrees well with the intention of these ranking metrics, as when multiple protein models are ranked at the highest quality, we cannot determine which to use in the real world. CASP15 evaluators considered all predictions, and treated the first model as the highest predicted quality in these cases, allowing for arbitrary rankings to be introduced.

Our methods, when evaluated against these modified CASP15 ranking loss metrics, generally improved from their predecessor APOLLO (see Figure 1). The performance of our EBMs are comparable to the performance of the transformer backbones, but performed worse compared to the MetaLayer backbones. This suggests EBMs can complement the estimates produced by GNNs, but do not universally match their performance for accuracy estimation tasks. Our EBM–Transformer and Regression–Transformer ranked fifth and fourth, respectively, in the “TM-score ranking loss category”. Our Regression–MetaLayer ranked third in the “QS-score ranking loss category”. For individual target performance metrics with respect to CASP ranking loss, refer to the Supplementary Document (S2).

When analyzing the predicted models, it is vital to holistically view the accuracy estimates across multiple metrics. This reduces the probability of outliers obfuscating the potential utility of a model. To further investigate where our methods performed best, we ranked single model methods again using the CASP ranking loss, but against strictly two-chain and more-than-two-chain targets (see Figure 2). We see our EBM–MetaLayer improve four and seven places when evaluating two-chain compared to more-than-two-chain models for overall fold and interface scores, respectively. For interface scores, our Regression-MetaLayer ranks fourth for two-chain and third for more than two-chain models, suggesting comparable performance independent of the number of chains. When estimating the overall fold accuracy, our Regression and EBM–Transformer architectures moved down in rankings for two-chain targets, suggesting stronger performance when evaluating more-than-two-chain proteins.
Figure 1. The ranking of CASP15 single-model methods based on loss. All 23 two-chain models (dimers) and 17 more-than-two-chain models released by CASP15 were evaluated for a combined total of 40 multimeric protein targets (x-axis) spanning 10,472 predicted models [16]. Loss refers to the CASP15 ranking loss [16] with included modifications to account for non-unique top predicted quality scores.

In addition to these aforementioned metrics, we implemented normalized discounted cumulative gain on the top three (NDCG@3) [20] ranked models per target.

\[
r_i = \begin{cases} 
1 - (Score(M) - Score(m_i))^2 & \text{if } Score(m_i) \text{ is unique} \\
0 & \text{else} 
\end{cases} 
\tag{1}
\]

\[
\text{DCG@3} = \sum_{i=1}^{i=3} \frac{r_i}{\log_2(i+1)} 
\tag{2}
\]

For each position \(i\) in a given ranking, the relevance \(r_i\), defined by Equation (1), is set as one minus the squared difference between the true quality score of the top-ranked model and the predicted score of the \(i\)th model.

(A) TM-score ranking loss

(B) QS-score ranking loss
If a model at position \( i \) does not have a unique quality value, we set \( r_i \) equal to zero. The formula used to compute the DCG@3 for a given target and predictor is defined by Equation (2). We divide the DCG@3 of the predictors by the DCG@3 of the true ranking of the models for normalization. In this way, the maximum achievable NDCG@3 for each target is one, and therefore the maximum total score any method could achieve is forty (one for each target).

Figure 2. Ranking of all CASP15 single-model methods with different numbers of chains based on loss.

Figure 3 ranks the single-based methods upon the sum of their respective NDCG@3 values over all CASP15 targets. Our Regression–MetaLayer ranked first for overall fold accuracy, and our Regression–Transformer ranked first for overall interface accuracy. Furthermore, the majority of our methods (excluding our EBM–MetaLayer) achieved total scores greater than half of the greatest possible NDCG@3 score, across both the overall fold and the interface accuracy rankings. This indicates that the top three ranked models by the majority of our methods often contain suitable choices for higher quality models. For individual target performance metrics with respect to NDCG@3, refer to the Supplementary Document (S2).
We also see our respective methods order predicted models with a higher degree of precision than other top-ranked single model predictors on specific CASP15 targets (see Figures 4 and 5). In Figure 4, the top-ranked model by “Chae Pred” and “MULTICOM_egnn” (C) had a TM-score of ≈0.696. The top-ranked model by our “EBM–Transformer” (B) had a TM-score of ≈0.982. Both models (Figure 4B,C) are aligned and superimposed with the native structure (Figure 4A) using PyMol [21]. In Figure 5, the interface residues are defined as residues within 8Å of any other residue from another chain. Residue positions are set as CB coordinates for all residues, except for GLY, which is set as CA. Both models (Figure 5B,C) are aligned and superimposed with the native structure (Figure 5A) using PyMol [21]. The top-ranked model by “Chae Pred” (Figure 5C) had QS-score ≈0.019. The top-ranked model by our “Regression–MetaLayer” (Figure 5B) had QS-score ≈0.549. It can be seen in Figure 5 that the arrangements of the interfacing residues selected by our methods highly match those of the native structure compared to the top model selected by “Chae Pred”.

Figure 3. Ranking of all CASP15 single-model methods based on NDCG@3.
3. Materials and Methods

3.1. Datasets

Our data comprised of homomer and heteromer models from CASP13-14 experiments, and approximately 10,000 protein complex structures from the protein databank [22], chosen before May of 2022, the start of CASP15. In this way, our methods had access to the same information as groups from CASP15, allowing for fair comparisons between them. Using PyRosetta [23], we further generated examples by performing perturbation upon these structures. Our data were randomly split by associated native pdb_id into training and validation sets of 23,313 and 2,369 examples, respectively. The TM-Score [24] and QS-Best [25] were used as the target values for the overall fold score and overall interface score, respectively.

Each node in an input graph represents a residue in a multimeric protein model. ESM embeddings [26], including their mean, median, standard deviation, and variance, were used as parts of the node features. We also generated 83 additional node features, stemming from a total of six types: one hot encoding of the amino acid, position-specific scoring matrix created using PSI-BLAST [27] from a multiple sequence alignment, normalized Rosetta [13] energy scores, SOV_refine [28] scores for sequence-based and model-based secondary structures and solvent accessibility, fixed sequence-based sinusoidal positional
encodings [7], and MASS [29] protein statistical potentials, including pseudo-bond angle potential, accessible surface potential at the atomic level, sequence separation-dependent potential, contact-dependent potential, relative solvent accessibility potential, and volume-dependent potential.

Edges were created for any residue pairs that have a CB-CB distance ≤ 8Å (CA in the case of Glycine). Ten features were generated for each edge from a total of five types: the distance between the two residues, the angle between the two residues, torsional angles (ω, θ, and φ) between the two residues, contact-dependent potential (CDP), and relative solvent accessibility potential (RSAp). CDP and RSAp are only included if the residue pairs are part of the interface.

3.2. Training Procedure

Energy-based models (EBMs) are a specific class of deep learning architecture inspired by statistical physics that predict a compatibility score (energy) between a given pair of feature and target vectors. EBMs learn a distribution of choices over an input, as opposed to a single output prediction. Consider the following definitions: G = Example Space, X = Feature Space, and Y = Target Space. Given an example pair \((x, y) \in G\), with \(x \in X \& \ y \in Y\), the objective is to maximize \(EBM(x, t)\) when \(t \in Y - \{y\}\) and minimize \(EBM(x, t)\) when \(t = y\). The ideal EBM would hold the following property \(\forall (x, y) \in G:\)

\[
EBM(x, t) = \min \{EBM(x, a) \mid \forall a \in Y\} \iff t = y
\]

During inference, given a fixed feature vector \(x \in X\), the choice of \(t \in Y\) that minimizes the energy \(EBM(x, t)\) is the predicted target for feature vector \(x\).

Training EBMs is computationally expensive, especially for high-dimensional datasets, due to the need to sample and process multiple adversarial inputs per training example. For this reason, we utilized global graph embeddings from pre-trained graph neural networks (GNNs) as input features. The supervised GNNs that produced these embeddings, associated ground truth values, and statistically sampled adversarial target values \((0, 1)\) reside in the space defined in [30]. The EBMs were fed fixed global graph embeddings, as pre-trained, were trained separately from the EBMs using a weighted version of the L1 loss function. For details about training the GNNs, see the Supplementary Document (S1).

Our energy-based models (EBMs) were trained using the loss function defined by Equation (4) for energy-based regression [30]. The EBMs were fed fixed global graph embeddings, associated ground truth values, and statistically sampled adversarial target values \((0, 1)\). Both accuracy scores reside within the space \((0, 1)\) and the target values described in [30] reside in the space \((-\infty, \infty)\). We used the logit function to send accuracy scores to the target space \((-\infty, \infty)\) and the sigmoid function to return logits to the \((0, 1)\) space.

\[
f(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \log \frac{\exp \{f_{b}(x_{i}, y_{i}^{(i,0)})\} - \log(P_{N}(y_{i}^{(i,0)}|y_{i}))}{\sum_{m=0}^{M} \exp \{f_{p}(x_{i}, y_{i}^{(i,m)})\} - \log(P_{N}(y_{i}^{(i,m)}|y_{i}))}
\]

\[
P_{N}(y_{i}|y_{i}) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{N}(y_{i}; y_{i}, \sigma^{2} I)
\]

\[
\{y_{i}^{(i,m)}\}_{m=1}^{M} \sim P_{N}(y_{i}|y_{i})
\]

\[
P_{b}(y) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{N}(y; 0, \beta \sigma^{2} I)
\]

\[
y_{i}^{(i,0)} \triangleq y_{i} + v_{i}, v_{i} \sim P_{b}(y)
\]

In Equation (4), \(f_{b}\) represents the energy produced by our EBM given an input vector \(x_{i}\) and logit \(y_{i}\). We treated \(M - 1\), the number of adversarial (noisy) samples, as a hyper-parameter, and held \(K = 1\) as fixed. Equation (8) defines \(y_{i}^{(i,0)}\), the true target value \(y_{i}\), with additional noise sampled from Equation (7) of intensity \(\beta\), which is a tunable hyper-
The set of adversarial logits is defined by Equation (6). Adversarial logits are sampled from the normal distribution (5) over the \((-\infty, \infty)\) space, with the modification of adding a sliding standard deviation \(\sigma\) to account for asymptotes in the logit function. The third and first quartiles are positioned at approximately 0.675 standard deviations from the mean in a typical normal distribution. We redefine \(\sigma\) as Equation (9), and define the tunable hyper-parameter \(\gamma \in (0, 1)\) that determines \(\sigma\), given a true target value \(y_i = \mu\):

\[
\sigma = \begin{cases} 
\frac{\text{Logit}(\mu+\gamma) - \text{Logit}(\mu)}{0.675} & \text{if } \mu + \gamma < 1 \\
\frac{\text{Logit}(\mu) - \text{Logit}(\mu-\gamma)}{0.675} & \text{else}
\end{cases}
\]  

(9)

The hyper-parameter \(\gamma\) indicates the distance between the quartiles and the mean in the \((0, 1)\) space, and (9) determines the approximate \(\sigma\) to achieve this in the \((-\infty, \infty)\) space. In this way, we hold the normal distributions defined by Equations (5) and (7), which reside in the \((-\infty, \infty)\) space, generally consistent when sent back to the \((0, 1)\) space (using the sigmoid function). Inference through our energy-based models was performed with brute force. Given a fixed global feature vector \(X\), we tested logits associated with evenly distributed values in the range \((0, 1)\), and chose the logit \(Y\) that maximized the negative energy (output of our EBMs) as the estimated accuracy of the input protein structure \(\sim X\).

### 3.3. Model Architectures

All model architectures and their respective hyper-parameters were tuned through SHERPA [31] using a genetic algorithm with a mutation rate of 0.1. Refer to the Supplementary Document (SI) for descriptions of the hyper-parameter search space and objective functions utilized by the genetic algorithm. Our models were trained using adaptive gradient clipping with AutoClip [32].

Our pre-trained models utilized GNN layers; see Figure 6. GNN layers are either a graph transformer layer [5] or a MetaLayer [33] followed by ReLU activation. Graph transformer layers were followed by TopKPooling [34–36]. Global attention pooling [37] was used to produce the global feature vector \(X\). MetaLayer and global attention pooling blocks were composed of fully connected neural networks. A fully connected neural network is used to reduce the global feature vector \(X\) to the final output prediction. Fully connected layers and GNNs were conditionally (determined by the genetic algorithm) followed by normalization layers [38–40].

Our EBMs were trained separately, and took both the fixed global graph embedding \(X\) (from the top-performing pre-trained model) and a logit value \(Y' = \mu\) as input. Both inputs \(X\)
and $Y'$ are concatenated first before being passed into the first fully connected layer (FCL). Following each FCL are dropout, ReLU activation, and concatenating skip connections (represented as dotted lines) from the input logit $Y'$; see Figure 7. We treated the depth of the skip connections as a hyper-parameter. Fully connected layers were conditionally (determined by the genetic algorithm) followed by LayerNorm [39].

![Figure 7.](image)

The top pre-trained models (GNN Backbones) and EBMs were chosen as the models with the lowest value from the objective function used by the genetic algorithm. Refer to the Supplementary Document (S1) for the descriptions and chosen parameters of the architectures associated with our top-performing models. APOLLO, which was the name of our server participating in CASP15, had a similar architecture to that of our current implementations, but was trained on a smaller dataset.

3.4. Implementation

All of our methods were implemented with PyTorch [41] and PyTorch Geometric [42] using NVIDIA A100 GPUs.

4. Conclusions

In this work, we developed multiple graph and energy-based architectures for estimating the accuracy of the predicted multimeric protein structures. This task is computationally demanding, and has crucial implications within the medical and biological research fields. Understanding the energy landscape of a protein is analogous to understanding the ways in which the protein can conform in three dimensional space. In theory, the native conformation of a protein would correspond to its lowest free-energy state. Similarly, quality scores inferred from the protein models are attempting to discern a measure of deviance from its corresponding native structure. This work integrates both of these concepts by learning a viable energy function, using deep neural networks, which can directly infer the qualities of individual multimeric protein models. Furthermore, we capitalized on the power of cutting-edge graph neural network architectures to process spatial and topological information advantageously from input protein structures. Our results against the CASP15 targets support the ability of our methods to generalize and produce high-quality insights into the qualities of predicted multimeric protein models.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/ijms25116250/s1.

Author Contributions: Conceptualization, A.J.S., C.Z., T.L. and Z.W.; methodology, A.J.S., C.Z., T.L. and Z.W.; software, A.J.S. and C.Z.; validation, A.J.S.; resources, A.J.S., C.Z., T.L. and Z.W.; writing—original draft preparation, A.J.S. and C.Z.; writing—review and editing, Z.W. and T.L.; visualization, A.J.S.; supervision, Z.W.; project administration, Z.W.; funding acquisition, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Institute of General Medical Sciences under Award Number R35GM137974 to Z.W.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: We would like to thank Gabriel Studer from the Torsten Schwede group who, upon our request, provided CASP15 customized global score analysis.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations
The following abbreviations are used in this manuscript:

- Multimer: Multiple Chain Protein
- Monomer: Single Chain Protein
- EMA: Model Accuracy Estimation
- CASP: The Critical Assessment of Protein Structure Prediction
- GNN: Graph Neural Network
- FCL: Fully Connected Layer
- EBM: Energy-Based Model
- CB: Beta Carbon
- CA: Alpha Carbon
- CDP: Contact Dependent Potential
- RSAP: Relative Solvent Accessibility Potential
- MSE: Mean Squared Error
- L1: Mean Absolute Error
- TM-score: Overall Fold Score
- QS-score: Overall Interface Score
- DCG: Discounted Cumulative Gain
- NDCG: Normalized Discounted Cumulative Gain

References