Briefings in Bioinformatics, 2022, 1–14

https://doi.org/10.1093/bib/bbac614

Problem Solving Protocol

High-accuracy protein model quality assessment using attention graph neural networks

Peidong Zhang, Chunqiu Xia and Hong-Bin Shen

Corresponding author. Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, 200240 Shanghai, China. Tel: +86-21-34205320; Fax: +86-21-34204022; E-mail: hbshen@sjtu.edu.cn

Abstract

Great improvement has been brought to protein tertiary structure prediction through deep learning. It is important but very challenging to accurately rank and score decoy structures predicted by different models. CASP14 results show that existing quality assessment (QA) approaches lag behind the development of protein structure prediction methods, where almost all existing QA models degrade in accuracy when the target is a decoy of high quality. How to give an accurate assessment to high-accuracy decoys is particularly useful with the available of accurate structure prediction methods. Here we propose a fast and effective single-model QA method, QATEN, which can evaluate decoys only by their topological characteristics and atomic types. Our model uses graph neural networks and attention mechanisms to evaluate global and amino acid level scores, and uses specific loss functions to constrain the network to focus more on high-precision decoys and protein domains. On the CASP14 evaluation decoys, QATEN performs better than other QA models under all correlation coefficients when targeting average *LDDT*. QATEN shows promising performance when considering only [illegible]curacy decoys. Compared to the embedded evaluation modules of predicted $C_{\alpha}-RMSD$ (*pRMSD*) in RosettaFold and predicted [illegible]*LDDT*) in AlphaFold2, QATEN is complementary and capable of achieving better evaluation on some decoy structures generated [illegible]aFold2 and RosettaFold themselves. These results suggest that the new QATEN approach can be used as a reliable independent [illegible]nent algorithm for high-accuracy protein structure decoys.

Keywords: quality assessment, deep learning, GNNs, attention mechanism, high-accuracy decoys

Introduction

Proteins are involved in almost all biological processes and cellular functions. Functions of proteins highly depend on their spatial structures, namely the three-dimensional arrangement of amino acids [1]. Experimental tests of protein folding are generally expensive and time-consuming [2]. With the development of high-throughput sequencing, protein sequence data accumulate rapidly [3]. In order to bridge the gap between protein sequence and structure database, the development of in silico tertiary structure prediction methods has been a long-term challenging topic in structural bioinformatics [4–13]. The Critical Assessment of Techniques for Protein Structure Prediction (CASP) is designed to benchmark the progress of protein structure prediction methods. In 2020, The latest artificial intelligence algorithms, AlphaFold2 [6] and RosettaFold [14], have achieved significant improved performance in CASP14. Their algorithms have the capability of making accurate protein structure predictions comparable to the lab experiments-based solution. However, there is still a lack of independent convincing computational methods for ranking high-accuracy decoys.

Quality assessment (QA) aims to estimate the accuracy of a protein decoy, but without the knowledge of its ground truth [15, 16]. Global distance test-total score (*GDT-TS*) [17] and local distance difference test (*LDDT*) [18] are widely used to measure the difference between the predicted and native structures. Current QA techniques can be generally divided into two categories: single-model methods that operate on a single protein decoy to estimate its quality, and multi-model methods that use consistency between several candidates to estimate their qualities [19]. Multi-model methods typically have a pool of decoys that are generated by different methods or from different templates, and assume that correct fragments of the target structure are already embedded in the pool. Hence, they evaluate the accuracy of the current decoy by clustering and extracting the consensus information in the pool. Obviously, the performance of multi-model methods depends on the quality and size of the pool [20], and an excellent multi-model QA approach often needs a large pool of dozens of models and hundreds of decoys, resulting in high computational costs accordingly [21]. In contrast, single-model methods usually extract inherent features without external predictors. They mainly are based on the topology and energetic analysis of a single decoy and predict the distance from the current conformation to the lowest energy conformation through structural analysis.

The single-model methods are closer to indicate the distance from current conformation to the lowest energy conformation

Peidong Zhang is a Masters student at the Institute of Image Processing and Pattern Recognition and Key Laboratory of System Control and Information Processing of Shanghai Jiao Tong University.

Chunqiu Xia is a PhD student at the Institute of Image Processing and Pattern Recognition and Key Laboratory of System Control and Information Processing of Shanghai Jiao Tong University.

Hong-Bin Shen is a professor at the Institute of Image Processing and Pattern Recognition and Key Laboratory of System Control and Information Processing of Shanghai Jiao Tong University.

Received: September 23, 2022. **Revised:** November 23, 2022. **Accepted:** December 12, 2022

and have attracted increasing attention in CASP competition. In the recent CASP14, single-model QA methods account for more than 70% of all model QA methods. They typically use different feature combinations and machine learning methods to learn the implicit relationship between decoy structure and its quality. For instance, in ProQ2 and ProQ3 [22, 23], features such as atom-atom contacts, residue-residue contacts, surface exposure, predicted and observed secondary structures, and Rosetta energy terms are fed into support vector machines (SVMs) for calculation.

With the development of deep learning techniques, more neural network-based (NN-based) methods are proposed. For instance, ProQ3D [24] replaces the SVM in ProQ3 with a multilayer perceptron model, and ProQ4 uses more protein structural features such as dihedral angles, secondary structures, hydrogen bond energy and co-evolutionary information for training with multi-stream structure. DeepQA [25] uses a deep belief network to make predictions by using structural, chemical and knowledge-based energy scores. A more intuitive idea in recent years is to feed decoy structures into 3D convolutional networks or graph NNs. For example, 3DCNN_MQA [26] and Ornate [27] project the decoy structure onto a 3D grid and use 3D convolution to convert this voxelized representation into a quality score. On the other hand, ProteinGCN [28] and GraphQA [29] represent decoy structures on the graph at the atomic and amino acid levels, respectively, and then automatically learn the best embeddings from the original node and edge features for local and full-graph attribute prediction. ModFOLD7 [30] and ModFOLD8 [31] are consensus approaches, which integrate single-model methods such as CDA, SSA, ProQ, ProQ2D, ProQ3D, VoroMQA, DBA, MF5s, MFcQ and ResQ7 [32]. This series of methods all top the CASP13 and CASP14 competitions.

However, according to CASP14, the QA is lagging behind the development of protein structure prediction methods [33], where existing QA algorithms could not provide accurate assessment for high-accuracy decoys. Although the overall performance of these QA models in CASP14 is acceptable, for instance, they can still distinguish the poor decoys ($GDT\text{-}TS<0.5$), but when considering those decoys close to native structures (e.g. $GDT\text{-}TS \geq 0.5$), a significant drop in QA accuracy is observed. This highlights the importance of developing protein QA for high-accuracy decoys with the available of some excellent structure prediction algorithms, such as AlphaFold2 and RosettaFold. The potential reasons for the failure of many existing QA methods on the high-accuracy decoys are the features used in QA model, such as homology and coevolution information, which have been widely used and learned in different structure prediction models. The feature overlap may confuse the QA model when evaluating outputs from different structure predictors. In contrast, topological information is more important and effective than evolutionary features which have been used in the stage of protein structure prediction.

In this study, we propose QATEN, a novel single-model QA method based on graph NNs and self-attention mechanism. Protein coevolutionary information, which has been widely used in structure prediction, is not required by QATEN. It only considers types of heavy atoms in amino acid, chemical bonds and topology information as the input. We also design a novel loss function to enhance the attention to high-accuracy decoys and more realistic local segments in one decoy. Compared with most of existing deep learning-based QA pipeline, QATEN has much fewer trainable parameters and is computationally efficient (as shown in section 4 of Supplementary Information). On the CASP14 datasets, QATEN is highly competitive compared with all other single-model QA methods, and achieves state-of-the-art performance on high-precision samples. On some decoys predicted by AlphaFold2 and RosettaFold, our model can provide more correlated predictions.

Materials and methods

Benchmark datasets

Nine benchmark datasets are used in this study. DeepAccNet dataset and GNNRefine dataset are merged for training and validation [33, 34], and we randomly select 5% of the decoys from the two datasets to form the validation dataset. Supplementary Figure S1 shows the decoy quality distribution of these two datasets. Besides, we select decoys from CASP14 EMA (model accuracy evaluation) experiments to form CASP14 dataset, and then select high-accuracy decoys in CASP14 dataset to form CASP14_GDT ($GDT\text{-}TS \geq 0.5$) and CASP14_LDDT (average $LDDT \geq 0.5$) datasets. We also use AlphaFold2 and RosettaFold to predict protein sequence downloaded from CASP14 and RCSB PDB (https://www.rcsb.org/), then use decoys to compose four datasets (CASP14_alphafold, CASP14_rosettafold, RCSB_alphafold and RCSB_rosettafold). These seven datasets are used as independent test sets. Details of the datasets are provided in section 1 of Supplementary Information.

Evaluation metrics

For the main experiments, we restrict the measures of global performances to *GDT-TS* and the average *LDDT*, as they are widely used and are the official scores in the CASP competition. For each QA method, we consider the predicted and ground-truth scores to compute Pearson ($\mathcal{P}$), Kendall ($\mathcal{K}$) and Spearman ($\mathcal{S}$) correlation coefficients across all decoys of all targets. Detailed explanations of the three correlation indicators are available in section 2 of Supplementary Information.

Particularly, RosettaFold and AlphaFold2 also provide an assessment of amino acid residue levels for their predicted decoy structures. AlphaFold2 predicts *LDDT* in its evaluation module while RosettaFold predicts $C_\alpha\text{-}RMSD$. For these two models, we calculate Pearson coefficients between *pLDDT* and *LDDT*, *pRMSD* and *RMSD* on their own predicted decoys, respectively.

Feature extraction

GNNs have been widely used in bioinformatics [34–37], including drug response prediction, cell type identification and protein function prediction. A natural way to represent decoy structures is to model them as graphs, which have been shown to be effective in recent works [28, 29, 38, 39]. We consider the 50 nearest neighboring atoms in the three-dimensional space of each node atom and connect them to the node atom in the protein graph. In protein graphs, the node feature is the one-hot encoding of the atom types (167-d for 167 heavy atom types in 20 standard amino acids); as for edge features (43-d), we specifically include the following three categories.

Distance-based features: we generate pairwise Euclidean distance for atoms connected in the protein graph and use the Gaussian extension to generate a 39-d vector representation of the distance.

Orientational features: a local coordinate reference frame is defined by a triplet ($\boldsymbol{e_x}$,$\boldsymbol{e_y}$, $\boldsymbol{e_z}$) in the backbone. We use $\boldsymbol{v_M}$ and $\boldsymbol{v_N}$ to characterize the coordinates of atoms *M* and *N* in the global frame. Then, atom *B* along with its previous atom *A* and next atom *C* in the backbone are used to derive the three basis vectors $\boldsymbol{e_x}$,$\boldsymbol{e_y}$ and

$\boldsymbol{e}_\mathbf{z}$, which are calculated as follows:

$$\boldsymbol{e}_\mathbf{x} = \frac{\mathbf{c}_{\mathbf{BC}} \times \mathbf{c}_{\mathbf{AB}}}{|\ \mathbf{c}_{\mathbf{BC}} \times \mathbf{c}_{\mathbf{AB}}\ |} \tag{1}$$

$$\boldsymbol{e}_\mathbf{z} = \frac{\mathbf{c}_{\mathbf{AB}} - \mathbf{c}_{\mathbf{BC}}}{|\ \mathbf{c}_{\mathbf{AB}} - \mathbf{c}_{\mathbf{BC}}\ |} \tag{2}$$

$$\boldsymbol{e}_\mathbf{y} = \boldsymbol{e}_\mathbf{z} \times \boldsymbol{e}_\mathbf{x} \tag{3}$$

where $\mathbf{c}_{\mathbf{MN}} = \boldsymbol{v}_\mathbf{M} - \boldsymbol{v}_\mathbf{N}$. Thus, each edge will obtain a 3-d orientational feature that represents the projection of another atom in the local coordinate reference frame. This feature is translationally and rotationally invariant because projections onto local frames are independent of the global position of the protein structure in 3D space.

Bond-related feature: the bond information is encoded as a binary scalar. If a covalent bond connects any two atoms of an edge, it will be set to 1. Otherwise, it will be set to 0.

The above-mentioned three types of features (i.e. distance-based features, orientational features and the bond-related feature) are concatenated as the edge features in our work, which are expected to cover the topological information. The ablation experiments in section 5.1 of Supplementary Information also indicate that these three types of edge features all contribute to the final model performance.

Model architecture

Overview of QATEN pipeline

The QATEN is the first self-attentional model focusing on the ranking of high-precision decoy structures, to the best of our knowledge. It uses the attention mechanism to mine structural features and designs loss function to drive model to pay more attention to high-precision decoys and domains of decoys. The graph convolutional layers in QATEN is invariant to rotation and translation since it is built from the relative positions between nodes [29, 39]. Thus, it is unnecessary to apply data augmentation. Figure 1(A) shows the flowchart of QATEN, which mainly includes three major steps: first, it uses a fully connected layer to embed the original node features, then feeds them and edge features into graph attention layer. Secondly, it will concatenate edge features and corresponding nodes features through adjacency matrix, and update the node representations using self-attention mechanism. Lastly, QATEN will calculate the graph score by graph pooling at the global protein level as well as at the local residue level.

The GNN-based module design is one of the key factors for our QA model. It is composed of multiple self-attention modules and shortcut connections. Node features are updated in each module. As shown in Figure 1(B), the embedding layers are firstly used to represent the raw node features and edge features. When passing through five layers of GNNs, edge features remain unchanged and are spliced at corresponding nodes. The node features after GNN learning and updating are expanded as one dimension and the downstream results will be obtained by two pooling networks, including different graph pooling methods and fully connected layers.

Given a protein graph $\mathcal{G} = (\mathcal{A}, \mathcal{V}, \mathcal{U})$, where $\boldsymbol{a}_\mathbf{ij} \in \mathcal{A}$ indicates whether the ith atom and jth atom are connected, $\boldsymbol{v}_\mathbf{i} \in \mathcal{V}$ indicates node features of the ith atom, $\boldsymbol{\mu}_\mathbf{ij} \in \mathcal{U}$ indicates the edge features between the ith atom and jth atom. We will introduce QATEN formulation below.

Embedding layer

The 167-d one-hot vector of node features are reduced to 16-d through embedding layers and fully connected layers. It allows QATEN to be deeply trained without significant loss of information. Embedded node features and edge features are input into graph attention layers.

Graph attention layer

We use a self-attention module on feature map that indicates topological and biological information between each heavy atom and its nearest 50 neighbors. Then we design a shortcut connection to accelerate convergence and avoid overfitting. Figure 1(C) shows the architecture of one layer. In Figure 1(C), N is the length of atoms, M is the number of neighbors and h is the number of multi-heads. c_a, c_b and c_m, respectively, represent the dimension of node features, edge features and feature map $\mathbf{z}$. In each layer, we concatenate edge features and connected nodes features as the feature map $\mathbf{z}$ for the next layer's input. The representation of feature map $\mathbf{z}$ is shown in formula (4), where $\bigoplus$ is the concatenation between vectors:

$$\mathbf{z}_{ij}^{(k)} = \boldsymbol{v}_i^{(k)} \bigoplus \boldsymbol{v}_j^{(k)} \bigoplus \boldsymbol{\mu}_{ij}^{(k)} \tag{4}$$

For each layer, the total calculation formula is as follows:

$$\boldsymbol{v}_i^{(k+1)} = \delta\left(\boldsymbol{v}_i^{(k)} + \sum\nolimits_j \left(\boldsymbol{\tau}_{ij}^{(k)} \cdot \mathcal{W}_{output}^{(k)} + \boldsymbol{b}_{output}^{(k)}\right)\right) \tag{5}$$

The superscript k in the formula means the kth module in the network, which can be 1, 2, 3, 4, 5. $\delta(\cdot)$ represents ReLU nonlinear activation function. $\sum_j(\cdot)$ represents summing tensors on the second dimension, which means summing the features of all neighboring nodes to the current node. $\mathcal{W}_{output}^{(k)}$ and $\boldsymbol{b}_{output}^{(k)}$ are the convolution weight matrix and convolution bias matrix in the fully connected layer that is used to capture the multi-head information. In addition, $\boldsymbol{\tau}_{ij}^{(k)}$ is the feature map calculated by the former self-attention mechanism:

$$\boldsymbol{\tau}_{ij}^{(k)} = \sigma\left(\boldsymbol{\lambda}_{gating}^{(k)}\right) \odot \left(\vartheta\left(\boldsymbol{\lambda}_{queries}^{(k)} \cdot \left(\boldsymbol{\lambda}_{keys}^{(k)}\right)^{T}\right) \cdot \boldsymbol{\lambda}_{values}^{(k)}\right) \tag{6}$$

where $\boldsymbol{\lambda}_{gating}^{(k)}, \boldsymbol{\lambda}_{queries}^{(k)}, \boldsymbol{\lambda}_{keys}^{(k)}, \boldsymbol{\lambda}_{values}^{(k)}$ are four feature maps calculated, respectively, by four multi-head fully connected layers with reconstructed protein map $\mathbf{z}_{ij}^{(k)}$ as input. $\vartheta(\cdot)$ represents the softmax function, and $\sigma(\cdot)$ represents the sigmoid function. $\odot$ is the element-wise product.

$$\boldsymbol{\lambda}_{gating}^{(k)} = \delta\left(\mathbf{z}_{ij}^{(k)} \cdot \mathcal{W}_{gating}^{(k)} + \boldsymbol{b}_{gating}^{(k)}\right) \tag{7}$$

$$\boldsymbol{\lambda}_{queries}^{(k)} = \delta\left(\mathbf{z}_{ij}^{(k)} \cdot \mathcal{W}_{queries}^{(k)} + \boldsymbol{b}_{queries}^{(k)}\right) \tag{8}$$

$$\boldsymbol{\lambda}_{keys}^{(k)} = \delta\left(\mathbf{z}_{ij}^{(k)} \cdot \mathcal{W}_{keys}^{(k)} + \boldsymbol{b}_{keys}^{(k)}\right) \tag{9}$$

$$\boldsymbol{\lambda}_{values}^{(k)} = \delta\left(\mathbf{z}_{ij}^{(k)} \cdot \mathcal{W}_{values}^{(k)} + \boldsymbol{b}_{values}^{(k)}\right) \tag{10}$$

where $\mathcal{W}_{gating}^{(k)}, \mathcal{W}_{queries}^{(k)}, \mathcal{W}_{keys}^{(k)}, \mathcal{W}_{values}^{(k)}$ and $\boldsymbol{b}_{gating}^{(k)}, \boldsymbol{b}_{queries}^{(k)}, \boldsymbol{b}_{keys}^{(k)}, \boldsymbol{b}_{values}^{(k)}$ are the convolution weight matrices and convolution bias

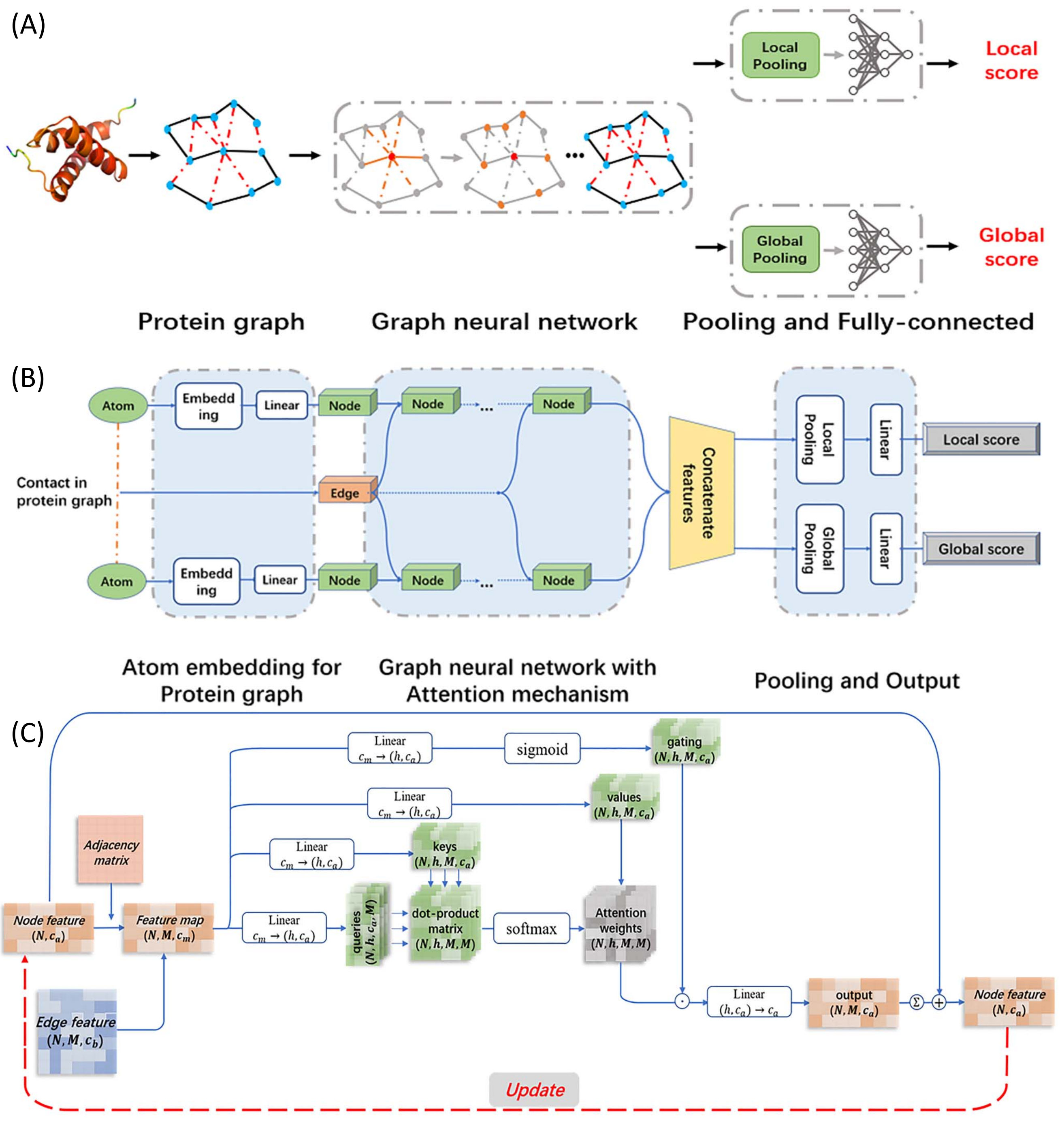

Figure 1. (**A**) The flowchart of QATEN that mainly includes three steps: (1) extract atomic and geometric features from the initial structure and represent the initial decoy structure as a protein graph; (2) predict and update the node embeddings in the graph using GNNs; and (3) calculate the result scores through different pooling network. (**B**) The network architecture of QATEN. The original features of a pair of connected atoms in the protein graph are represented as node features and edge features by embedding layers. Edge features remain unchanged and are spliced at corresponding nodes, and node features are updated in GNNs. Node features are concatenated and expanded as one dimension, then sent into downstream pooling networks. (**C**) One layer based on self-attention module in GNNs, and we design a shortcut connection to avoid overfitting.

matrices in the fully connected layer of the kth module. We have also conducted the ablation experiments about attention module, and the results are shown in section 5.2 of the Supplementary Information.

Graph pooling

For the node embedding representation $\boldsymbol{v}'_i$ obtained by the forward calculation, the global and local scores are obtained by using the graph pooling technique:

$$\mathrm{Score}_G = \delta\left(\left(\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{v}'_i\right)\cdot \mathcal{W}_l + \boldsymbol{b}_l\right) \tag{11}$$

$$\mathrm{Score}_{R_k} = \delta\left(\left(\frac{1}{N_{R_k}}\sum_{j\in R_k}\boldsymbol{v}'_j\right)\cdot \mathcal{W}_m + \boldsymbol{b}_m\right) \tag{12}$$

where Score_G and Score_{R_k} represent the final global score and the kth amino acid's local score. N_{R_k} represents the set of atoms that

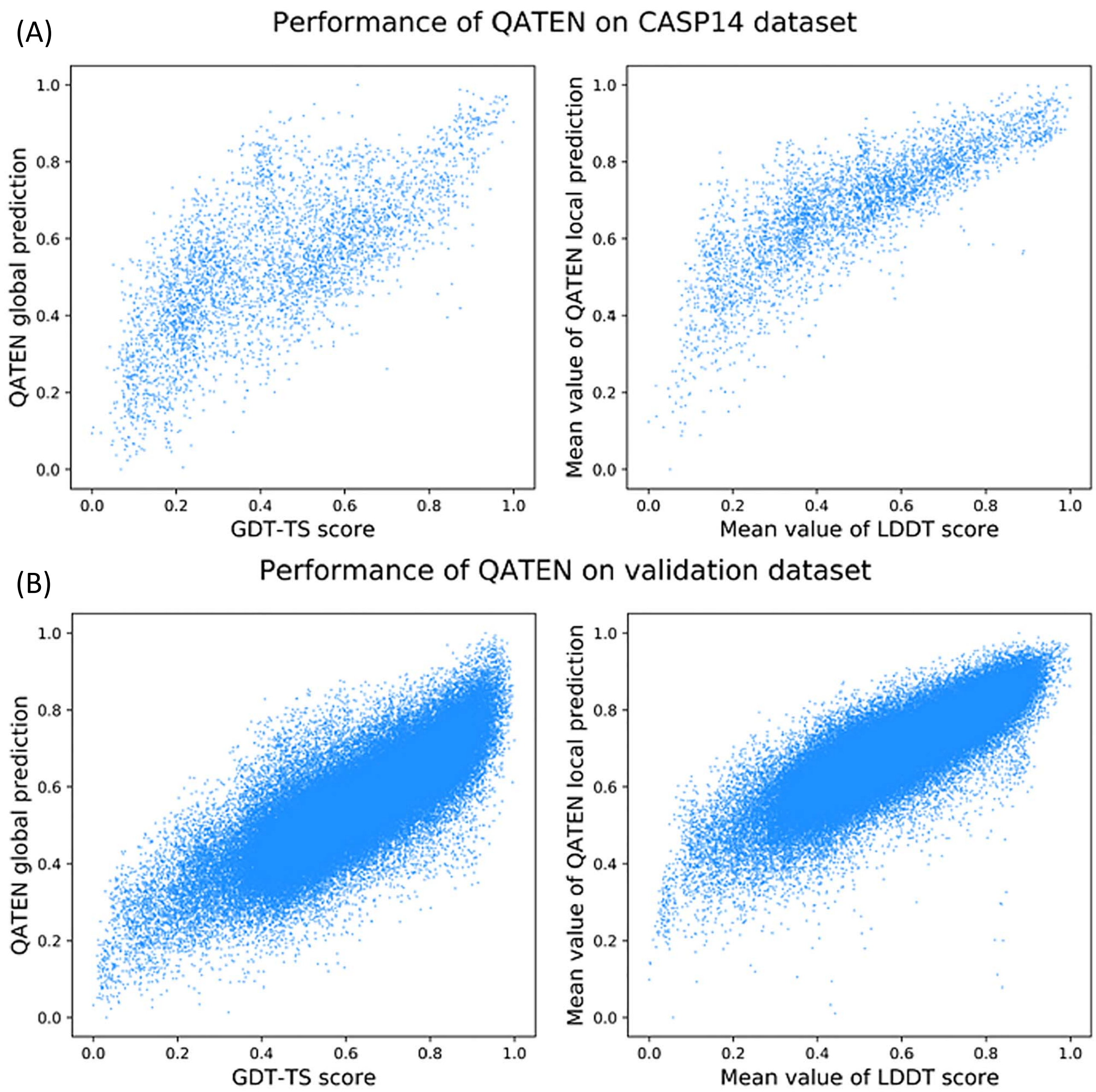

Figure 2. The correspondence between QATEN predictions and real measures on the validation set and CASP14 test set. (**A**) $\mathcal{P}$ is 71.77% on CASP14 dataset (3897 decoys) when considering *GDT-TS* as the metric. And $\mathcal{P}$ is 82.17% on CASP14 dataset when considering the mean value of *LDDT* as the metric. (**B**) $\mathcal{P}$ is 81.25% on the validation dataset (87,634 decoys) when considering *GDT-TS* as the metric. And $\mathcal{P}$ is 84.74% on the validation dataset when considering *LDDT* as the metric.

up the *k*th amino acid, $\mathcal{W}_l$, $\boldsymbol{b}_l$, $\mathcal{W}_m$, $\boldsymbol{b}_m$ are the weight matrices as matrix in the global and local fully connected layer.

function

We design a novel loss function for the training model to focus on high-precision decoys and the domains of decoys. Structure prediction models could not accurately predict unfamiliar protein sequence before CASP13; thus, it was a relatively easier challenge to identify decoys and domains that were incorrectly modeled. Most QA models tend to design more complex networks and use more features, including MSA information, PSSM and DSSP, to improve their results. However, general advances in structure prediction models in CASP14 have made it difficult for QA models to successfully rank high-precision decoys. AUC has been used as a metric in CASP14 for QA method evaluation, which is expected to reflect the ability of QA models to distinguish high-accuracy decoys ($GDT\text{-}TS \geq 0.5$) from low-quality decoys ($GDT\text{-}TS < 0.5$). Nevertheless, Supplementary Figure S2 shows general success for almost all QA models in AUC, whereas Table S2–S3 show obvious accuracy loss of QA models when separately ranking all decoys and the highly accurate decoys.

Thus, the current QA challenge is to accurately evaluate and rank the decoy structures with general or good accuracies independently, which will guide the direction of further precise local refinement. To this motivation, we design the loss function in formula (13). Specifically, we have global label (*GDT-TS*) for each decoy and local labels (*LDDT*) for all amino acids in the decoy. We need to consider both constraints when designing the loss

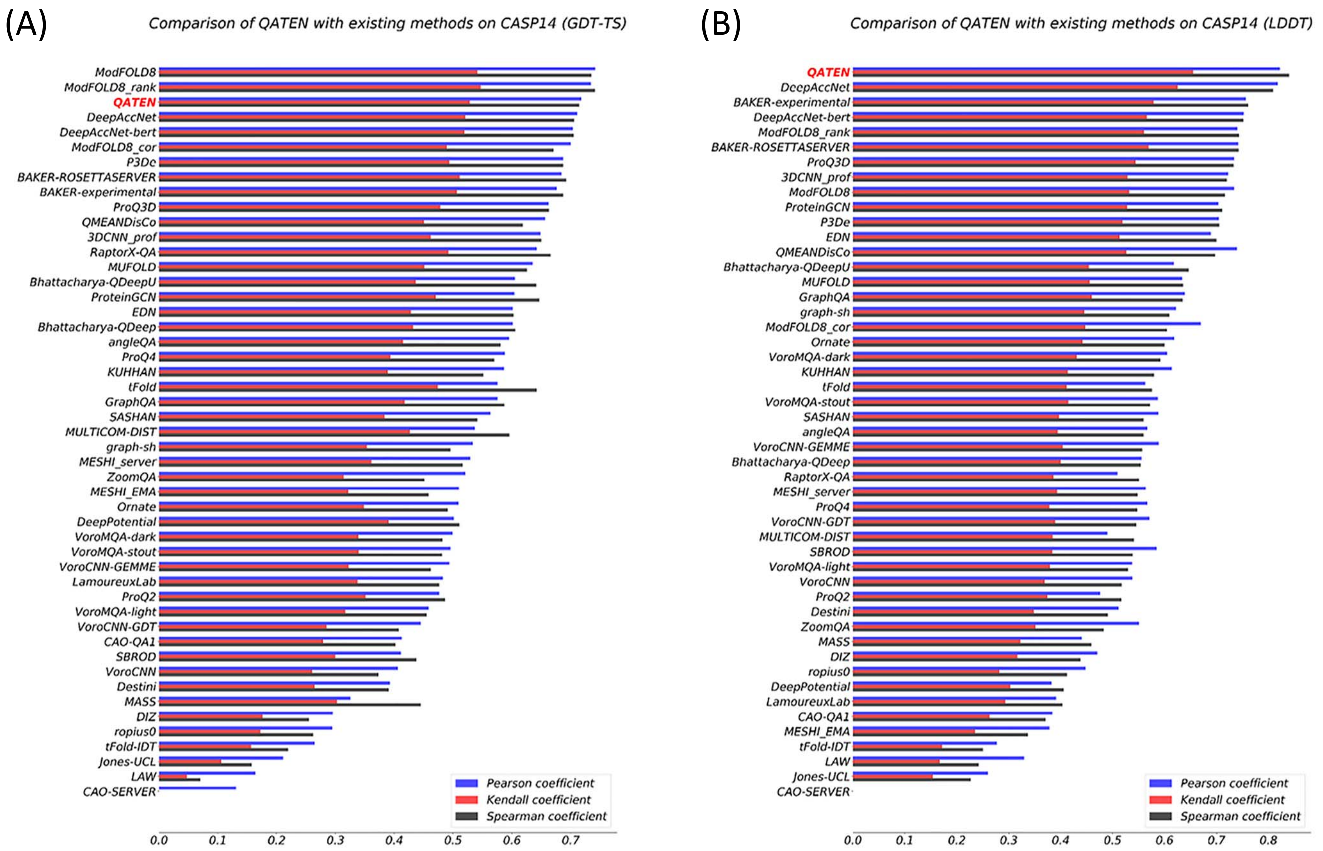

Figure 3. Ranking performance of QATEN and other baseline models on CASP14 dataset by Pearson ($\mathcal{P}$), Kendall ($\mathcal{K}$) and Spearman ($\mathcal{S}$) correlation coefficients. (**A**) Using *GDT-TS* as the measure. (**B**) Using average *LDDT* as the measure.

function, so the two items calculate the distance, respectively. Assuming that a decoy consists of L amino acids, the final score obtained by the network is $Score = (Score_G, Score_{R_1}, \ldots\ldots, Score_{R_{L-1}}, Score_{R_L})$, and the corresponding label is also $Y = (Y_G, Y_{R_1}, \ldots\ldots, Y_{R_{L-1}}, Y_{R_L})$, then the loss of the decoy is defined by the following formula:

$$Loss = (1 + \varepsilon\,(Score_G - 0.5))\,(Y_G - Score_G)^2 + \sum\nolimits_i \left(1 + \varepsilon\left(Score_{R_i} - 0.5\right)\right)\left(Y_{R_i} - Score_{R_i}\right)^2 \qquad (13)$$

Different from a simple MSE loss function, we use two step functions $\varepsilon(\cdot)$ as thresholds. 0.5 is often considered as a threshold in decoy quality, which is also considered in our loss function. The former $\varepsilon(\cdot)$ helps QATEN rank decoys that are generally well predicted, so as to select the decoys that are closest to the natural structures. The latter $\varepsilon(\cdot)$ helps QATEN specifically evaluate the high-precision domains, thus indicating more accurate refinement directions.

Obviously, the loss function will focus on high-precision decoys and local segments in each decoy that are closer to the ground truth. It explains the success of QATEN on high-precision decoy datasets and decoys generated by AlphaFold2 and RosettaFold. The basic distance loss ensures that QATEN is also able to identify decoys modeled incorrectly. Besides, we do not enforce the weight of global labels in the loss function, resulting in QATEN being more successful in the local evaluation under *LDDT* analysis. The ablation experiments about loss function can be found in section 5.3 of Supplementary Information.

Results and discussion

Performance on the validation dataset and CASP14 dataset

Figure 2 shows the correspondence between QATEN predictions and real measures on the validation set and CASP14 test set. The prediction results of all decoy structures of these target proteins not included in the training set are well correlated with *GDT-TS* and average *LDDT*. Almost all of the samples are distributed around the positive correlation line ($y = x$). Overall, although the detailed prediction is not pixel perfect (e.g. in the CASP14 test set, the predicted value of some decoy structures is lower than the actual *GDT-TS* score), our results still show a good correlation to both indicators and are sufficient to provide good support for the structure prediction model.

Comparison with other QA models

We test QATEN on the decoys of 26 CASP14 protein targets and compare it with all single-model QA methods in CASP14 [33]. Besides, we also compare QATEN with some publicly available deep learning-based software such as DeepAccNet, DeepAccNet-Bert [40], ProteinGCN [28] and ZoomQA [41]. We directly use their released model parameters. Performance of QATEN is slightly lower than ModFold8 and ModFold8_rank, comparable with DeepAccNet while better than the others on all the $\mathcal{P}$, $\mathcal{K}$ and $\mathcal{S}$ correlation coefficients when targeting *GDT-TS* (Figure 3A). As shown in Figure 3(B), QATEN performs better than the others on all the $\mathcal{P}$, $\mathcal{K}$ and $\mathcal{S}$ correlation coefficients when targeting average *LDDT*. It is worth mentioning that QATEN only uses a much smaller network with about 80k trainable parameters.

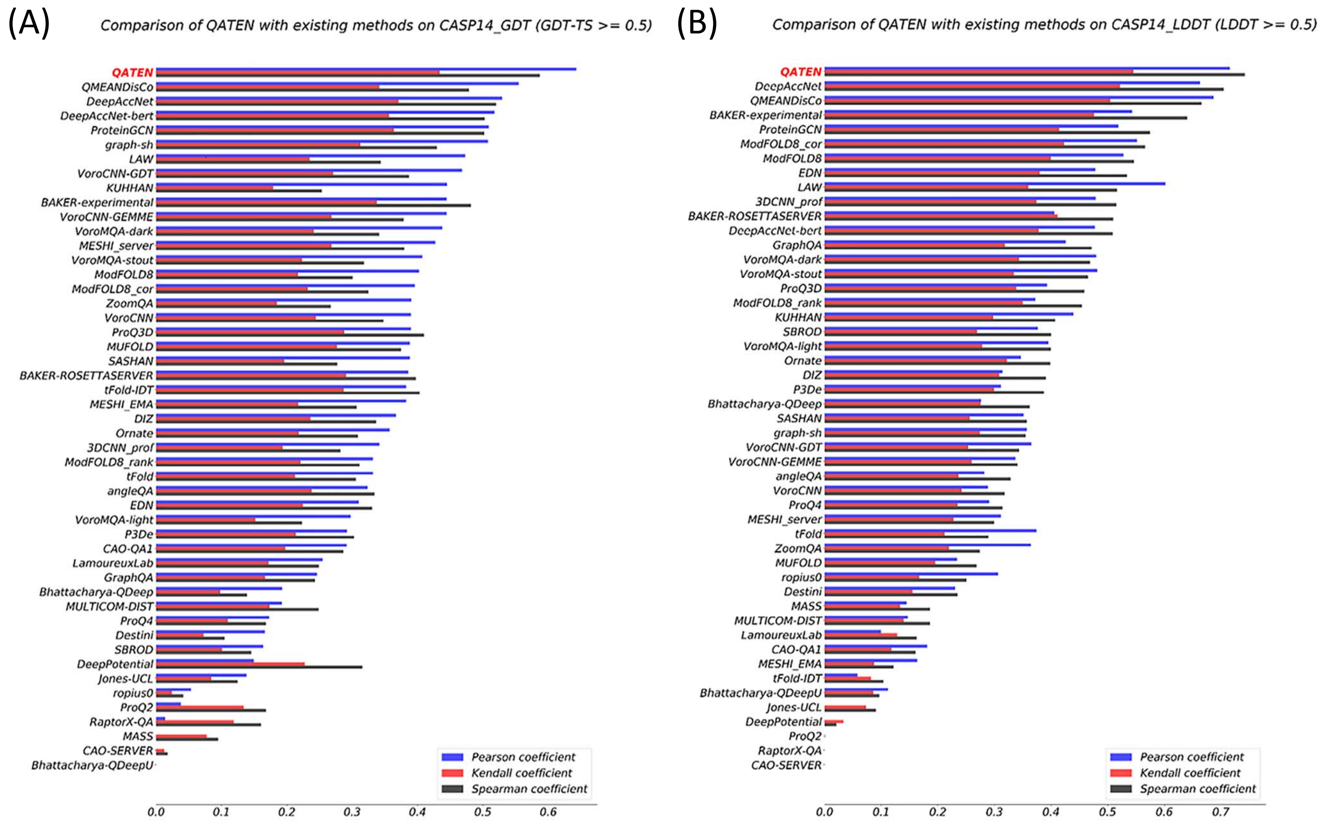

Figure 4. Ranking performance of QATEN and other baseline models on high-accuracy CASP14 dataset by Pearson ($\mathcal{P}$), Kendall ($\mathcal{K}$) and Spearman ($\mathcal{S}$) correlation coefficients. (**A**) Considering *GDT-TS* as a measure. (**B**) Considering average *LDDT* as a measure.

QATEN outperforms existing QA models on high-accuracy decoys

QATEN achieves the best performance on CASP14_GDT and CASP14_LDDT when considering the correlation coefficient on high-precision decoys (Figure 4A and B). When considering high-precision decoys, we observe a significant decrease in accuracy in most models which can be demonstrated by comparing Figures 3–4. A potential reason is that most QA models focus on learning coevolutionary information, which has been done by structure prediction methods. Traditional QA models tend to separate high-accuracy decoy structures, which may neglect the ranking and selection of them. In fact, with the success of AlphaFold2 and RosettaFold, as well as advances of other structure prediction models on CASP14, a significant amount of high-precision decoy structures are predicted, which makes QA models underperform on the high-accuracy decoys. Detailed values of Figures 3–4 are given in Supplementary Tables S2–S3.

Table 1 shows the results of our further comparison of high-precision decoys, which are further divided into two parts: relatively accurate ($0.5 \leq GDT\text{-}TS < 0.8$) and highly accurate ($GDT\text{-}TS \geq 0.8$), and QATEN achieves the best performance on correlation indicators. A complete comparison list could be found in Supplementary Table S4.

Comparison with the QA module in RosettaFold

QA module in RosettaFold uses predicted C_{α}-*RMSD* (*pRMSD*) to locally evaluate its predicted decoy structure; thus, the correlation coefficients are obtained by calculating[[ineq82]]*pRMSD* and *RMSD*. If the $\mathcal{P}$ of a decoy is higher than 0.5, it is defined as a *hit decoy*. Table 2 shows the number of *hit decoys* and average $\mathcal{P}$ by QATEN and QA module in RosettaFold, on CASP14_rosettafold and RCSB_rosettafold datasets respectively. As indicated by the results on the datasets of this study, QATEN provides a more precise assessment on general decoy and hit much more decoys, both in CASP14_rosettafold and RCSB_rosettafold datasets. These results indicate QATEN's possible complementary support for the *pRMSD* evaluation module of RosettaFold.

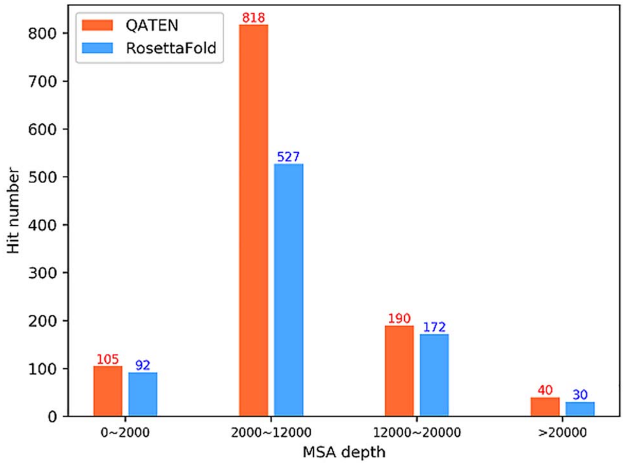

Figure 5. Distribution of the *hit decoys* according to MSA depth. QATEN outperforms on decoys with different MSA depths and has more advantages on the decoys with MSA depths from 2000 to 12,000.

We merge CASP14_rosettafold and RCSB_rosettafold and further explore the experimental results from the perspective of MSA depth (the MSA is searched and generated by RosettaFold) and secondary structure. MSA depth in Figure 5 is generally higher

Table 1. Ranking performance of QATEN and other top-10 baseline models on different bins of accuracy (using *GDT-TS* as the distinguishing criterion)

Methods/group name	Relatively accurate ($0.5 \leq GDT\text{-}TS < 0.8$)[a]			Highly accurate ($GDT\text{-}TS \geq 0.8$)[b]		
	$\mathcal{P}$	$\mathcal{K}$	$\mathcal{S}$	$\mathcal{P}$	$\mathcal{K}$	$\mathcal{S}$
QATEN	**0.5568**	**0.3841**	**0.5275**	**0.4066**	**0.3281**	**0.4253**
DeepAccNet	0.4559	0.3193	0.4559	0.3405	0.2710	0.3636
ProteinGCN	0.4339	0.3095	0.4147	0.2823	0.2535	0.3379
DeepAccNet-bert	0.4292	0.3049	0.4373	0.2345	0.2316	0.3020
tFold-IDT	0.4160	0.2851	0.3965	0.0909	0.0559	0.0813
QMEANDisCo	0.4042	0.2565	0.3718	0.2305	0.2142	0.3128
graph-sh	0.3874	0.2555	0.3574	0.0267	0.0032	0.0096
MUFOLD	0.3668	0.2624	0.3614	0.0744	0.0539	0.0818
ProQ3D	0.3575	0.2654	0.3839	0.1476	0.0985	0.1443
MESHI_server	0.3272	0.2178	0.3119	0.0857	0.1064	0.1501
VoroCNN-GEMME	0.3223	0.2088	0.2974	0.0641	0.0474	0.0711

[a] 1144 decoys. [b] 49 decoys. Bold font is used to indicate the state-of-the-art under current metric.

Table 2. Comparison between QATEN and the QA module in RosettaFold

Dataset	Method	N_hit_{decoy}	Average $\mathcal{P}$
RCSB_rosettafold	**QATEN**	1051	0.6905
	pRMSD	731	0.6132
CASP14_rosettafold	**QATEN**	102	0.664
	pRMSD	90	0.6138

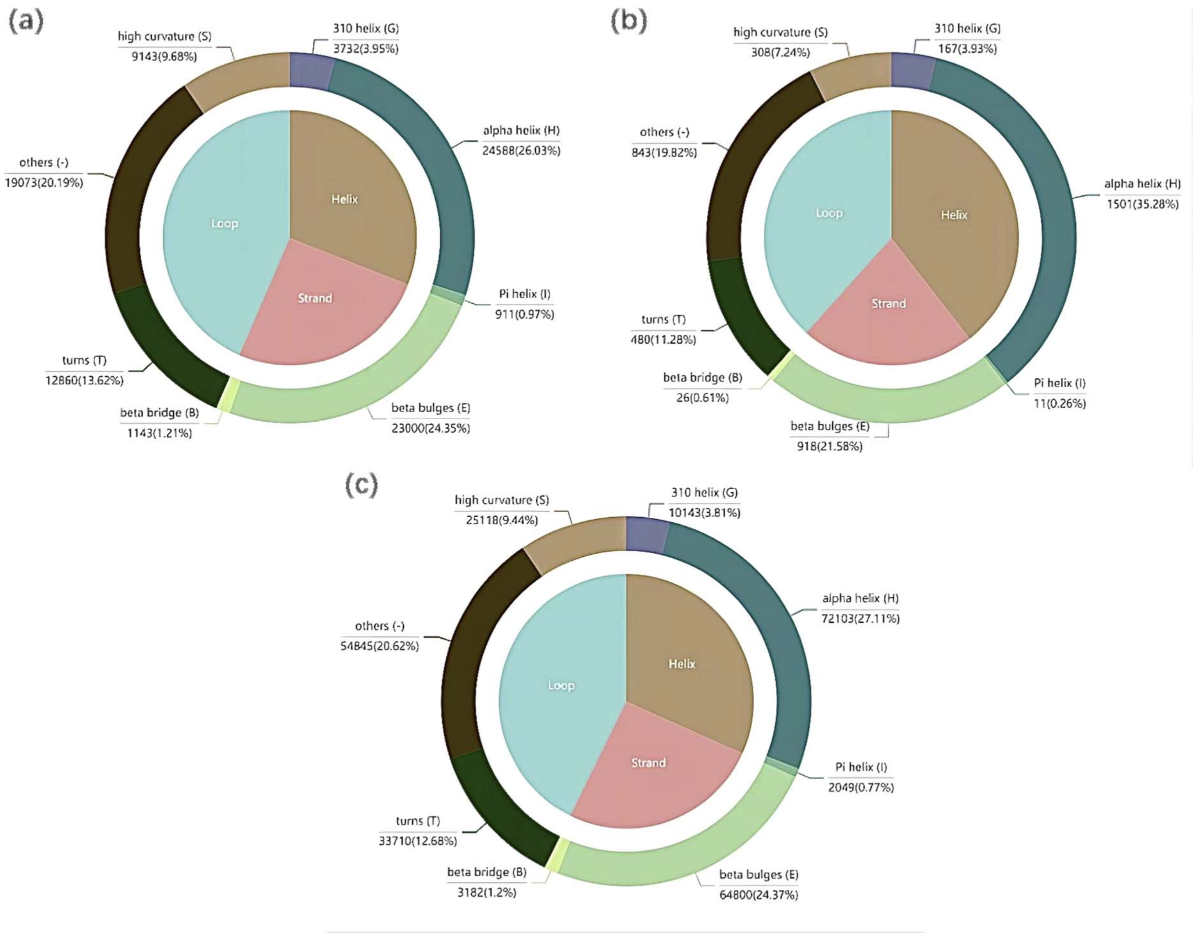

Figure 6. Distribution of secondary structure elements in decoys only hit by (**A**) QATEN or (**B**) QA module in RosettaFold. Additionally, the distribution of secondary structure elements in all decoys is also shown in (**C**). The distribution of secondary structure elements mainly concentrates on Strand, β bridge and longer sets of hydrogen bonds and β bulges, and Helix, α helix and π helix.

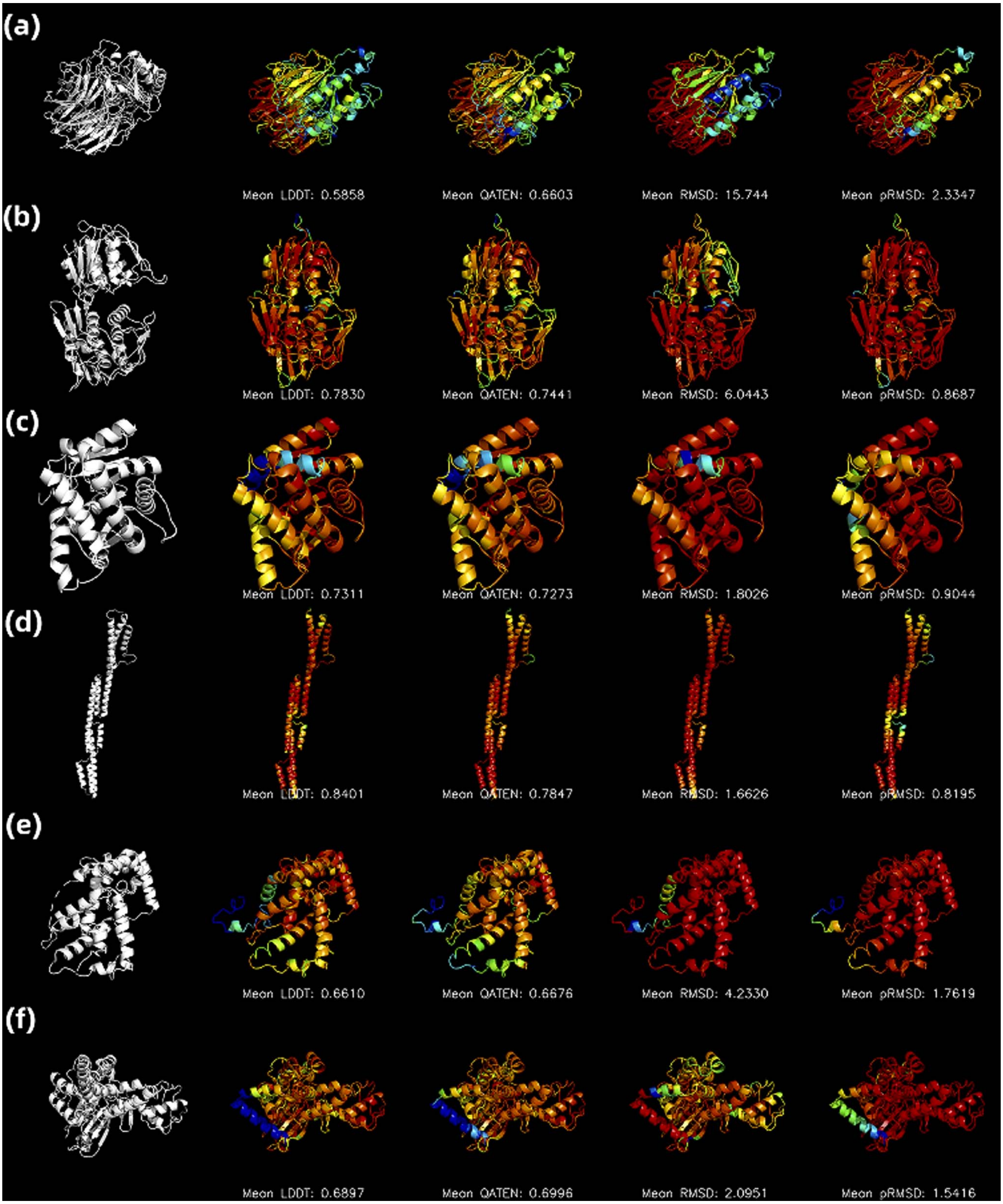

Figure 7. Comparison of QATEN and *pRMSD* of RosettaFold on 7FH7-model-1 (**A**), 7PXY-model-4 (**B**), T1025-model-4 (**C**), T1030-model-5 (**D**), T1042-model-4 (**E**) and T1079-model-4 (**F**). In each line, the first subgraph (gray) represents the native structure after alignment. The second to fifth subgraphs all represent the decoy structure, and the colors on the structure represent the score distribution of different evaluation methods (*LDDT*, score of QATEN, *RMSD*, *pRMSD* of RosettaFold), and the mean value of the score is shown under structures. Particularly, *RMSD* and *pRMSD* are back-normalized to be between 0 and 1 for ease of comparison. The range of color (red-orange-yellow-green-cyan-blue-purple) is correlated with the range of scores (1–0).

Table 3. Comparison between QATEN and confidence predictor in AlphaFold2

Dataset	Method	N_hit_{decoy}	Average $\mathcal{P}$
RCSB_alphafold	**QATEN**	817	0.5892
	pLDDT	774	0.6102
CASP14_alphafold	**QATEN**	105	0.6368
	pLDDT	102	0.647

than that in Figure 8 because Rosetta tends to ignore redundancy when searching for homologous information. As shown in Figure 5, QATEN outperforms on decoys with different depths of MSA and has more advantages than the decoys when the depth of MSA is lower.

We count the distribution of secondary structure elements on 359 decoys only hit by QATEN and 27 decoys only hit by the QA module in RosettaFold, as shown in Figure 6. The difference of distribution of secondary structure elements in two sets mainly concentrates on Strand and Helix. Compared with general decoy in CASP14_rosettafold and RCSB_rosettafold datasets, decoys only hit by QATEN do not show a significant difference. More helices (+7.8%) can be observed on those decoys only hit by QA module in RosettaFold. These results indicate that the embedded QA module in RosettaFold evaluates helix structure better, which is consistent with its structure prediction module. These results also suggest the potential complementarity between the embedded QA module of RosettaFold and QATEN. Comparison with the confidence predictor in AlphaFold2 also indicate similar phenomenon.

In Figure 7, we make six cases comparison between the QATEN and the RosettaFold assessment *pRMSD*. The structure prediction module in RosettaFold performs quite well, because the model has been further improved for the target proteins proposed in CASP14. As can be seen from Figure 7(C) and (D), RosettaFold predicts these two decoys well but QATEN can provide a more precise evaluation ($\mathcal{P}$ as 0.9035 and 0.7444) than the assessment module of RosettaFold ($\mathcal{P}$ as 0.39 and 0.5366) at decoys with simple structures. Figure 7(B) and (F) show QA module in RosettaFold probably ignores some domains that need improvement and gives spuriously high *pRMSD*, resulting in bad evaluations ($\mathcal{P}$ as 0.202 and 0.3681). QATEN precisely identifies these domains for improvement and achieves higher $\mathcal{P}$ (0.7541 and 0.7569). Figure 7(A) and (D) shows two decoys relatively poorly predicted by structure prediction module in RosettaFold, which are also poorly evaluated by QA module in RosettaFold ($\mathcal{P}$ as 0.5425 and 0.5366). QATEN provides more accurate evaluations ($\mathcal{P}$ as 0.8682 and 0.7444) than RosettaFold does.

Comparison with the confidence predictor in AlphaFold2

The confidence predictor in AlphaFold2 uses *pLDDT* to make the local evaluation of its predicted decoy structures. Likewise, we also compare the number of *hit decoys* and average $\mathcal{P}$ by QATEN and confidence predictor in AlphaFold2, shown in Table 3. Although QATEN is 1–2% points lower on average $\mathcal{P}$ for all decoys, it is slightly better in both datasets on numbers of *hit decoy*. Protein chains in RCSB_alphafold have more homologous information, consequently helping widen the gap on average $\mathcal{P}$ between QATEN and AlphaFold2. It is worth noting that the confidence predictor is also involved in the training and recycling process [6], and both datasets are generated by AlphaFold2 itself, which may help the embedded QA module gain advantages.

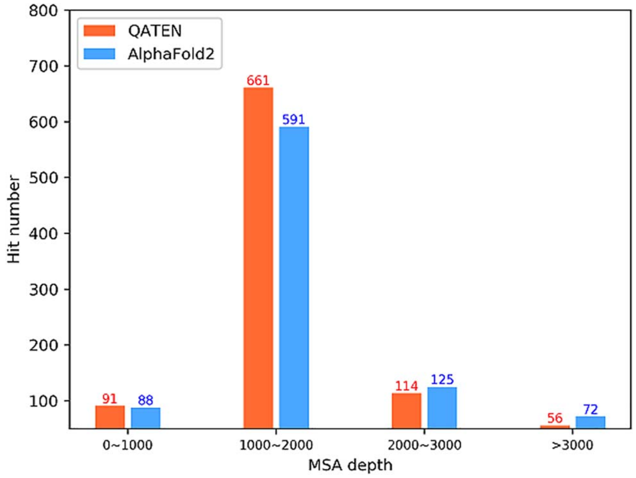

Figure 8. Distribution of the *hit decoys* according to MSA depth. QATEN performs better on decoys with MSA depth lower than 2000, whereas the confidence predictor in AlphaFold2 hit almost all decoys (93.75%, 80 decoys in total) with MSA depth above 3000.

We merge CASP14_alphafold and RCSB_alphafold datasets and further explore the potential preference in embedded QA module, from the perspectives of multiple sequence alignment (MSA) depth and secondary structure. Figure 8 shows the distribution of *hit decoys* according to MSA depth (MSA is searched and generated by AlphaFold2). QATEN outperforms confidence predictor in AlphaFold2 when the depth of MSA is lower than 2000 and underperforms when that is higher than 2000. It is still challenging for predicting orphans by AlphaFold2 for the potential dependence on MSA information [42, 43], and results in Figure 8 suggest QATEN may provide complementarily accurate evaluation on protein decoys with less available MSA.

We find 166 decoys only hit by QATEN and 120 decoys only hit by confidence predictor in AlphaFold2, and derive the distribution of secondary structure elements in those decoys by DSSP [44], as shown in Figure 9. Compared with general decoys, more helices (+11%) and less strands (−8%) could be significantly observed on those decoys only hit by confidence predictor in AlphaFold2. Less helices (−10%) and more strands (+8.5%) could be observed on those decoys only hit by QATEN. These results also suggest the complementarity between the two QA methods.

In Figure 10, we also make six cases comparison between the QATEN and the confidence predictor in AlphaFold2, with each graph representing a decoy sample predicted by AlphaFold2. As can be seen from Figure 10(A) and (B), AlphaFold2 predicts

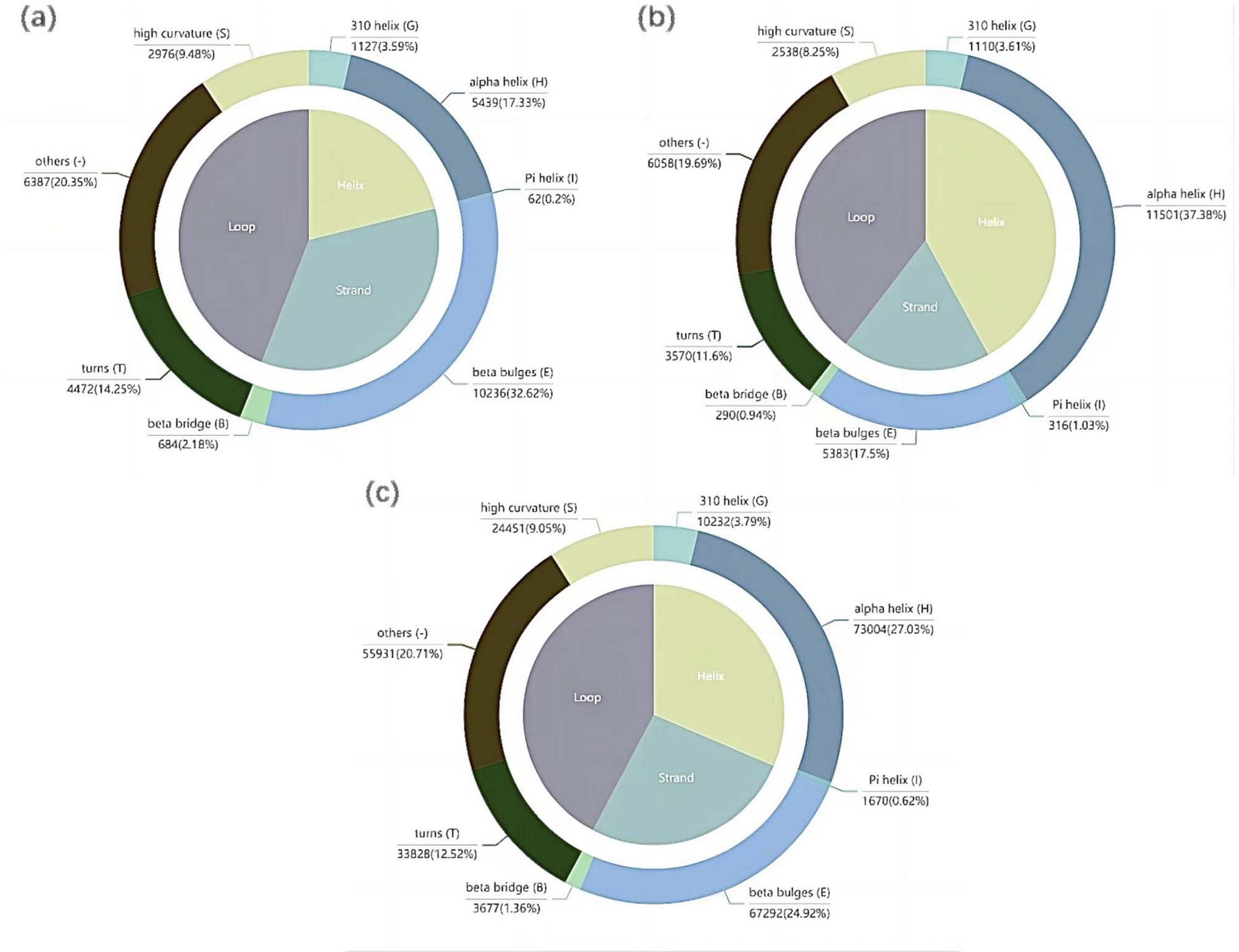

Figure 9. Distribution of secondary structure elements in decoys only hit by (**A**) QATEN or (**B**) confidence predictor in AlphaFold2. Additionally, the distribution of secondary structure elements in all decoys is also shown in (**C**). The distribution of secondary structure elements mainly concentrates on Strand, β bridge and longer sets of hydrogen bonds and β bulges, and Helix, α helix and π helix.

5SBV-model-3 and 7B5U-model-5 quite well but confidence predictor in AlphaFold2 may overestimate some domains of these decoys. The confidence predictor ignores these domains, resulting in high *pLDDT* that is highly different from *LDDT* ($\mathcal{P}$ as 0.5938 and 0.6328). QATEN provides a more conservative assessment, but achieves higher $\mathcal{P}$ (0.7178 and 0.7555) than that of *pLDDT*. It suggests QATEN could find the domains need improvement in the decoys overestimated by the confidence predictor in AlphaFold2. As shown in Figure 10(C) and (D), AlphaFold2 predicts T1031 and T1039 well but the confidence predictor in AlphaFold2 provides relatively poor evaluation. $\mathcal{P}$ of *pLDDT* are 0.3895 and 0.5996. QATEN provides closer assessment scores to *LDDT*, and achieves higher $\mathcal{P}$ as 0.6 and 0.7281. Figure 10(E) and (F) shows two decoys relatively poorly predicted and evaluated (T1029-model-3 and T1033-model-4), QATEN performs better evaluations on them with 13% and 38% higher on $\mathcal{P}$. These results indicate that QATEN can be a complementary QA tool for the decoys generated by AlphaFold2.

Conclusions

This paper presents a new method QATEN for independent protein QA. QATEN uses GNNs and the attention mechanism to update node representation in protein graph from the decoys structural features, then concatenates and expands the predicted node features into a pooling network to obtain the final results. In order to get more accurate evaluations on high-accuracy decoys, we have designed new attention network architecture and loss functions. In our current implementation, QATEN does not make use of any sequence homologs and secondary structure features. QATEN is also computationally efficient with simple network architecture. Evaluation of performance on multiple benchmark datasets shows that QATEN can derive accurate evaluations. We provide online service and open source the code of QATEN for non-commercial use. The web server and source code of QATEN is available at http://www.csbio.sjtu.edu.cn/bioinf/QATEN/ and https://github.com/CQ-zhang-2016/QATEN.

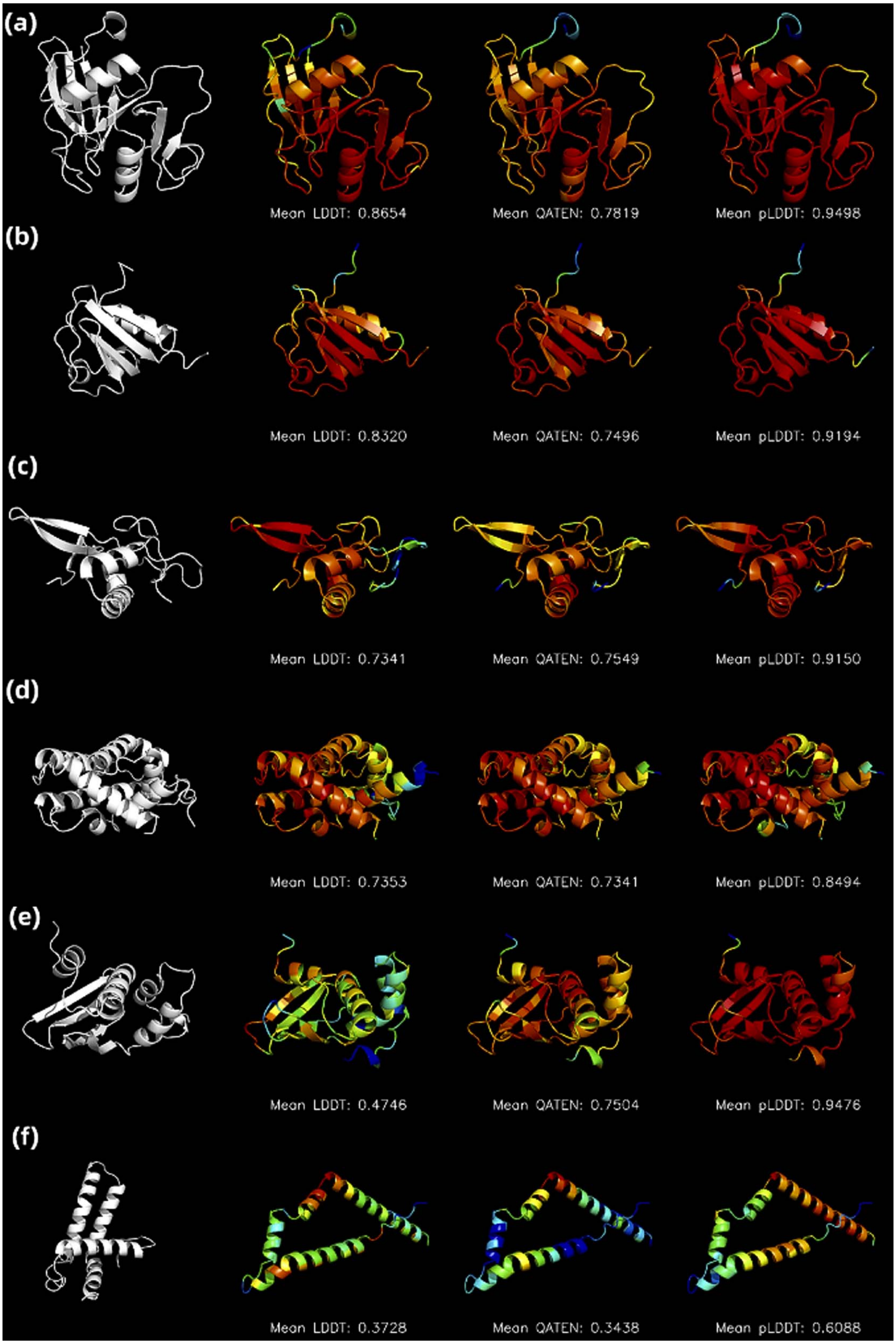

Figure 10. Comparison of QATEN and *pLDDT* of AlphaFold2 on 5SBV-model-3 (**A**), 7B5U-model-5 (**B**), T1031-model-4 (**C**), T1039-model-1 (**D**), T1029-model-3 (**E**) and T1033-model-4 (**F**). In each line, the first subgraph (gray) represents the native structure after alignment. The second to fourth subgraphs all represent the decoy structures, and the colors on the structure represent the score distribution of different evaluation methods (*LDDT*, score of QATEN, *pLDDT* of AlphaFold2), and the mean value of the score are shown under structures. The range of color (red-orange-yellow-green-cyan-blue-purple) is correlated with the range of scores (1–0).

Key Points

- We propose a novel graph neural network (GNN)-based model QATEN for high-accuracy protein model quality assessment.
- In our model, GNNs and attention mechanisms are effectively used, and specific loss functions enable the QATEN to focus more on high-accuracy decoys and domains.
- QATEN is compared with the state-of-the-art single-model quality assessment models on benchmark datasets, and shows promising performance when considering only high-accuracy decoys.
- QATEN is capable of achieving evaluation on decoy structures with high computational efficiency.

Supplementary data

Supplementary data are available online at https://academic.oup.com/bib.

Data availability

The datasets and source code can be free[illegible]ed from http://www.csbio.sjtu.edu.cn/bioinf/QATEN/

Funding

National Natural Science Foundation of China (No. 61725302, 62073219); Science and Technology Commission of Shanghai Municipality (22511104100).

References

1. Gawehn E, Hiss JA, Schneider G. Deep learning in drug discovery. *Mol Inform* 2016;**35**(1):3–14.
2. Jacobson MP, Sali A. Comparative protein structure modeling and its applications to drug discovery. *Annu Rep Med Chem* 2004;**39**:259–76.
3. Zhang GJ, Zhou XG, Yu XF, *et al.* Enhancing protein conformational space sampling using distance profile-guided differential evolution. *IEEE/ACM Trans Comput Biol Bioinform* 2017;**14**(6): 1288–301.
4. Ma J, Wang S, Zhao F, *et al.* Protein threading using context-specific alignment potential. *Bioinformatics* 2013;**29**(13):i257–65.
5. Yang J, Anishchenko I, Park H, *et al.* Improved protein structure prediction using predicted interresidue orientations. *Proc Natl Acad Sci USA* 2020;**117**(3):1496–503.
6. Jumper J, Evans R, Pritzel A, *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* 2021;**596**(7873):583–9.
7. Rohl CA, Strauss CEM, Misura KMS. Protein structure prediction using Rosetta. *Methods Enzymol* 2004;**83**:66–93.
8. Zhang Y. I-TASSER server for protein 3D structure prediction. *BMC Bioinform* 2008;**9**:40.
9. Wei G-W. Protein structure prediction beyond AlphaFold. *Nat Mach Intell* 2019;**1**(8):336–7.
10. Yang J, Shen HB. MemBrain-contact 2.0: a new two-stage machine learning model for the prediction enhancement of transmembrane protein residue contacts in the full chain. *Bioinformatics* 2018;**34**(2):230–8.
11. Hu J, Li Y, Zhang M, *et al.* Predicting protein-DNA binding residues by weightedly combining sequence-based features and boosting multiple SVMs. *IEEE/ACM Trans Comput Biol Bioinform* 2017;**14**(6):1389–98.
12. Kallberg M, Wang H, Wang S, *et al.* Template-based protein structure modeling using the RaptorX web server. *Nat Protoc* 2012;**7**(8):1511–22.
13. Zhang B, Liu D, Zhang Y, *et al.* Accurate flexible refinement for atomic-level protein structure using cryo-EM density maps and deep learning. *Brief Bioinform* 2022;**23**(2):bbac026.
14. Baek M, DiMaio F, Anishchenko I, *et al.* Accurate prediction of protein structures and interactions using a three-track neural network. *Science* 2021;**373**(6557):871–6.
15. Cozzetto D, Kryshtafovych A, Ceriani M, *et al.* Assessment of predictions in the model quality assessment category. *Proteins* 2007;**69**(Suppl 8):175–83.
16. Cheng J, Choe MH, Elofsson A, *et al.* Estimation of model accuracy in CASP13. *Proteins* 2019;**87**(12):1361–77.
17. Zemla A. LGA: a method for finding 3D similarities in protein structures. *Nucleic Acids Res* 2003;**31**(13):3370–4.
18. Mariani V, Biasini M, Barbato A, *et al.* lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics* 2013;**29**(21):2722–8.
19. Won J, Baek M, Monastyrskyy B, *et al.* Assessment of protein model structure accuracy estimation in CASP13: challenges in the era of deep learning. *Proteins* 2019;**87**(12):1351–60.
20. Chen J, Siu SWI. Machine learning approaches for quality assessment of protein structures. *Biomolecules* 2020;**10**(4):626. https://doi.org/10.3390/biom10040626.
21. Cao R, Bhattacharya D, Adhikari B, *et al.* Large-scale model quality assessment for improving protein tertiary structure prediction. *Bioinformatics* 2015;**31**(12):i116–23.
22. Uziela K, Shu N, Wallner B, *et al.* ProQ3: improved model quality assessments using Rosetta energy terms. *Sci Rep* 2016;**6**:33509.
23. Ray A, Lindahl E, Wallner B. Improved model quality assessment using ProQ2. *BMC Bioinform* 2012;**13**:224.
24. Uziela K, Menéndez Hurtado D, Shu N, *et al.* ProQ3D: improved model quality assessments using deep learning. *Bioinformatics* 2017;**33**(10):1578–80.
25. Cao R, Bhattacharya D, Hou J, *et al.* DeepQA: improving the estimation of single protein model quality with deep belief networks. *BMC Bioinformatics* 2016;**17**(1):495.
26. Sato R, Ishida T. Protein model accuracy estimation based on local structure quality assessment using 3D convolutional neural network. *PLoS One* 2019;**14**(9):e0221347.
27. Pages G, Charmettant B, Grudinin S. Protein model quality assessment using 3D oriented convolutional neural networks. *Bioinformatics* 2019;**35**(18):3313–9.
28. Sanyal S, Anishchenko I, Dagar A, *et al.* ProteinGCN: protein model quality assessment using graph convolutional networks. *bioRxiv* 2020;2020.04.06.028266.
29. Baldassarre F, Menéndez Hurtado D, Elofsson A, *et al.* GraphQA: protein model quality assessment using graph convolutional networks. *Bioinformatics* 2021;**37**(3):360–6.
30. Maghrabi AHA, McGuffin LJ. Estimating the quality of 3D protein models using the ModFOLD7 server. *Methods Mol Biol* 2020;**2165**: 69–81.
31. McGuffin LJ, Aldowsari FMF, Alharbi SMA, *et al.* ModFOLD8: accurate global and local quality estimates for 3D protein models. *Nucleic Acids Res* 2021;**49**(W1):W425–30.
32. Yang J, Wang Y, Zhang Y. ResQ: an approach to unified estimation of B-factor and residue-specific error in protein structure prediction. *J Mol Biol* 2016;**428**(4):693–701.

33. Kwon S, Won J, Kryshtafovych A, *et al.* Assessment of protein model structure accuracy estimation in CASP14: old and new challenges. *Proteins* 2021;**89**(12):1940–8.
34. Liu Q, Hu Z, Jiang R, *et al.* DeepCDR: a hybrid graph convolutional network for predicting cancer drug response. *Bioinformatics* 2020;**36**(Supplement_2):i911–8.
35. Yin Q, Liu Q, Fu Z, *et al.* scGraph: a graph neural network-based approach to automatically identify cell types. *Bioinformatics* 2022;**38**(11):2996–3003.
36. Gligorijevic V, Renfrew PD, Kosciolek T, *et al.* Structure-based protein function prediction using graph convolutional networks. *Nat Commun* 2021;**12**(1):3168.
37. Ma T, Liu Q, Li H, *et al.* DualGCN: a dual graph convolutional network model to predict cancer drug response. *BMC Bioinform* 2022;**23**(Suppl 4):129.
38. Fout AM. Protein Interface Prediction using Graph Convolutional Networks. *Proceedings of NIPS*, 2017;**30**:6533–42.
39. Jing X, Xu J. Fast and effective protein model refinement using deep graph neural networks. *Nat Comput Sci* 2021;**1**(7):462–9.
40. Hiranuma N, Park H, Baek M, *et al.* Improved protein structure refinement guided by deep learning based accuracy estimation. *Nat Commun* 2021;**12**(1):1340.
41. Hippe K, Lilley C, William Berkenpas J, *et al.* ZoomQA: residue-level protein model accuracy estimation with machine learning on sequential and 3D structural features. *Brief Bioinform* 2022;**23**(1):bbab384. https://doi.org/10.1093/bib/bbab384.
42. Jones DT, Thornton JM. The impact of AlphaFold2 one year on. *Nat Methods* 2022;**19**(1):15–20.
43. Chowdhury R, Bouatta N, Biswas S, *et al.* Single-sequence protein structure prediction using a language model and deep learning. *Nat Biotechnol* 2022;**40**(11):1617–23.
44. Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 1983;**22**(12):2577–637.