Energy-based graph convolutional networks for scoring protein docking models

Yue Cao | Yang Shen

1Department of Electrical and Computer Engineering, Texas A&M University, College Station, Texas
2TEES-AgriLife Center for Bioinformatics and Genomic Systems Engineering, Texas A&M University, College Station, Texas

Correspondence
Yang Shen, Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843.
Email: yshen@tamu.edu

Funding information
National Institutes of Health, Grant/Award Number: R35GM124952

Abstract
Structural information about protein-protein interactions, often missing at the interactome scale, is important for mechanistic understanding of cells and rational discovery of therapeutics. Protein docking provides a computational alternative for such information. However, ranking near-native docked models high among a large number of candidates, often known as the scoring problem, remains a critical challenge. Moreover, estimating model quality, also known as the quality assessment problem, is rarely addressed in protein docking. In this study, the two challenging problems in protein docking are regarded as relative and absolute scoring, respectively, and addressed in one physics-inspired deep learning framework. We represent protein and complex structures as intra- and inter-molecular residue contact graphs with atom-resolution node and edge features. And we propose a novel graph convolutional kernel that aggregates interacting nodes’ features through edges so that generalized interaction energies can be learned directly from 3D data. The resulting energy-based graph convolutional networks (EGCN) with multihead attention are trained to predict intra- and inter-molecular energies, binding affinities, and quality measures (interface RMSD) for encounter complexes. Compared to a state-of-the-art scoring function for model ranking, EGCN significantly improves ranking for a critical assessment of predicted interactions (CAPRI) test set involving homology docking; and is comparable or slightly better for Score_set, a CAPRI benchmark set generated by diverse community-wide docking protocols not known to training data. For Score_set quality assessment, EGCN shows about 27% improvement to our previous efforts. Directly learning from 3D structure data in graph representation, EGCN represents the first successful development of graph convolutional networks for protein docking.

KEYWORDS
energy-based models, graph convolutional networks, machine learning, protein docking, protein-protein interactions, quality estimation, scoring function

1 INTRODUCTION

Protein-protein interactions (PPIs) underlie many important cellular processes. Structural information about these interactions often helps reveal their mechanisms, understand diseases, and develop therapeutics. However, such information is often unavailable at the scale of protein-protein interactomes, which calls for computational protein docking methods. Two major tasks for protein docking are sampling (aiming at generating many near-native models or decoys) and scoring (aiming at identifying those near-natives among a large number of candidate decoys), both of which present tremendous challenges.
This study focuses on scoring by filling two gaps in protein docking. First, current scoring functions for protein docking often aim at relative scoring, in other words, ranking near-natives high. But they do not score the decoy/model quality directly (absolute scoring), which is more often known as quality estimation or quality assessment (without known native structures) in the protein-structure community of CASP. Although quality estimation methods based on machine learning are emerging for single-protein structure prediction, such methods are still rare for protein-complex structure prediction, that is, protein docking. Second, state-of-the-art scoring functions (for relative scoring) in protein docking are often based on machine learning with hand-engineered features, such as physical-energy terms, statistical potentials, and graph kernels. These features, heavily relying on domain expertise, are often not specifically tailored or optimized for scoring purposes. Recently, deep learning has achieved tremendous success in image recognition and natural language processing, largely due to its automated feature/representation learning from raw inputs of image pixels or words. This trend has also rippled to structural bioinformatics and will be briefly reviewed later.

To fill the aforementioned gaps for scoring in protein docking, we propose a deep learning framework with automated feature learning to estimate the quality of protein-docking models (measured by inter-RMSD or iRMSD) and to rank them accordingly. In other words, the framework simultaneously addresses both relative scoring (ranking) and absolute scoring (quality assessment) using features directly learned from data. To that end, our deep learning framework predicts binding free energy values of docking models (encounter complexes) whereas these values are correlated to known binding free energy values of native complexes according to model quality.

Technical challenges remain for the framework of deep learning: how to represent protein-complex structure data and learn from such data effectively? Current deep learning methods in structural bioinformatics often use 3D volumetric representations of molecular structures. For instance, for protein-structure quality assessment, atom density maps have been used as the raw input to 3D Convolutional Neural Networks (CNN). For protein-ligand interaction prediction, the Atomic Convolutional Neural Network (ACNN) uses the neighbor matrix as input and uses radical pooling to simulate the additive pairwise interaction. And for RNA structure QA, grid representations of the structures are used as the input to 3D CNN. However, learning features from volumetric data of molecular structures present several drawbacks. First, representing the volumetric input data as pixel data through tensors would require discretization, which may lose some biologically meaningful features while costing time. Second, such input data are often sparse, resulting in many convolutional operations for zero-valued pixels and thus low efficiency. Third, the convolutional operation is not rotation-invariant, which demands rotational augmentation of training data and increased computational cost.

To effectively learn features from and predict labels for protein-docking structure models, we represent proteins and protein-complexes as intra- and inter-molecular residue contact graphs with atom-resolution node and edge features. Such a representation naturally captures the spatial relationship of protein-complex structures while overcoming the aforementioned drawbacks of learning from volumetric data. Moreover, we learn from such graph data by proposing a physics-inspired graph convolutional kernel that pools interacting nodes’ interactions through their node and edge features. We use two resulting energy-based graph convolutional networks (EGCN) of the same architecture but different parameters to predict intra- and inter-molecular energy potentials for protein-docking models (encounter complexes), which further predicts these encounter complexes’ binding affinity and quality (iRMSD) values. Therefore, our EGCN is capable of both model ranking and quality estimation (or quality assessment without known native structures). We note the recent surging of graph neural networks in protein modeling, such as protein interface prediction and protein structure classification. We also note that this is the first study of graph neural networks for protein docking.

The rest of the article is organized as follows. In Section 2, we first describe the data sets for training, validating, and testing our EGCN models. We proceed to introduce contact graph representations for proteins and protein complexes and initial features for nodes and edges; our energy-based graph convolutional networks as the intra- or inter-molecular energy predictor for encounter complexes; and additional contents (label, loss function, and optimization) for training our EGCN. Next we compare in Section 4 EGCN’s performances against a state-of-the-art scoring function for ranking and against our previous efforts for both ranking and quality estimation. We also share in Section 5 our thoughts on how EGCN can be further improved in prediction accuracy and training efficiency. Lastly, we summarize major contributions to method development and major results of method performances in Section 6.

2 | MATERIALS AND METHODS

2.1 | Data

2.1.1 | Training and validation sets

We randomly choose 50 protein-protein pairs from protein benchmark 4.0, all with known binding affinities (kd) values, as in our earlier work and split them into training and validation pairs with the ratio 4:1. For each training or validation pair, we perform rigid docking using ClusPro and retain the top-1000 decoys according to ClusPro’s default scoring, and introduce flexible perturbation for each decoy using nMA. In total, we have 40 000 training and 10 000 validation decoys.

2.1.2 | Test sets

We consider three test sets of increasing difficulty levels.
1. The first test set includes the rest 107 protein pairs from the protein benchmark set 4.0.24 As in the training and validation sets, these unbound protein pairs are rigidly docked through ClusPro. The top 1000 decoys after scoring are flexibly perturbed through cNMA, leading to 107 000 decoys.

2. The second test set includes 14 recent critical assessment of predicted interactions (CAPRI) targets30 undergoing the same protocol above (ClusPro + cNMA) for 14 000 decoys. Besides unbound docking seen in the benchmark test set, many of the targets here involve homology-unbound and homology-homology docking.29 The homology models were automatically built using the webserver iTASSER when CARPI targets were released to predictors, with few exceptions detailed in our previous study.29

3. The last test set is Score_set, a CAPRI benchmark for scoring.31 It includes 13 earlier CAPRI targets, each of which has 400 to 2000 flexible decoys generated through various protocols by the community.

We summarize PDB IDs for training, validation, and the three test sets in Tables S1-S5. Binding affinities for some of the benchmark test set and nearly all of the CAPRI test set and Score_set (marked with stars in the supplementary tables) were not available and instead predicted from protein sequences.32

2.2 | Graphs and features

The structure of a protein or protein complex is represented as a graph $G$ where each residue corresponds to a node and each intra- or inter-molecular residue-pair defines an edge. Such intra- and inter-molecular contact graphs can be united under bipartite graphs: The two sets of nodes correspond to the same protein for intramolecular graphs and they do to binding partners for intermolecular graphs. Atom-resolution features are further introduced for nodes and edges of such bipartite contact graphs. In other words, to represent $G$, we have initialized two node feature matrices (denoted by $X_A \in \mathbb{R}^{N_1 \times M}$ and $X_B \in \mathbb{R}^{N_2 \times M}$) and an edge feature tensor (denoted by $A \in \mathbb{R}^{N_1 \times N_2 \times K}$), where $N_1$ or $N_2$ denote the number of residues for either protein $A$ or $B$ (which can be the same protein in the case of intramolecular contact graphs), $M$ the number of features per node, and $K$ the number of features per edge. As revealed later in our graph convolutional networks, node feature matrices will be learned, that is, updated layer after layer, by interacting with neighboring nodes’ features through the fixed edge tensors.

For node-feature matrix $X$, we use $M = 4$ features for each node. Each side chain is represented as a pseudo atom whose position is the geometric center of the side chain. And the first three node features are the side-chain pseudo atom’s charge, nonbonded radii, and geometric center of the side chain. And the first three node features are set at 0 for pairs involving prolines (without HN).

For edge-feature tensor $A$, we use $K = 11$ features for each edge. These features are related to pairwise atomic distances between the two corresponding residues. Specifically, each residue is a point cloud of six atoms, including five backbone atoms (N, HN, CA, C, and O as named in CHARMM27) and one side-chain pseudo atom (named SC). For numerical efficiency, we have picked 11 pairwise distances (including potential hydrogen bonds) out of the total $6 \times 6 = 36$ (see Table 1) and used their reciprocals for the edge features. We use a cutoff of 12 Å for pairwise atomic distances and set edge features at zero when corresponding distances are above the cutoff.

In this study, in order to address input proteins of varying sizes, we consider the maximum number of residues to be 500 and hence fix $N$ at 500 for contact graphs. Correspondingly, smaller proteins’ feature matrices/tensors $X$ and $A$ are zero-padded as needed so that feature matrices/tensors for all input data are of the same sizes: $X \in \mathbb{R}^{N \times M}$ and $A \in \mathbb{R}^{N \times N \times K}$.

2.3 | Energy-based graph convolutional networks

2.3.1 | Background on principle-driven energy model

To score protein-docking models, we try to use machine learning to model $\Delta G$, the binding energy of encounter complexes, which can be written as:

$$\Delta G = G_C - G_{R_u} - G_{L_u}$$

where $G_C$, $G_{R_u}$, and $G_{L_u}$ denote the Gibbs free energies of the complex, unbound receptor and unbound ligand, respectively. For $G_C$, we can further decompose it into the intramolecular energies within two individual proteins and the intermolecular energy between two proteins:

### TABLE 1  The inter-residue atom pairs whose distances are converted to edge features

| Index | Atom 1 | Atom 2 |
|-------|--------|--------|
| 1     | SC     | SC     |
| 2     | SC     | O      |
| 3     | SC     | N      |
| 4     | O      | SC     |
| 5     | O      | O      |
| 6     | O      | N      |
| 7     | N      | SC     |
| 8     | N      | O      |
| 9     | N      | N      |
| 10    | HN     | O      |
| 11    | O      | HN     |

Note: Atom names follow the convention in CHARMM, except that “SC” corresponds to pseudo-atoms for side chains. Note that the last two features are set at 0 for pairs involving prolines (without HN).
where $G_R$, $G_L$, $G_{RL}$ are the Gibbs free energies within the encountered receptor, within the encountered ligand and between them, respectively. Therefore, the binding energy in classical force fields can be written as:

$$\Delta G = G_R + G_L - G_{Ru} - G_{Lu} + G_{RL}$$

In flexible docking, unbound and encountered structures of the same protein are often different. Such protein conformational changes indicate that $G_R - G_{Ru} \neq 0$ and $G_L - G_{Lu} \neq 0$.

### 2.3.2 Extension to data-driven energy model

It is noteworthy that the expression of $\Delta G$ above consists of four terms measuring intramolecular free energies of individual proteins and one term measuring intermolecular free energies across two proteins. Therefore, we decide to use two machine-learning models $f$ of the same neural-network architecture but different parameters to approximate the two types of energy terms as follows:

$$\Delta G = f_R(X_R) + f_R(X_L) + f_R(X_{Ru}, A_{Ru}),$$

where $f_R$ and $f_R$ are the intra- and inter-molecular energy models (graph convolutional networks here) parameterized by $\theta$ and $\theta$, respectively. Subscripts are included for node feature matrices $X$ or edge feature matrices $A$ to indicate identities of molecules or molecular pairs. The parameters are to be learned from data specifically for the purpose of quality estimation.

The architecture of the neural networks is summarized in Figure 1 whereas individual components are detailed later.

#### 2.3.3 Energy-based graph convolutional (EGC) layer

The node matrices ($X_A$ and $X_B$) and the edge tensor $A$ of each contact graph are first fed to three consecutive graph convolutional layers inspired by physics. Specifically, energy potentials are often pairwise additive and those between atoms ($i$ and $j$) are often of the form $\frac{F_{ij}}{r_{ij}}$, where $x_i$ and $x_j$ are atom "features," $r$ is the distance between them, and $a$ is an integer. For instance, $x$ being a scalar charge and $a$ being 1 lead to a Coulombic potential.

Inspired by the energy form, we propose a novel graph convolutional kernel that "pools" neighbor features to update protein A's node features layer by layer:

$$x^{(l+1)}_{ip} = \text{LeakyReLU} \left( \sum_{j=1}^{K} \sum_{k=1}^{M} \left( W^{(l+1)}_{kp} x^{(l)}_j \right)^T A_{ik} \left( W^{(l+1)}_{kp} x^{(l)}_j \right) \right),$$

where subscripts $i$ and $j$ are node indices for proteins A and B, respectively, $p$ node-feature index, and $k$ edge-feature index; superscripts $(l)$ and $(l+1)$ indicate layer indices. Accordingly, $x^{(l)}_j$ of size $M^{(l)} \times 1$ is the node feature vector for the $ith$ node of protein A in layer $l$, $x^{(l+1)}_{ip}$ is the $p$th node-feature for the $ith$ node of protein A in layer $(l+1)$ and $W^{(l+1)}_{kp}$ of size $10 \times M^{(l)}$ is the trainable weight matrix for the $k$th edge feature and the $p$th node feature ($p = 1, ..., M^{(l+1)}$) in the same layer $(l+1)$. In this way we sum all the interactions between node $i$ in protein A and all neighboring nodes $j$ in protein B through the $K = 11$ edge features. Similarly we update node features for each node in protein B by following

![Figure 1](image_url)  

**Figure 1** The architecture of the proposed graph convolutional network (GCN) models for intra- or inter-molecular energies. In our work, there are five types of such models together for predicting encounter-complex binding energy, including four intramolecular models with shared parameters for the unbound or encountered receptor or ligand as well as one intermolecular model for the encounter complex. In each type of model, the inputs (to the left of the arrow) include a pair of node-feature matrices ($X_A$ and $X_B$) for individual protein(s) and an edge-feature tensor, $A$, for intra- or inter-molecular contacts. And the inputs are fed through three layers of our energy-based graph convolution layers that learn from training data to aggregate and transform atomic interactions, followed by multihead attention module and fully connected layers for the output of intra- or inter-molecular energy.
\[ x_{lp}^{(l+1)} = \text{LeakyReLU} \left( \sum_{i=1}^{K} \sum_{k=1}^{L} (W_{lp}^{(l+1)} x_i^{(l)}) \right) \]

where \( \hat{A}_{ik} \) is an element of \( \hat{A} \), a permuted \( A \) with the first two dimensions swapped. When calculating intramolecular energy within a single protein, the second molecule’s feature update can be skipped for numerical efficiency.

We only do the convolution on the node features, while keeping the edge features the same across all the EGC layers. The output node-feature matrices \( X_{L}^{(l+1)} \) and \( X_{\theta}^{(l+1)} \) for both proteins, together with the edge feature tensor \( A \), will be used as the input (after batch normalization) for the next layer \((l+1)\). Whereas \( M^{(0)} = M = 4 \), we choose \( M^{(1)} = 2 \) and \( M^{(2)} = M^{(3)} = 5 \) without particular optimization.

### 2.3.4 Multihead attention and fully connected layers

After the three EGC layers the output node feature matrices \( X_{p}^{(l)} \) and \( X_{\theta}^{(l)} \) \((L = 3 \text{ in this study})\) for both proteins (or two copies of the single matrix in the intramolecular case) are concatenated and fed into a multihead attention module,\(^{16,35}\) whose output subsequently goes through three fully connected (FC) layers with 128, 64, and 1 output, respectively. A dropout rate of 20% is applied to all but the last FC layer.

### 2.3.5 Label, loss function, and model training

The label here is the binding energy \( \Delta G \) although it is not available for encounter complexes. We thus use the same idea as in our previous work.\(^{10}\) Specifically, using known \( k_d^{25-27} \) binding affinity of native complexes, or their values predicted from protein sequences,\(^{32}\) we estimate \( k_d' \), the binding affinity of an encounter complex, to weaken with the worsening quality of the encounter complex (measured by iRMSD):

\[ k_d' = k_d \cdot \exp(\alpha \cdot (\text{iRMSD})^q) \]

where \( \alpha \) and \( q \) are hyperparameters optimized using the validation set. Specifically, \( \alpha \) is searched between 0.5 and 10 with a step size of 0.5 and \( q \) among 0.25, 0.5, 0.75, 1, 1.5, and 2. The optimized \( \alpha \) and \( q \) are 1.5 and 0.5, respectively. Therefore, for each sample (corresponding to an encounter complex or decoy), the predicted label is the previously defined \( \Delta G \), the actual label is \( RT \ln (k_d') = RT \ln (k_d) + RT \alpha \cdot (\text{iRMSD})^q \), and the error is simply the difference between the two.

We learn parameters \( \theta \) and \( \theta' \) of two graph convolutional networks of the same architecture, by minimizing the loss function of mean squared errors (MSE) over samples. Model parameters include \( W_{lp}^{(l)} \) in the EGC layers as well as those in the multihead attention module and FC layers. One set of learned parameter values, \( \theta \) are shared among all intramolecular energy models, and the other set, \( \theta' \) are for the intermolecular energy model.

The model is trained for 50 epochs through the optimizer Adam\(^{36}\) with a batch size of 16 and the learning rate is tuned to be 0.01 using the validation set. The training and validation losses over the 50 epochs are shown in Figure S1.

Note that all training and validation cases, but not all test cases, are with known \( k_d \) values.\(^{25-27}\) Those test cases with unknown \( k_d \) values instead used sequence-predicted values.\(^{32}\) Although the accuracy of affinity prediction for these cases is not available without ground truth, the original study\(^{32}\) reported a correlation range of 0.836 to 0.998 and 0.739 to 0.992 for their self-consistency and jack-knife test sets, respectively. More importantly, the accuracy of affinity prediction only affects absolute scoring but does not do so for relative scoring (as a constant additive error would be introduced to all structure models of the same target).

### 2.4 Assessment metrics

We are aiming at not only relative scoring (ranking) but also absolute scoring (quality estimation) of protein-docking models. Two sets of assessment metrics are therefore introduced.

For ranking models for a given protein-protein pair, we use enrichment factor which is the number of acceptable protein-docking models among the top \( P\% \) ranked by a scoring function divided by that ranked by random. Conceptually, enrichment factor measures the fold improvement of ranking acceptable models high relative to random ranking. Here acceptable models are defined according to the \( \text{CAPRI criteria}^{37} \) with iRMSD within 4 Å (ligand RMSD or native contacts were not considered).

In addition, for all targets in each given test set, we use the number of targets with at least acceptable docking-models (including that of medium or high-quality models according to the \( \text{CAPRI iRMSD criteria}^{37} \)) when such models were among top 1, 5, or 10 ranked by a scoring function. To remove redundancy among decoys, we clustered all decoys of each target using the program FCC (Fraction of Common Contacts)\(^{38}\) with default parameters and only passed those FCC-retained cluster representatives to each scoring function.

For quality estimation of all docking models in various quality ranges, we use the root mean square error (RMSE) between the real iRMSD and our predicted iRMSD (as obtained from our predicted labels). Conceptually, RMSE measures the proximity between predicted and actual quality measures.

### 2.5 Baseline methods

We use two baseline scoring functions here. The first one, only applicable to model ranking, is integration of residue- and atom-based potentials for docking (IRAD).\(^{12}\) One of the top-performing scorers in recent CAPRI. IRAD uses a linear combination of atom-based ZRANK potentials (van der Waals, electrostatics, and ACE desolvation contact
energy) and five residue-based potentials. It optimizes the weights of these potentials through optimizing hit rates for ZDOCK 3.0 decoys generated for docking benchmark 3.0.12

The second one is a Random Forest (RF) model from our previous work,10 which both ranks models and estimates model quality. For comparability among models, the RF model is retrained using the same training and validation sets in this study. Similar to IRAD, RF is a combination of eight potentials, although all atom-based and calculated in the CHARMM27 force field, namely bond, angle, dihedral and Urey-Bradley terms of internal energy, van der Waals, nonpolar contribution of the solvation energy based on solvent-accessible surface area (SASA), and two solvation energy terms based on Generalized Born with a simple SWitching (GBSW). Unlike IRAD, RF combines the eight atom-based potentials in random forest. More importantly, RF is trained to optimize binding-energy estimation and absolute scoring for encounter complexes in our data set and can be used (although not optimized) to rank models using predicted absolute scores ([relative] binding affinity or iRMSD).

Just like RF, EGCN is trained to optimize absolute scoring and can be applied to (although not optimized to) relative scoring. However, unlike both baselines where classical energy terms from various force fields are calculated for structure models and then combined for scoring these models, EGCN directly learns energies from structure data to optimize energy estimation and absolute scoring. Representing 3D structure data as spatial graphs, EGCN uses physics-inspired, energy-based graph convolutional networks newly developed for this purpose, as detailed earlier.

3 | RESULTS

3.1 | Relative scoring (model ranking)

We first analyze our model’s performances of relative scoring and compare them to IRAD. Note that real affinity values are essentially not needed for EGCN just for relative scoring and errors in affinity prediction would not affect EGCN’s ranking results. For both benchmark and CAPRI test sets, we note that the proposed EGCN significantly outperforms both RF and IRAD (Figure 2A,B). In particular, for the top-ranked 10% for the benchmark test set, EGCN achieved an average enrichment factor of 3.5, which represents a nearly 40% improvement compared to RF (2.6) and nearly 120% improvement against IRAD (1.6). In all cases, the improvement margins decrease as the top-ranked percentages increase (i.e., more models are retained), due to the fact that limited and relatively few acceptable models are available among the decoys. Although the training/validation set and the benchmark test set involve unbound docking, the CAPRI test set additionally involves the more challenging case of homology docking. Nevertheless, for the more challenging CAPRI test set, EGCN’s average enrichment factor only slightly dropped from 3.5 to 3.0 in the top-ranked 10% and did so from 3.1 to 2.3 in the top-ranked 20%.

For Score_set, the CAPRI historical benchmark set, EGCN outperforms RF as shown in Figure 2C. Its average performance is comparable to IRAD although it has considerably lower variance across targets (Figure 2C). It is noteworthy that, although both the benchmark and CAPRI test decoys are generated by the same protocol (ClusPro + cNMA) as the training/validation decoys, Score_set represents completely different and heterogeneous decoy-generating protocols from the community. Impressively, for the top-ranked 10%, EGCN’s average enrichment factor is almost the same (nearly 3.0) for Score_set as it is for the CAPRI test set. For the top-ranked 20%, the factor slightly decreased to 2.1 for Score_set compared to 2.3 for the CAPRI test set.

We have also evaluated relative scoring performances using the CAPRI iRMSD criteria37 and compared IRAD, RF, and EGCN in Table 2. For both the benchmark and CAPRI test sets, EGCN significantly outperformed IRAD. First, for the benchmark test set of 107 cases, EGCN generated 46 targets with at least acceptable top-5 predictions (the most possible number being 70 for the decoy set), representing more than 50% increase compared to 30 achieved by IRAD. The performance improvement for ranking medium and high-quality predictions was even more impressive: EGCN generated 21 and 5 targets with medium and high-quality top-5 predictions, respectively, representing 62% and 150% increases compared to IRAD. Second, for the CAPRI test set, although the total number of targets (14) can be too few to provide statistical significance, EGCN significantly outperforms both RF and IRAD (Figure 2A,B).
again increased the number of at least acceptable, medium, and high-quality top-5 models from IRAD’s 4, 2, 0 to 6, 3, 2, respectively. Last, even for Score_set where test decoys were generated by diverse protocols different from EGCN training decoys, EGCN had comparable or slightly better ranking performances, producing six targets with at least acceptable top-5 predictions (including four targets with medium and one with high-quality predictions). We find the EGCN ranking performance particularly impressive considering that, unlike IRAD, the current EGCN was trained for absolute scoring and not optimized for relative scoring. More thoughts to further improve its ranking performance will be described in Section 5.

### 3.2 Absolute scoring (quality estimation)

We next analyze EGCN’s absolute scoring performances and compare them to our previous RF model. IRAD, among many other scoring functions, is only for relative scoring or ranking and thus not compared here. EGCN significantly outperforms RF in quality estimation across all test sets (Figure 3). For the benchmark test set, EGCN estimates all docking models’ interface RMSD values with an error of 1.32 Å, 1.43 Å, and 1.51 Å when the models’ iRMSD values are within 4 Å (acceptable), between 4 Å and 7 Å, and between 7 Å and 10 Å, respectively. These values represent 14%-22% improvement against RF’s iRMSD prediction errors. Both EGCN and RF’s prediction errors remain relatively flat when models are acceptable or close with iRMSD values within 10 Å whereas they rise out the range when precise quality estimation is no longer desired for those far incorrect models. This performance trend is partially by chance, reflecting the quality distribution in the training set. But it can be guaranteed by design through reweighting training decoys of different quality ranges, as we did before.

Very similar performances and trends are observed for the CAPRI test set and the CAPRI Score_set, EGCN’s quality estimation performances only deteriorate slightly in the more challenging CAPRI test set involving homology docking: an error of 1.42 Å, 1.51 Å, and 1.80...
when the models’ iRMSD values are within 4 Å, between 4 Å and 7 Å, and between 7 Å and 10 Å, respectively. For Score_set involving diverse and distinct decoy generation protocols, EGCN’s quality-estimation performances further deteriorate slightly to an error of 1.99 Å, 2.22 Å, and 2.61 Å when the models’ iRMSD values are within 4 Å, between 4 Å and 7 Å, and between 7 Å and 10 Å, respectively, which has shown even more improvements (27%-28%) relative to RF. It is noteworthy that these two test sets are more challenging than the benchmark test set for additional reasons. Unlike the case of relative scoring, binding affinities of native complexes are needed for absolute scoring. But their values have to be predicted for all but one target pairs in these two test sets (13 out of 14 or 12 out of 13), compared to just 34 out of 107 for the benchmark test set.

To further assess the impact of affinity prediction on absolute scoring, we split the benchmark test set into two subsets (73 with known affinity values and 34 without) and analyzed EGCN on these two subsets separately (Figure S2). Indeed, the subset with known affinity values had seen lower errors in iRMSD prediction. Specifically, the RMSE values for decoy models within 4 Å, between 4 Å and 7 Å, and between 7 Å and 10 Å were 1.20 Å, 1.22 Å, and 1.50 Å, respectively, when affinity values were known, compared to 1.52 Å, 1.73 Å, and 1.53 Å, respectively, when affinity values were predicted from sequences. The deterioration in iRMSD prediction due to the unavailable affinity values did not appear to be a major concern, although its sensitivity to the accuracy of affinity prediction is unclear due to the lack of ground-truth affinity values.

Taken together, these results suggest that, learning energy models directly from structures as in EGCN represents a much more accurate and robust data-driven approach than doing so from structure-derived energy terms as in RF. The EGCN model performances are less sensitive to target difficulty than they are to the training vs. test data distributions (reflected in the ways training/test decoys are generated).

4 | DISCUSSION

The accuracy of the EGCN model could be further improved along a few directions. The first and the foremost important direction is to provide better-quality training data whose distributions align better to those of the test data. Specifically in our case, this demands more diverse protocols to generate training decoys (as opposed to the only ClusPro + cNMA protocol used in the study). Moreover, decoys in different quality ranges could be given different priorities as well. The second direction is to train deeper models (going beyond 4 Å when the models are trained). In this study) and to include more edge features (for instance, including distances besides their reciprocals so that internal energies polynomial in distances can be captured). The third direction is to split the two objectives of relative and absolute ranking. We are training just one model for both model ranking and quality estimation whereas the loss function is perfectly aligned to quality estimation alone. Instead, a separate model with its own loss function can be trained just for model ranking.

The computational cost of training the EGCN model can also be reduced. Currently, we use an edge feature tensor $A \in \mathbb{R}^{N \times N \times K}$ and pool all other nodes’ features to update each node’s feature vector. However, due to the sparsity of protein contact graphs, the edge feature tensor can enjoy a sparse representation of $A \in \mathbb{R}^{N \times N \times K}$ where $N$ is the maximum number of node neighbors and $N \ll N$. Here two residues are neighbors if any of the 11 atom-pair distances is below the threshold (12 Å in this study). Accordingly, when updating the feature vector for node i, that is, $x_i^{(l+1)}$, we just need to sum over all its neighboring nodes j for an equivalent expression.

5 | CONCLUSION

In this article, we propose a novel EGCN for scoring protein-docking models in both relative and absolute senses: ranking docking models and estimating their quality measures (iRMSD). We represent a protein or an encounter complex as an intramolecular or intermolecular residue contact graph with atom-resolution node features and edge features. Inspired by physics, we design a novel graph convolutional kernel that maps the inputs (current node features and fixed edge features) to energy-like outputs (updated node features). Using such energy-based graph convolution layers and state-of-the-art attention mechanisms, we train two graph convolutional networks of identical architectures and distinct parameters to predict intra- and inter-molecular free-energy, respectively. The two networks are trained to together predict an encounter complex’s energy value whereas the true value is approximated by the corresponding native complex’s energy discounted according to the quality of the encounter complex.

The first GCN development for protein docking, our EGCN model is tested against three data sets with increasing difficulty levels: unbound docking with a single decoy-generating mechanism, unbound and homology docking with the single mechanism, as well as unbound and homology docking with diverse decoy-generating mechanisms from the community. For ranking protein-docking models (decoys), EGCN is found to perform better or equally well compared to a state-of-the-art method (IRAD) that has found great success in CAPRI as well as our previous RF model. For quality estimation that has seen few method developments in the field, EGCN is again found to outperform our previous RF model. In both cases, EGCN is relatively insensitive to target difficulty and is of limited sensitivity to training decoys (specifically, the way they are generated). Compared to our previous RF model that learns indirectly from structure-derived energy terms, the EGCN model learns directly from structures and thus shows improved accuracy and robustness.

ACKNOWLEDGMENTS

This work was supported by the National Institutes of Health (R35GM124952). Portions of this research were conducted with high-performance research computing resources provided by Texas A&M University.
