GRAPHQA: PROTEIN MODEL QUALITY ASSESSMENT USING GRAPH CONVOLUTIONAL NETWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Proteins are ubiquitous molecules whose function in biological processes is determined by their 3D structure. Experimental identification of a protein's structure can be time-consuming, prohibitively expensive, and not always possible. Alternatively, protein folding can be modeled using computational methods, which however are not guaranteed to always produce optimal results.
GRAPHQA is a graph-based method to estimate the quality of protein models, that possesses favorable properties such as representation learning, explicit modeling of both sequential and 3D structure, geometric invariance and computational efficiency. In this work, we demonstrate significant improvements of the state-of-the-art for both hand-engineered and representation-learning approaches, as well as carefully evaluating the individual contributions of GRAPHQA.

1 INTRODUCTION

Protein molecules are predominantly present in biological forms, responsible for their cellular functions. Therefore, understanding, predicting and modifying proteins in biological processes are essential for medical, pharmaceutical and genetic research. Such study is mainly focused on discovering proteins' mechanical and chemical properties through the determination of their structure.

At the high level, a protein molecule is a chain of hundreds of smaller molecules called amino acids. Identifying a protein's amino-acid sequence is nowadays straightforward. However, the function of a protein is primarily determined by its 3D structure. Spatial folding can be experimentally determined, but the existing procedures are time-consuming, prohibitively expensive and not always possible. Thus, several computational techniques are developed for protein structure prediction (Arnold et al., 2006; Xu, 2019). So far, no single method is always best, i.e. different proteins are best modeled by different methods. Moreover, many methods produce multiple models. Therefore, there is a need to evaluate the models after their generation. This work focuses on Quality Assessment (QA) of computationally-derived models of a protein to identify the best one.

QA, also referred to as model accuracy estimation (MAE), estimates the quality of computational protein models in terms of their divergence from the native structure. The downstream goal of QA is two-fold; to find the best model in a pool of models and to refine a model based on its local quality.

While computational protein folding has recently received attention from the machine learning community (Ingraham et al., 2019b; Anand & Huang, 2018; Evans et al., 2018), QA has not. This is despite the importance of QA for structural biology and the availability of standard datasets to benchmark machine learning techniques, such as the biannual CASP event (Won et al., 2019). The field of bioinformatics, on the other hand, has witnessed noticeable progress in QA for more than a decade. With earlier works using support vector machines (Ray et al., 2012) and recently adopting deep learning methods such as LSTM (Conover et al., 2019), 1D CNN (Hurtado Menendez et al., submitted), and 3D CNNs (Derevyanko et al., 2018; Pagès et al., 2018).

In this work, we apply Graph Convolutional Networks to QA, which bring several desirable properties over previous methods and, through extensive set of experiments, we show significant improvements over the state-of-the-art, and offer informative qualitative and quantitative analyses.

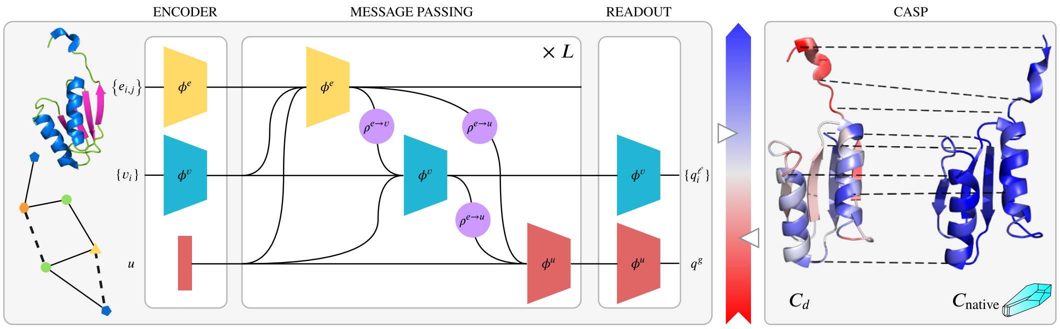


Figure 1: **Protein Quality Assessment.** GRAPHQA predicts local and global scores from a protein's graph using message passing among residues with chemical bond or spatial proximity. CASP QA algorithms score protein models by comparison with experimentally-determined conformations.

1.1 RELATED WORKS

Protein Quality Assessment (QA) methods are evaluated in CASP (Moult et al., 1995) since CASP7 (Cozzetto et al., 2007). Current techniques can be divided into two categories, single-model methods (Wallner & Elofsson, 2003) which operate on a single protein model to estimate its quality and consensus methods (Lundstrom et al., 2001) that use consistency of several protein models to estimate the quality of each model. Single-model methods are applicable to a single protein in isolation and in the recent CASP13, performed comparable to or better than consensus methods for the first time (Cheng et al., 2019). Recent single-model QA works are based on deep learning except VoroMQA that takes a statistical approach on atom-level contact area (Olechnovic & Venclovas, 2017). 3D-CNN (Derevyanko et al., 2018) adopts a volumetric representation of proteins. Ornate improves 3D-CNN by defining a canonical orientation (Pagès et al., 2018). ProQ3D (Uziela et al., 2017) uses a multi-layer perceptron on fixed-length protein descriptors. ProQ4 (Hurtado et al., 2018a) adopts a pre-trained 1D CNN that is fine-tuned in a siamese configuration with a rank loss. VoroMQA and ProQ3D are among the top performers of CASP13 (Won et al., 2019).

Graph Convolutional Networks (GCNs) bring the representation learning power of CNNs to graph data, and have been recently applied with success to multiple domains, e.g. physics (Gonzalez et al., 2018), visual scene understanding (Narasimhan et al., 2018) and natural language understanding (Kipf & Welling, 2017). Molecules can be naturally represented as graphs and GCNs have been proven effective in several related tasks including molecular representation learning (Duvenaud et al., 2015), protein interface prediction (Fout et al., 2017), chemical property prediction (Niepert et al., 2016; Gilmer et al., 2017; Li et al., 2018a), drug-drug interaction (Zitnik et al., 2018), drug-target interaction (Gao et al., 2018) molecular optimization (Jin et al., 2019), and generation of proteins, molecules and drugs (Ingraham et al., 2019a; You et al., 2018; Liu et al., 2018; Li et al., 2018b; Simonovsky & Komodakis, 2018). However, to the best of our knowledge, GCNs have never been applied to the problem of protein quality assessment.

1.2 CONTRIBUTIONS

- This work is the first to tackle QA with GCN which bring several desirable properties including representation learning (3DCNN, Ornate), geometric invariance (VoroMQA, Ornate), sequence learning (ProQ4, AngularQA), explicit modeling of 3D structure (3DCNN, Ornate, VoroMQA) and computational efficiency.
- Thanks to these desirable properties, a simple GCN setup achieves improved results compared to the more sophisticated state-of-the-art methods such as ProQ4. This is demonstrated through extensive experiments on multiple datasets and scoring regimes.
- Novel representation techniques are employed to explicitly reflect the sequential (residue separation) and 3D structure (angles, spatial distance and secondary structure) of proteins.
- Enabled by the use of GCN, we combine the optimization of local and global score for QA improving over the performance of a global-only scoring method.
- Through an extensive set of ablation studies, the significance of different components of the method, including architecture, loss, and features, are carefully analyzed.

Anonymous git repository:
`https://anonymous.4open.science/r/94d976ce-9166-4379-b3c4-3c982752a931`

2 METHOD

We start describing our method by arguing for representation of protein molecules as graphs in learning tasks, then we define the problem of protein quality assessment (QA), and finally we present the proposed GRAPHQA architecture.

2.1 PROTEIN REPRESENTATION AS GRAPHS

Proteins are large molecular structures that perform vital functions in all living organisms. At the chemical level, a protein consists of one or more chains of smaller molecules, which we interchangeably refer to as **residues** for their role in the chain, or as **amino acids** for their chemical composition. The sequence of residues $S = \{ a_i \}$ that composes a protein represents its **primary structure**, where a_i is one of the 22 amino acid types. The interactions between neighboring residues and with the environment dictate how the chain will fold into complex structures that represent the protein's **secondary structure** and **tertiary structure**.

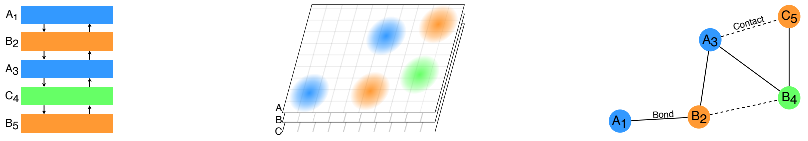
Therefore, for learning tasks involving proteins, a suitable representation should reflect both the identity and sequence of the residues, i.e. the primary structure, and geometric information about the protein's arrangement in space, i.e. its tertiary structure. Some works (Hurtado et al., 2018b; Conover et al., 2019) use RNN or 1D-CNN to model proteins as sequence with the spatial structure potentially embedded in the handcrafted residue features. Other recent works (Derevyanko et al., 2018; Pagès et al., 2018) explicitly model proteins' spatial structure using 3D-CNN but ignore its sequential nature. We argue that a graph-based learning can explicitly model both the sequential and geometric structures of proteins. Moreover, it accommodates proteins of different lengths and spatial extent, and is invariant to rotations and translations.

In the simplest form, a protein can be represented as a linear graph, where nodes represent amino acids and edges connect consecutive residues according to the primary structure. This set of edges, which represent the covalent **bonds** that form the protein backbone, can be extended to include the interactions between non-consecutive residues, e.g. through Van der Waals forces or hydrogen bonds, commonly denoted as **contacts**. By forming an edge between all pairs of residues that are within a chemically reasonable distance of each other, the graph becomes a rich representation of both the primary and tertiary structure of the protein (figure 2). We refer to this representation, composed of residues, bonds and contacts, as the **protein graph**:

$$\mathcal{P} = \Big(\{ \boldsymbol{v}_i \}, \quad \{ \boldsymbol{e}^{\text{bond}}_{i,j} \mid |i-j| = 1 \} \cup \{ \boldsymbol{e}^{\text{contact}}_{i,j} \mid |i-j| > 1, \|C_i - C_j\| \leq d_{\text{max}} \} \Big), \qquad (1)$$

where **conformation** $C = \{ (x,y,z)_i \}$ is the spatial arrangement of the residues, i.e. its tertiary structure, $i, j = 1, \ldots, |S|$ represent residue indices and d_{max} is a cutoff distance for contacts.

With the protein's structure encoded in the graph, residue and relationship features can be represented as nodes and edges attributes, $\boldsymbol{v}_i$ and $\boldsymbol{e}_{i,j}$ respectively. Section 3.2 describes, in detail, an attribution that preserves the sequence information and 3D geometry while remaining invariant to rotation.



(a) Sequential (b) Spatial (c) Graph

Figure 2: **Protein representations for learning.** Sequential representations for LSTM or 1D-CNN fail to represent spatial proximity of non-consecutive residues. Volumetric representations for 3D-CNN fail instead to capture sequence information and is not rotation invariant. Protein graphs explicitly represent both sequential and spatial structure, and are geometrically invariant by design.

2.2 Protein Quality Assessment

Experimental identification of a protein's **native structure** can be time consuming and prohibitively expensive. Alternatively, computational folding methods are used to generate **decoy** conformations for a specific **target** protein. Since no single method is consistently best, a Quality Assessment step is used to identify the decoys that most correctly represent the **native structure**

If the native structure C_{native} is experimentally determined, the quality of a decoy can be measured by comparison (Uziela et al., 2018), e.g., in the CASP challenge, decoys submitted for a target are scored against the unreleased native structure. Some comparative algorithms compute global (per decoy) scores, which can be used for ranking and represent the principal factor for CASP, while others produce local (per residue) scores which help identify incorrect parts of a decoy.

In most scenarios, however, the native structure is not available and quality must be estimated based on decoy properties of the decoy, e.g., in drug development, it would be unpractical to synthesize samples of novel proteins and researchers rely on folding and quality assessment instead.

Here we introduce GraphQA, a graph-based QA neural network that learns to predict local and global scores, with minimal feature and model engineering, using existing datasets of scored proteins. In this paper, we train GraphQA on two widely-used scoring algorithms: the Global Distance Test Total Score (Zemla, 2003), which is the official CASP score for decoy-level quality assessment, and the Local Distance Difference Test (Mariani et al., 2013), a residue-level score. We denote them as $q^g := \text{GDT\_TS}(C^d, C^{\text{native}})$ and $\{\, q_i^\ell \,\} := \text{LDDT}(C^d, C^{\text{native}})$.

With $\text{GraphQA}_i^\ell(\mathcal{P})$ and $\text{GraphQA}^g(\mathcal{P})$ denoting the network's local and global predictions, the learning objective is to minimize the following two Mean Squared Error (MSE) losses for each decoy:

$$\mathcal{L}_\ell = \sum_i^{|S|} \left[\text{GraphQA}_i^\ell(\mathcal{P}) - q_i^\ell\right]^2 \qquad\qquad \mathcal{L}_g = \left[\text{GraphQA}^g(\mathcal{P}) - q^g\right]^2. \tag{2}$$

Note that, for the sole purpose of sorting decoy according to ground-truth quality, training with a ranking loss would be sufficient (e.g. Derevyanko et al. (2018)). Instead, MSE forces output to match the quality score, which is a harder objective, but results in a network can be more easily inspected and possibly used to improve existing folding methods in an end-to-end fashion (section 4.2).

2.3 GraphQA Architecture

GraphQA is a graph convolutional network that operates on protein graphs using the message-passing algorithm described in Battaglia et al. (2018). The building block of GraphQA, a graph layer, takes a protein graph as input (with an additional global feature $\boldsymbol{u}$), and performs the following propagation steps to output a graph with updated node/edge/global features and unchanged structure:

$$\begin{array}{rlrl}
\boldsymbol{e}'_{i,j} = \phi^e\left(\boldsymbol{e}_{i,j}, \boldsymbol{v}_i, \boldsymbol{v}_j, \boldsymbol{u}\right) & \text{Update edges} & \bar{\boldsymbol{e}}' = \rho^{e\to u}\left(\{\, \boldsymbol{e}'_{i,j} \,\}\right) & \text{Aggregate all edges} \\
\bar{\boldsymbol{e}}'_i = \rho^{e\to v}\left(\{\, \boldsymbol{e}'_{i,j} \,\}\right) & \text{Aggregate edges} & \bar{\boldsymbol{v}}' = \rho^{v\to u}\left(\{\, \boldsymbol{v}'_i \,\}\right) & \text{Aggregate all nodes} \\
\boldsymbol{v}'_i = \phi^v\left(\bar{\boldsymbol{e}}'_i, \boldsymbol{v}_i, \boldsymbol{u}\right) & \text{Update nodes} & \boldsymbol{u}' = \phi^u\left(\bar{\boldsymbol{e}}', \bar{\boldsymbol{v}}', \boldsymbol{u}\right) & \text{Update global features}
\end{array}$$

where ϕ represent three update functions that transform nodes/edges/global features, and ρ represent three aggregation functions that aggregate features at various levels.

Similarly to convolutional layers, multiple graph layers are stacked allowing local information to propagate to increasingly larger neighborhood (i.e. receptive field), thus enabling the network to learn quality-related features at multiple scales: secondary structures in the first layers, e.g. α-helices and β-sheets, and larger structures in deeper layers e.g. domain structures and arrangements.

Layers in GraphQA are conceptually divided into three groups, see figure 1. **Encoder** increases the node and edge features' dimensions through 2×(Linear-Dropout-ReLU) transformation and adds a global bias. Then, at the core of GraphQA, L **message-passing** layers operate on the encoded graph leveraging its structure to propagate and aggregate information. The update functions ϕ consist of Linear-Dropout-ReLU transformations, with the size of the linear layers progressively decreasing. We use average pooling for the aggregation functions ρ, since preliminary experiments with max/sum pooling performed poorly. Finally, the **readout** layer outputs local and global quality scores by applying a Linear-Sigmoid operation to the latest node and global features, respectively.

3 Experiments

3.1 Experimental Setup

Following the common practice in Quality Assessment, we use the data from past years' editions of CASP, encompassing several targets with multiple scored decoys each. Removing all sequences with $|S| < 50$ from CASP 7-10 results in a dataset of $\sim$100k scored decoys $(\mathcal{P}, \{ q_i^\ell \}, q^g)^{t,d}$, which we randomly split into a training set (402 targets) and a validation set for hyper-parameter optimization (35 targets). CASP 11 and 12 are set aside for testing against top-scoring methods (table 3).

We evaluate GraphQA against state-of-the-art QA methods on the following metrics. At the global level, we compare the predicted and ground-truth GDT_TS scores and report Root Mean Squared Error (RMSE), Pearson correlation coefficient computed across all decoys of all targets (R), Pearson correlation coefficient computed on a per-target basis and then averaged over all targets (R_{target}).
At the local level, we compare the predicted and ground-truth LDDT scores and report: RMSE, the Pearson correlation coefficient computed across all residues of all decoys of all targets (R), and the Pearson correlation coefficient computed on a per-decoy basis and then averaged over all decoys of all targets (R_{model}). Of these, we focus on R_{target}, that measures the ability to and R_{model}, since they respectively measure the ability to distinguish the correctly-predicted parts of a model from those that need improvement. A description of these and other metrics can be found in appendix F.

3.2 Features

Node features The node attributes $\boldsymbol{v}_i$ of a protein graph $\mathcal{P}$ represent the identity, statistical, and structural features of the i-th residue. We encode the residue identity by a one-of-22 encoding of the corresponding amino acid. Following Hurtado et al. (2018b), we also add two residue-level statistics computed using Multiple Sequence Alignment (MSA) (Rost et al., 1994), namely **self-information** and **partial entropy**, each described by a 23-dimensional vector. Finally, we add a 14-dimensional vector of 3D spatial information including the dihedral angles, surface accessibility and secondary structure type as determined by DSSP (Kabsch & Sander, 1983).

Edge features An edge $\boldsymbol{e}_{i,j}$ represents either a contact or a bond between two residues i and j w.r.t. to the conformation $C = \{ (x, y, z)_i \}$. An edge always exists between two consecutive residues, while non-consecutive residues are only connected if $||C_i - C_j|| < d_{\text{max}}$ with d_{max} optimized on the validation set. We further enrich this connectivity structure by an 8D edge feature vector encoding spatial and sequential distances. Spatial distance is encoded by a radial basis function $\exp(-d_{i,j}^2/\sigma)$, with σ determined on the validation set. Sequential distance is defined as the number of amino acids between the two residues in the sequence and expressed using a **separation encoding**, i.e. a one-hot encoding of the separation $|i - j|$ according to the classes $\{ 0, 1, 2, 3, 4, 5 : 10, > 10 \}$.

3.3 Optimization and Hyperparameter Search

The MSE losses in equation 2 are weighted as $\mathcal{L}_{tot} = \lambda_\ell \mathcal{L}_\ell + \lambda_g \mathcal{L}_g$ and minimized using Adam Optimizer (Kingma & Ba, 2014) with L_2 regularization. GraphQA is significantly faster to train than LSTM or 3D-CNN methods, e.g. 35 epochs takes $\sim$2 hours on one NVIDIA 2080Ti GPU with batches of 200 graphs. This allows us to perform extensive hyper-parameter search. Table 4 reports the search space, as well as the parameters of the model with highest R_{target} on the validation set.

4 Evaluation

We compare GraphQA with the following methods, either for their state-of-the-art performances or because they represent a class of approaches for Quality Assessment. ProQ3D (Uziela et al., 2017) computes fixed-size statistical descriptions of the decoys in CASP 9-10, including Rosetta energy terms, which are then used to train a Multi Layer Perceptron on quality scores. In ProQ4 (Hurtado Menendez et al., submitted), a 1D-CNN is trained to predict residue-level LDDT scores from a vectorized representation of protein sequences, a global score is then obtained by averaging over all residues. The CNN is pretrained on a large dataset of protein secondary structures and then fine tuned on CASP 9-10 using a siamese configuration to improve ranking performances. Their results are reported on both CASP 11, which is used as a validation set, and CASP 12. 3DCNN (Derevyanko

Table 1: **Comparison of state-of-the-art QA methods.** At the residue level we compare LDDT scores and report Pearson correlation and Pearson correlation per model. At the global level we compare GDT_TS scores and report Pearson correlation and Pearson correlation per target.

	CASP 11				CASP 12			
	GDT_TS		LDDT		GDT_TS		LDDT	
	R	R_{target}	R	R_{model}	R	R_{target}	R	R_{model}
ProQ3D	.772	.452	.84	**.61**	.806	.609		
ProQ4			.77	.56			.772	.516
VoroMQA	.651	.457			.605	.559		
Rwplus		.206			-.096	.417		
AngularQA	.651	.439						
3D CNN	.629	.421				.607		
Ornate	.637	.386			.670	.491		
GraphQA	**.910**	**.740**	**.855**	**.610**	**.843**	**.745**	**.843**	**.573**
GraphQA$_{\text{RES}}$	.836	.609	.799	.529	.816	.673	.796	.507

et al., 2018) trains a CNN on a three-dimensional representation of atomic densities to rank the decoys in CASP 7-10 according to their GDT_TS scores. The fixed-size volumetric representation of this method is sensitive to rotations and does not scale well with protein size, but has the benefit of using no additional feature other than the atomic coordinates. Ornate (Pagès et al., 2018) applies a similar 3D approach to predict to predict local CAD-scores (Olechnovic & Venclovas, 2017) and achieves rotation invariance by specifying a canonical residue-centered orientation. Although optimized for local scoring, the average of the predicted scores is shown to correlate well with GDT_TS. AngularQA (Conover et al., 2019) feeds a sequence-like representation of the protein structure to an LSTM to predict GDT_TS scores. The LSTM network is trained on decoys from 3DRobot and CASP 9-11, while CASP 12 is used for model selection and testing. VoroMQA and RWplus (Olechnovič & Venclovas, 2017; Zhang & Zhang, 2010) are two statistical potential methods that represent an alternative to the other machine-learning based methods.

Table 1 compares the performances of GraphQA and other state-of-the-art methods on CASP 11 and 12, while figure 3 contains a graphical representation of true vs. predicted scores for all target in CASP 12, and an example funnel plot for the decoys of a single target. A more in-depth breakdown of the evaluation on the stage 1 and stage 2 splits of CASP 11 and 12 can be found in appendix F.

Of all methods, only GraphQA and ProQ4 co-optimize for local and global predictions, the former thanks to the graph-based architecture, the latter thanks to its siamese training configuration (the results reported for ProQ3D refer to two separate models trained for either local or global scores). At the local level, our method proves to be on par or better than ProQ3D and ProQ4, demonstrating the ability to evaluate quality at the residue level and distinguishing correctly predicted parts of the protein chain. At the global level, significantly higher R and R_{target} metrics indicate than GraphQA is more capable than other state-of-the-art methods at ranking decoys based on their overall quality.

As it is common, GraphQA relies on hand-engineered features like MSA and DSSP (section 5), yet we further prove that our method can learn directly from raw data. GraphQA$_{\text{RAW}}$ is a variant that relies uniquely on the one-hot encoding of amino acid identity, similarly to how 3D-CNN and Ornate employ atom identities only. The results for GraphQA$_{\text{RAW}}$ show that, even without additional features, our method outperforms purely representation learning methods.

4.1 Ablation Studies

In this section we analyse how various components of our method contribute to the final performance, ranging from optimization and architectural choices to protein feature selection. Unless stated otherwise, all ablation studies follow the training procedure described in section 3.3 for a lower number of epochs. We report results on CASP 11 as mean and standard deviation of 10 runs.

Local and global co-optimization We investigate the interplay between local and global predictions, specifically whether co-optimizing for both is beneficial or detrimental. At the global level, models trained to predict only global scores achieve a global RMSE of $0.129 \pm .007$, whereas models trained to predict both local and global scores obtain $0.117 \pm .006$, suggesting that local scores can provide additional information and help the assessment of global quality. At the local level instead, co-optimization does not seem to improve performances: models trained uniquely on local scores achieve a local RMSE of $0.121 \pm .002$, while models trained to predict both obtain $0.123 \pm .004$.

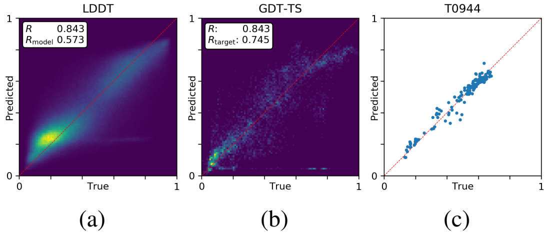
Connectivity and Architecture In this study, we test the combined effects of the depth of the network L and the cutoff value d_{max}. On the one hand, every additional message-passing layer allows to aggregate information from a neighborhood that is one hop larger than the previous, effectively extending the receptive field at the readout. On the other hand, the number of contacts included in the graph affects its connectivity and the propagation of messages, e.g. low d_{max} correspond to a low average degree and long shortest paths between any two residues, and vice versa. Therefore, to make predictions based on an holistic view of the protein, an architecture that operates on sparsely connected graphs will require more message-passing layers, while fewer layers are needed for denser representations. We noticed however, that this trade off is only properly exposed if $\boldsymbol{u}, \phi^u, \rho^u$ are removed from the architecture. In fact, this global pathway represents a propagation shortcut that connects all nodes in the graph and sidesteps the limitations of shallow networks. With the global pathway disabled, global predictions are computed in the readout layer by aggregating node features from the last MP layer.
Figure 4 reports the RMSE obtained by networks of different depth with no global path, operating on graphs constructed with different cutoff values. As expected, the shallow 3-layer architecture requires denser graphs to achieve the same performances of the 9-layer network. Surprisingly, local predictions seem to be more affected by these factors than global predictions, suggesting that a large receptive field is important even for local scores.

Node and Edge Features We evaluate the impact of node and edge features on the overall prediction performances (figure 5). For the nodes, we use the amino acid identity as a minimal representation and combine it with: a) DSSP features, b) partial entropy, c) self information, d) both DSSP and MSA features. All features improve both local and global scoring, with DSSP features being marginally more relevant for LDDT. For the edges, we evaluate the effect of having either: a) a binary indicator of bond/contact, b) geometric features, i.e. the euclidean distance between residues, c) sequential features, i.e. the categorical encoding of the separation between residues, d) both distance and separation encoding. If the addition of edge features seem to be benefit LDDT predictions, little improvement can be seen at the global level.

4.2 Visualization and Explainability

The design of GraphQA makes it suitable not only for scoring, but also to identify refinement opportunities for computationally created decoys. Figure 6 shows a decoy that correctly models the native structure of its target, but contains imperfections at one extremity, to which both GraphQA and LDDT assign low local scores. Unlike LDDT, however, GraphQA is fully differentiable and the trained model can be used to explain the factors that influenced a low score and provide useful feedback for computational structure prediction.

A simple approach for explaining predictions of a differentiable function $f(\boldsymbol{x})$ is Sensitivity Analysis (Baehrens et al., 2010; Simonyan et al., 2014), which uses $\|\nabla_{\boldsymbol{x}} f\|$ to measure how variations in the input affect the output. In figure 6 we consider the scores predicted for two different residues and compute the magnitude of the gradients w.r.t. the edges of the graph. Assuming that large mag-



(a) (b) (c)

Figure 3: Histograms of true vs. predicted LDDT (a) and GDT_TS (b) scores on CASP 12, (c) funnel plot of the decoys of target T0944 (PDB 5ko9).

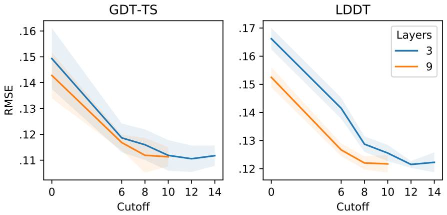


Figure 4: Trade-off between number of message-passing layers and connectivity of the protein graph.

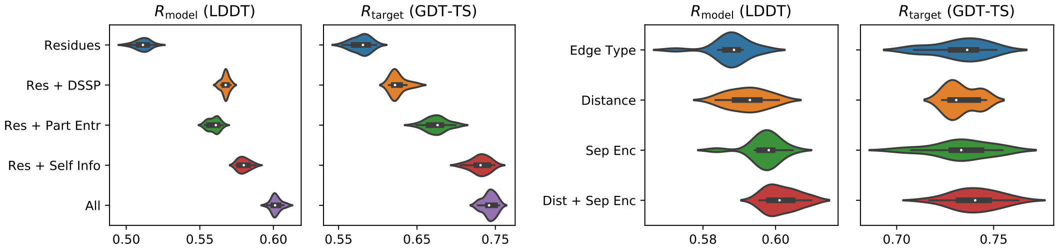
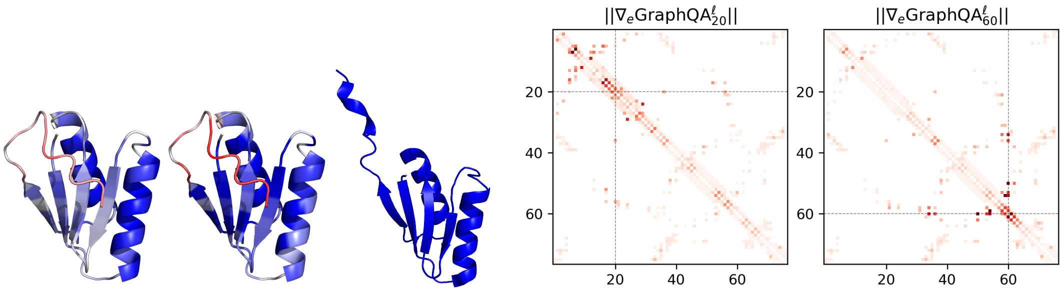


Figure 5: Ablation study of node (left) and edge (right) feature representation.

nitudes correspond to large errors in the input, it is interesting to note how the network is able to capture dependencies not only in the neighborhood of the selected residues, but also further away in the sequence.

Furthermore, we wanted to measure whether the global predictions of GraphQA could be used to improve the contact maps used by computational methods to build protein models. If the networks has learned a meaningful scoring function, then the gradient of the score w.r.t. the contact distances should aim in the direction of the native structure, indicating how the decoy contacts should be updated to improve its score. Considering all decoys of all targets in CASP 11, we obtain an average cosine similarity $\cos\left(\partial \text{GraphQA}^g / \partial \boldsymbol{d}, \boldsymbol{d}_{\text{decoy}} - \boldsymbol{d}_{\text{native}}\right)$ of $0.14 \pm .08$, which, represents a step in the direction of end-to-end protein model prediction.



(a) Predicted (b) True (c) Native (d) (e)

Figure 6: **One decoy of T0773 in CASP11.** Both GraphQA (a) and LDDT (b) assign low local scores to a segment of the decoy, highlighting a discrepancy w.r.t. the native structure (c). The gradient magnitude w.r.t. the edges of the predicted LDDT score for residues 20 and 60 reveal long range dependencies inside the protein graph.

5 Conclusion

For the first time we applied graph convolutional networks to the important problem of protein quality assessment (QA). Since proteins are naturally represented as graphs, GCN allowed us to collect the individual benefits of the previous QA methods including representation learning, geometric invariance, explicit modeling of sequential and 3D structure, simultaneous local and global scoring, and computational efficiency. Thanks to these benefits, and through an extensive set of experiments, we demonstrated significant improvements upon the state-of-the-art results on various metrics and datasets and further analyzed the results via thorough ablation and qualitative studies.

Finally, we wish that Quality Assessment will gain popularity in the machine learning community, that could benefit from several curated datasets and ongoing regular challenges. We believe that richer geometric representations, e.g. including relative rotations, and raw atomic representations could represent an interesting future direction for learning-based Quality Assessment.

REFERENCES

Namrata Anand and Possu Huang. Generative modeling for protein structures. In *Advances in Neural Information Processing Systems*, pp. 7494–7505, 2018.

Konstantin Arnold, Lorenza Bordoli, Jürgen Kopp, and Torsten Schwede. The swiss-model workspace: a web-based environment for protein structure homology modelling. *Bioinformatics*, 22(2):195–201, 2006.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

J. Cheng, M.H. Choe, A. Elofsson, K.S. Han, J. Hou, A.H.A. Maghrabi, L.J. McGuffin, D. Menendez-Hurtado, K. Olechnovic, T. Schwede, G. Studer, K. Uziela, C. Venclovas, and B. Wallner. Estimation of model accuracy in CASP13. *Proteins*, Jul 2019. doi: 10.1002/prot.25767.

Matthew Conover, Max Staples, Dong Si, Miao Sun, and Renzhi Cao. Angularqa: protein model quality assessment with lstm networks. *Computational and Mathematical Biophysics*, 7(1):1–9, 2019.

D. Cozzetto, A. Kryshtafovych, M. Ceriani, and A. Tramontano. Assessment of predictions in the model quality assessment category. *Proteins*, 69 Suppl 8:175–183, 2007. doi: 10.1002/prot.21669.

G Derevyanko, S Grudinin, Y Bengio, and G Lamoureux. Deep convolutional networks for quality assessment of protein folds. *Bioinformatics (Oxford, England)*, 34(23):4046–4053, 2018.

David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pp. 2224–2232, 2015.

R Evans, J Jumper, J Kirkpatrick, L Sifre, T Green, C Qin, A Zidek, A Nelson, A Bridgland, H Penedones, et al. De novo structure prediction with deeplearning based scoring. *Annu Rev Biochem*, 77:363–382, 2018.

Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using graph convolutional networks. In *Advances in Neural Information Processing Systems*, pp. 6530–6539, 2017.

Kyle Yingkai Gao, Achille Fokoue, Heng Luo, Arun Iyengar, Sanjoy Dey, and Ping Zhang. Interpretable drug target prediction using deep neural representation. In *IJCAI*, pp. 3371–3377, 2018.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1263–1272. JMLR. org, 2017.

Alvaro Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. In *International Conference on Machine Learning (ICML)*, pp. 4467–4476, 2018.

David Menéndez Hurtado, Karolis Uziela, and Arne Elofsson. Deep transfer learning in the assessment of the quality of protein models. *arXiv preprint arXiv:1804.06281*, 2018a.

David Menéndez Hurtado, Karolis Uziela, and Arne Elofsson. Deep transfer learning in the assessment of the quality of protein models. *arXiv preprint arXiv:1804.06281*, 2018b.

David Hurtado Menendez, Karolis Uziela, and Arne Elofsson. Transfer learning for quality assessment of protein models. *Bioinformatics*, submitted.

John Ingraham, Vikas K Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. *NeurIPS*, 2019a.

John Ingraham, Adam Riesselman, Chris Sander, and Debora Marks. Learning protein structure with a differentiable simulator. *ICLR*, 2019b.

Wengong Jin, Kevin Yang, Regina Barzilay, and Tommi Jaakkola. Learning multimodal graph-to-graph translation for molecular optimization. *ICLR*, 2019.

Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, 22(12):2577–2637, 1983.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. URL `https://openreview.net/forum?id=SJU4ayYgl`.

Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018a.

Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*, 2018b.

Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. Constrained graph variational autoencoders for molecule design. In *Advances in Neural Information Processing Systems*, pp. 7795–7804, 2018.

J. Lundstrom, L. Rychlewski, J. Bujnicki, and A. Elofsson. Pcons: a neural-network-based consensus predictor that improves fold recognition. *Protein Sci*, 10(11):2354–2362, Nov 2001.

Valerio Mariani, Marco Biasini, Alessandro Barbato, and Torsten Schwede. lddt: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, 29(21):2722–2728, 2013.

J. Moult, J.T. Pedersen, R. Judson, and K. Fidelis. A large-scale experiment to assess protein structure prediction methods. *Proteins*, 23(3):ii–v, Nov 1995. doi: 10.1002/prot.340230303.

Medhini Narasimhan, Svetlana Lazebnik, and Alexander Schwing. Out of the box: Reasoning with graph convolution nets for factual visual question answering. In *Advances in Neural Information Processing Systems NIPS*, pp. 2654–2665, 2018.

Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pp. 2014–2023, 2016.

K. Olechnovic and C. Venclovas. VoroMQA: Assessment of protein structure quality using interatomic contact areas. *Proteins*, 85(6):1131–1145, Jun 2017. doi: 10.1002/prot.25278.

Kliment Olechnovič and Česlovas Venclovas. Voromqa: Assessment of protein structure quality using interatomic contact areas. *Proteins: Structure, Function, and Bioinformatics*, 85(6):1131–1145, 2017.

Guillaume Pagès, Benoit Charmettant, and Sergei Grudinin. Protein model quality assessment using 3d oriented convolutional neural networks. *bioRxiv*, pp. 432146, 2018.

Arjun Ray, Erik Lindahl, and Björn Wallner. Improved model quality assessment using proq2. *BMC Bioinformatics*, 13(1):224, 2012.

Burkhard Rost, Chris Sander, and Reinhard Schneider. Redefining the goals of protein secondary structure prediction. *Journal of molecular biology*, 235(1):13–26, 1994.

Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *International Conference on Artificial Neural Networks*, pp. 412–422. Springer, 2018.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014. URL `http://arxiv.org/abs/1312.6034`.

K Uziela, Hurtado D Menéndez, N Shu, B Wallner, and A Elofsson. Proq3d: improved model quality assessments using deep learning. *Bioinformatics (Oxford, England)*, 33(10):1578, 2017.

Karolis Uziela, David Menéndez Hurtado, Nanjiang Shu, Björn Wallner, and Arne Elofsson. Improved protein model quality assessments by changing the target function. *Proteins: Structure, Function, and Bioinformatics*, 86(6):654–663, 2018.

B. Wallner and A. Elofsson. Can correct protein models be identified? *Protein Sci*, 12(5):1073–1086, May 2003. doi: 10.1110/ps.0236803.

Jonghun Won, Minkyung Baek, Bohdan Monastyrskyy, Andriy Kryshtafovych, and Chaok Seok. Assessment of protein model structure accuracy estimation in casp13: Challenges in the era of deep learning. *Proteins: Structure, Function, and Bioinformatics*, 2019.

Jinbo Xu. Distance-based protein folding powered by deep learning. *Proceedings of the National Academy of Sciences*, 116(34):16856–16865, 2019.

Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems (NeurIPS) 31*, 2018.

Adam Zemla. Lga: a method for finding 3d similarities in protein structures. *Nucleic acids research*, 31(13):3370–3374, 2003.

Jian Zhang and Yang Zhang. A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction. *PloS one*, 5(10): e15386, 2010.

Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 2018.

A Protein Quality Assessment

For the interested reader, we describe here in more detail how the Global Distance Test Total Score (Zemla, 2003) and the Local Distance Difference Test (Mariani et al., 2013) are computed. Furthermore, we provide an intuition over what the benefits and downsides of each method are and motivate why a better quality assessment should consider both a global measure and a local measure.

Global Distance Test Total Score (GDT_TS) Global Distance Test Total Score (GDT_TS) is a global-level score obtained by first superimposing the structure of a decoy to the experimental structure using an alignment heuristic, and then computing the fraction of residues whose position is within a certain distance from the corresponding residue in the native structure (figure 7). This percentage is computed at different thresholds and then averaged to produce a score in the range $[0, 100]$, which we normalize between 0 and 1 (table 2).

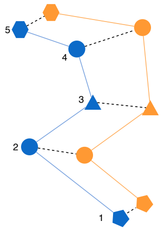


Figure 7: GDT_TS

Table 2

i	$\|C_i^d - C_i^{\text{native}}\|$	< 1	< 2	< 5	< 10
1	0.6 Å	x	x	x	x
2	1.2 Å		x	x	x
3	1.9 Å		x	x	x
4	2.5 Å			x	x
5	6.3 Å				x
		20%	60%	80%	100%

Local Distance Difference Test (LDDT) Local Distance Difference Test (LDDT), is a residue-level score that does not require alignment of the structures and compares instead the local neighborhood of every residue, in the decoy and in the native structure. If we define the neighborhood of a residue as the set of its contacts, i.e. the set of other residues that lie within a certain distance from it, we can express the quality of that residue as the percentage of contacts that it shares with the corresponding residue in the native structure.

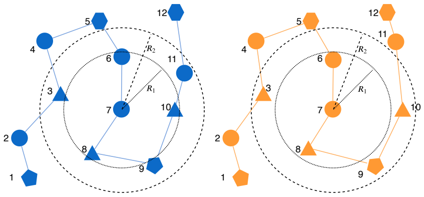


Figure 8: Example of LDDT scoring for residue 7: the residues within a radius R_1 are $\{6, 8, 10\}$ the native structure (left) and $\{6, 8\}$ for the decoy (right); at a radius R_2 we have $\{3, 6, 8, 9, 10, 11\}$ the native structure (left) and $\{3, 6, 8, 9, 10\}$ for the decoy (right).

B Protein Graphs Statistics

C Datasets

We consider all decoys of all target included in CASP 7-12, excluding proteins whose sequence is shorter than 50 residues and targets that have been canceled by the organizers.

Table 3: Datasets from previous CASP editions

Dataset	**Targets**	**Models**	**Usage**
CASP 7	95	19591	Train/Val
CASP 8	122	34789	
CASP 9	117	34946	
CASP 10	103	26254	
CASP 11	83	16094	Test
CASP 12	40	6924	
CAMEO	676	20891	

D Hyperparameter optimization

We perform a guided grid search over the following hyper parameter space. The final model is chosen to be the one with the highest R_{target} on the validation set. The following considerations were made:

- The values for d_{max} are chosen on the base that the typical bond length is $\sim 5\text{Å}$ and residue-residue interactions are negligible after $\sim 10\text{Å}$.
- The values for σ are chosen so that the RBF encoding of the edge length is approximately linear around $\sim 5\text{Å}$.
- The values for L are chosen to approximately match the average length of the shortest paths in the protein graphs at different cutoffs.
- In addition to what described in section 2.3, we also tested an architecture with BatchNorm layers between the Dropout and ReLU operations, but apart from a significant slowdown we did not notice any improvement.

Table 4: Hyper parameter space and best values

Hyper parameter	**Values**	**Best**
MP Layers L	3, 4, 5, 6, 7, 8, 9	6
MP input size $\boldsymbol{e}$	32, 64, 128	128
MP input size $\boldsymbol{v}$	64, 128, 256, 512	512
MP input size $\boldsymbol{u}$	64, 128, 256, 512	512
MP output size $\boldsymbol{e}$	8, 12, 16, 32	16
MP output size $\boldsymbol{v}$	8, 16, 32, 64	64
MP output size $\boldsymbol{u}$	8, 12, 16, 32	32
Cutoff d_{max}	6, 8, 10, 12	8
Sigma σ	10, 15, 20	15
Dropout rate	0, 0.1 0.2, 0.3, 0.4	0.2
Learning rate	$10^{-2}, 10^{-3}$	10^{-3}
Weight decay	$10^{-4}, 10^{-5}$	10^{-5}
Local weight λ^{ℓ}	1, 5, 10	1
Global weight λ^{g}	1, 5, 10	1

E Additional Metrics

The Quality Assessment literature is rich of metrics to measure the performances of a scoring method. In the main text we tried to keep the exposition uncluttered by only reporting figures for the most important metrics. Here we present a more extensive set of metrics, that further describe our method and can serve as future benchmark.

In the following, we use:

$t = 1, \ldots, T$	Target proteins
$d = 1, \ldots, D^t$	Decoys of a target
$i, j = 1, \ldots, \lvert S^t \rvert$	Residue indexes of a target
$q^{g,t,d} = \text{GDT\_TS}(C^{t,d}, C^{t,\text{native}})$	Global quality score (true)
$q_i^{\ell,t,d} = \text{LDDT}(C^{t,d}, C^{t,\text{native}})$	Local quality scores (true)
$\textsc{GraphQA}^g(\mathcal{P}^{t,d})$	Global quality score (predicted)
$\textsc{GraphQA}_i^\ell(\mathcal{P}^{t,d})$	Local quality scores (predicted)

Root Mean Squared Error (RMSE) We compute RMSE between all true and predicted scores, for both LDDT and GDT_TS.

For LDDT, it it the square root of:

$$\text{MSE} = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{D^t} \sum_{d=1}^{D^t} \frac{1}{|S^t|} \sum_{i=1}^{|S^t|} \left(q_i^{\ell,t,d} - \textsc{GraphQA}_i^\ell(\mathcal{P}^{t,d}) \right)^2$$

For GDT_TS, it it the square root of:

$$\text{MSE} = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{D^t} \sum_{d=1}^{D^t} \left(q^{g,t,d} - \textsc{GraphQA}^g(\mathcal{P}^{t,d}) \right)^2$$

Correlation coefficients We compute the Pearson (R), Spearman (ρ) and Kendall (τ) correlation coefficients between *all* true and predicted scores. Since all scores are treated equally, with no distinction between different decoys or different targets, a high value for these scores can be misleading. Thus, their *per-model* and *per-target* versions should be also checked.

For LDDT:

$$\begin{aligned} R &= \textsc{Pearson}\left(\{\, q_i^{\ell,t,d} \,\}, \{\, \textsc{GraphQA}_i^\ell(\mathcal{P}^{t,d}) \,\} \right) \\ \rho &= \textsc{Spearman}\left(\{\, q_i^{\ell,t,d} \,\}, \{\, \textsc{GraphQA}_i^\ell(\mathcal{P}^{t,d}) \,\} \right) \\ \tau &= \textsc{Kendall}\left(\{\, q_i^{\ell,t,d} \,\}, \{\, \textsc{GraphQA}_i^\ell(\mathcal{P}^{t,d}) \,\} \right) \end{aligned}$$

For GDT_TS:

$$\begin{aligned} R &= \textsc{Pearson}\left(\{\, q^{g,t,d} \,\}, \{\, \textsc{GraphQA}^g(\mathcal{P}^{t,d}) \,\} \right) \\ \rho &= \textsc{Spearman}\left(\{\, q^{g,t,d} \,\}, \{\, \textsc{GraphQA}^g(\mathcal{P}^{t,d}) \,\} \right) \\ \tau &= \textsc{Kendall}\left(\{\, q^{g,t,d} \,\}, \{\, \textsc{GraphQA}^g(\mathcal{P}^{t,d}) \,\} \right) \end{aligned}$$

Correlation coefficients per-model For every decoy of every target, we compute the Pearson (R_{model}), Spearman (ρ_{model}) and Kendall (τ_{model}) correlation coefficients between true and predicted residue-level scores (LDDT). We then report the average correlation coefficients across all decoys of all targets. The *per-model* correlation coefficients estimate the performance of the network to rank individual residues by their quality and distinguish correctly vs. incorrectly folded segments.

Per-model correlation coefficients are computed only for LDDT:

$$R_{\text{model}} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{D^t}\sum_{d=1}^{D^t}\textsc{Pearson}\left(\left\{\, q_i^{\ell,t,d}\,\right\},\left\{\,\textsc{GraphQA}_i^{\ell}(\mathcal{P}^{t,d})\,\right\}\right)$$

$$\rho_{\text{model}} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{D^t}\sum_{d=1}^{D^t}\textsc{Spearman}\left(\left\{\, q_i^{\ell,t,d}\,\right\},\left\{\,\textsc{GraphQA}_i^{\ell}(\mathcal{P}^{t,d})\,\right\}\right)$$

$$\tau_{\text{model}} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{D^t}\sum_{d=1}^{D^t}\textsc{Kendall}\left(\left\{\, q_i^{\ell,t,d}\,\right\},\left\{\,\textsc{GraphQA}_i^{\ell}(\mathcal{P}^{t,d})\,\right\}\right)$$

Correlation coefficients per-target For every target, we compute the Pearson (R_{target}), Spearman (ρ_{target}) and Kendall (τ_{target}) correlation coefficients between true and predicted decoy-level scores (GDT_TS). We then report the average correlation coefficients across all targets. With reference to the funnel plots, this would be the correlation between the markers in every plot, averaged across all plots. The *per-target* correlation coefficients estimate the performance of the network to rank the decoys of a target by their quality and select the ones with highest global quality.

Per-target correlation coefficients are computed only for GDT_TS:

$$R_{\text{target}} = \frac{1}{T}\sum_{t=1}^{T}\textsc{Pearson}\left(\left\{\, q^{g,t,d}\,\right\},\left\{\,\textsc{GraphQA}^{g}(\mathcal{P}^{t,d})\,\right\}\right)$$

$$\rho_{\text{target}} = \frac{1}{T}\sum_{t=1}^{T}\textsc{Spearman}\left(\left\{\, q^{g,t,d}\,\right\},\left\{\,\textsc{GraphQA}^{g}(\mathcal{P}^{t,d})\,\right\}\right)$$

$$\tau_{\text{target}} = \frac{1}{T}\sum_{t=1}^{T}\textsc{Kendall}\left(\left\{\, q^{g,t,d}\,\right\},\left\{\,\textsc{GraphQA}^{g}(\mathcal{P}^{t,d})\,\right\}\right)$$

First Rank Loss (FRL) For every target, we compute the difference in GDT_TS between the best decoy according to ground-truth scores and best decoy according to the predicted scores. We then report the average FRL across all targets. This represents the loss in (true) quality we would suffer if we were to choose a decoy according to our rankings and can is represented in the funnel plots by the gap between the two vertical lines indicating the true best (green) and predicted best (red).

FRL measures the ability to select a single best decoy for a given target. In our experiments, however, we noticed that FRL is extremely subject to noise, as it only considers top-1 decoys. Therefore, we consider NDCG to be a superior metric for this purpose, though we have not seen it used in the QA literature.

FRL is only computed for GDT_TS:

$$\text{FRL} = \frac{1}{T}\sum_{t=1}^{T}\left|\max\left\{\, q^{g,t,d}\,\right\} - q^{g,t,d*}\right|,$$

where $d* = \arg\max_d \left\{\,\textsc{GraphQA}^{g}(\mathcal{P}^{t,d})\,\right\}$ for every target t.

Recall at k (Rec@k) We can describe Quality Assessment as an information retrieval task, where every target represents a query and its decoys are the documents available for retrieval. If we consider the best decoy to have a score of 1 and all others to have zero score, we can compute the average Rec@k as the percentage of queries for which the best decoy is retrieved among the top-k results.

This metric, however, is subject to the same pitfalls of FRL, since it only considers the best decoy of every target and ignores the relevance of the others. As described below, NDCG offers a better perspective over the decoys retieved by a QA method.

Normalized Discounted Cumulative Gain at k (NDCG@k) For a given query we consider the top-k decoys ranked according to their predicted global scores. Discounted Cumulative Gain at k

(DCG@k) is computed as the cumulative sum of their ground-truth GDT_TS scores (gain), discounted however according to the position in the list. A high DCG@k is obtained therefore by a) selecting the k-best decoys to be part of the top-k predictions, and b) sorting them in order of decreasing quality (the higher in the list, the lower the discount).

Dividing DCG@k by $\text{DCG}^{\text{ideal}}$@$k$ (obtained by ranking according to the ground-truth scores), yields the Normalized Discounted Cumulative Gain NDCG@$k \in [0, 1]$, which can be compared and averaged across targets.

F Additional Results

The CASP 11 and 12 datasets are conventionally divided into: *stage 1*, containing 20 randomly-selected decoys per target, and *stage 2*, containing the top-150 decoys of each target. In the QA literature we found papers that only report results on either the dataset as a whole, or on stage 1 and stage 2. Furthermore, some papers report metrics on other methods whose values differ from the original papers for some unspecified reason. In the main text, we adhere to the following rules to summarize the metrics we collected:

- Metrics computed on stage 1 are considered noisy and ignored, since stage 1 splits contain only 20 randomly-selected decoys per target
- Metrics computed on stage 2 and on the whole dataset are considered equally valid, allowing to ”merge” results from papers with different scoring strategies
- If multiple values are reported from multiple sources for the same (method, dataset) pair, only the best one is reported

F.1 CASP 11

			GDT_TS								LDDT		
TestSet	Method	Source	FRL	R	R_{target}	RMSE	ρ	ρ_{target}	τ	τ_{target}	R	R_{model}	RMSE
CASP 11	ProQ3D	ProQ4									0.84	0.61	0.125
	ProQ4	ProQ4									0.77	0.56	0.147
	GRAPHQA$_{\text{RAW}}$	Ours	0.082	0.836	0.609	0.146	0.837	0.49	0.637	0.354	0.799	0.516	0.14
	GRAPHQA	Ours	0.071	0.91	0.74	0.117	0.918	0.622	0.747	0.461	0.852	0.602	0.122
stage 1	ProQ3D	3D CNN	0.046		0.755			0.673		0.529			
		Ornate	0.066	0.795	0.691		0.782	0.606	0.58	0.462			
	VoroMQA	3D CNN	0.087		0.637			0.521		0.394			
		Ornate	0.085	0.689	0.617		0.682	0.482	0.483	0.361			
	RWplus	3D CNN	0.122		0.512			0.402		0.303			
		Ornate	0.128	0.08	0.467		0.003	0.371	-0.016	0.274			
	3D CNN	3D CNN	0.064		0.535			0.425		0.325			
		Ornate	0.104	0.532	0.442		0.614	0.369	0.437	0.28			
	Ornate	Ornate	0.077	0.635	0.465		0.634	0.372	0.44	0.275			
	GRAPHQA$_{\text{RAW}}$	Ours	0.09	0.829	0.63	0.135	0.81	0.514	0.609	0.393	0.79	0.475	0.131
	GRAPHQA	Ours	0.035	0.923	0.788	0.09	0.924	0.647	0.755	0.515	0.861	0.57	0.108
stage 2	ProQ3D	3D CNN	0.066		0.452			0.433		0.307			
		Ornate	0.053	0.772	0.444		0.796	0.432	0.594	0.304			
	VoroMQA	3D CNN	0.063		0.457			0.499		0.321			
		Ornate	0.066	0.651	0.419		0.688	0.412	0.505	0.291			
	RWplus	3D CNN	0.089		0.206			0.248		0.176			
		Ornate	0.088	0.056	0.167		0.033	0.192	0.011	0.137			
	3D CNN	3D CNN	0.064		0.421			0.409		0.288			
		Ornate	0.074	0.629	0.375		0.655	0.363	0.433	0.254			
	Ornate	Ornate	0.055	0.637	0.386		0.673	0.371	0.475	0.259			
	GRAPHQA$_{\text{RAW}}$	Ours	0.071	0.82	0.379	0.149	0.82	0.357	0.618	0.251	0.787	0.529	0.142
	GRAPHQA	Ours	0.063	0.899	0.539	0.123	0.905	0.507	0.729	0.363	0.839	0.61	0.126

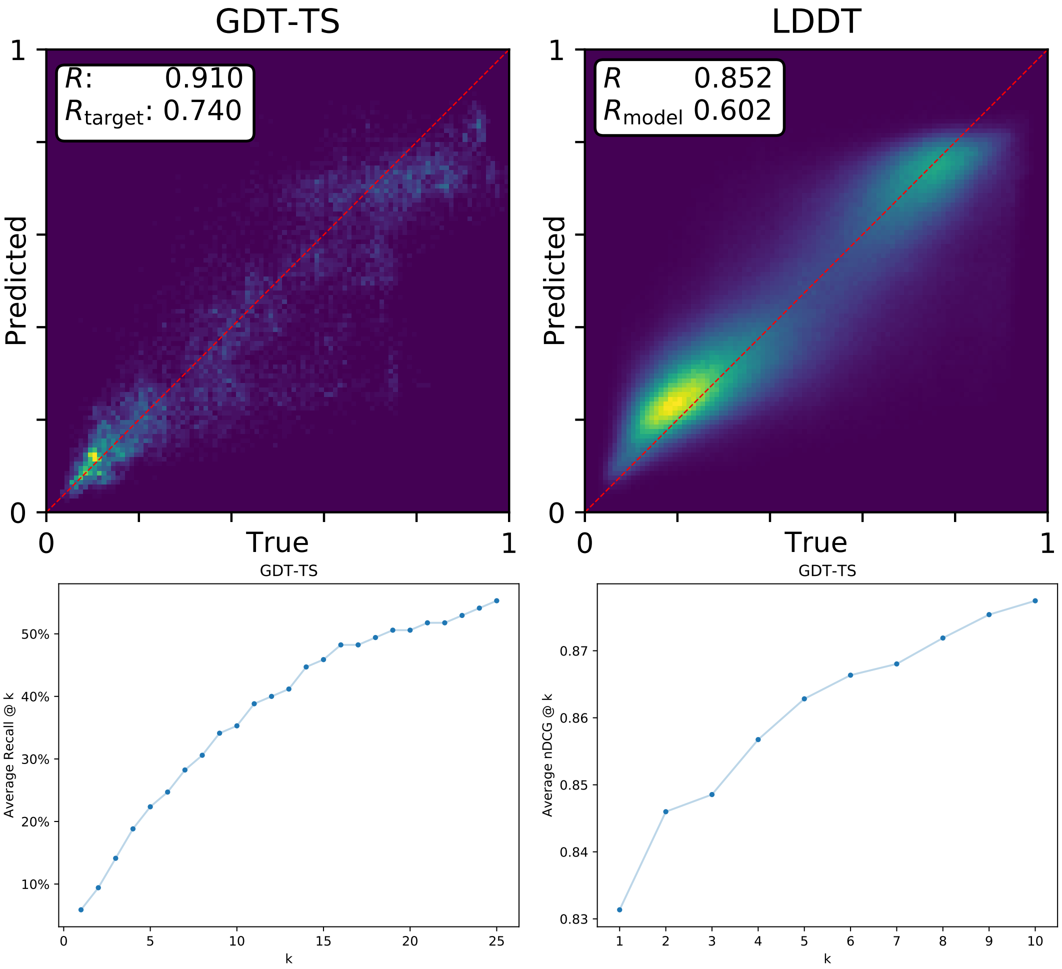


Figure 9: CASP 11: Histograms of true vs. predicted LDDT and GDT_TS scores, average recall @ k, average NDCG @ k

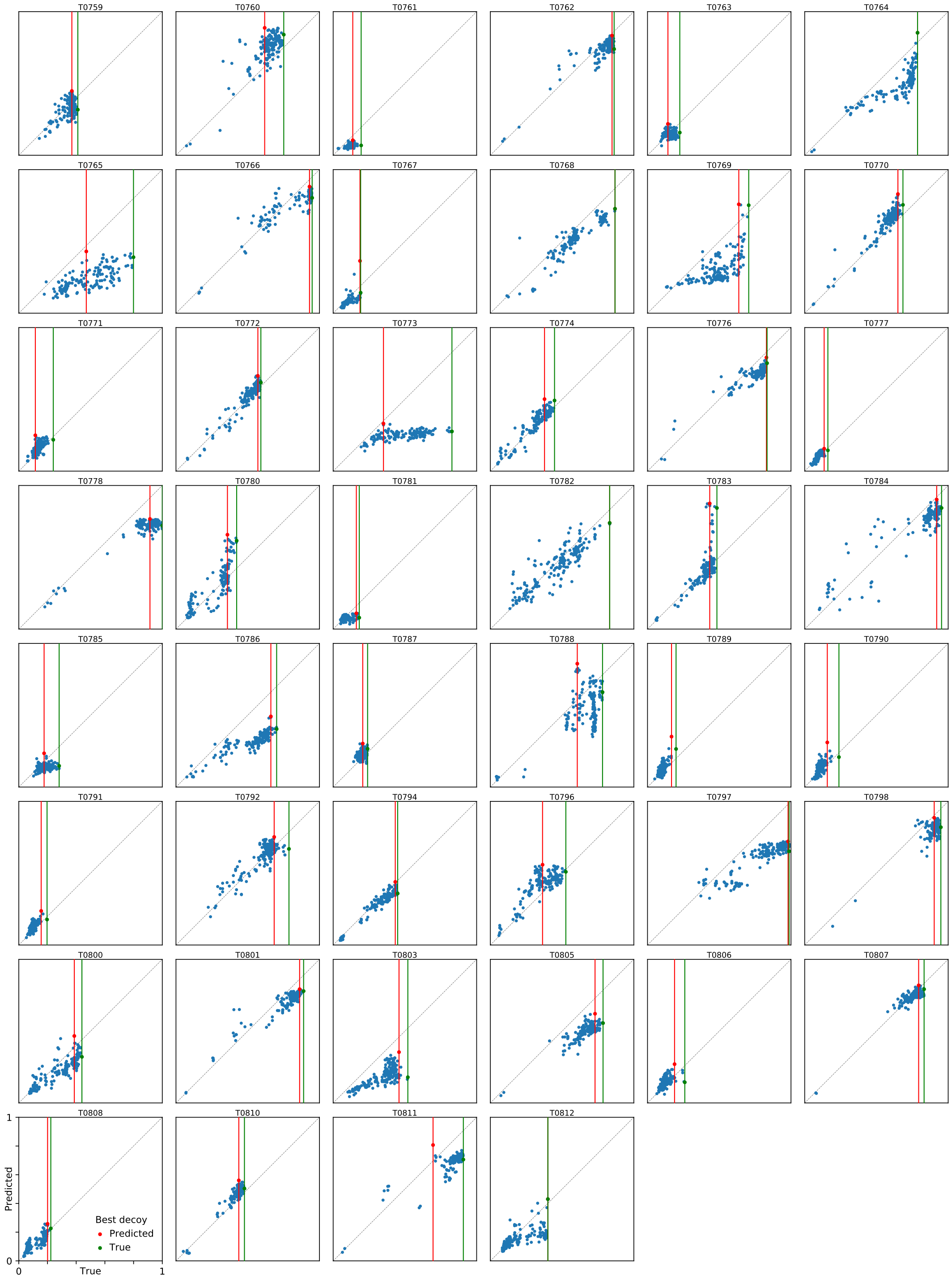


Figure 10: CASP 11: funnels (continues)

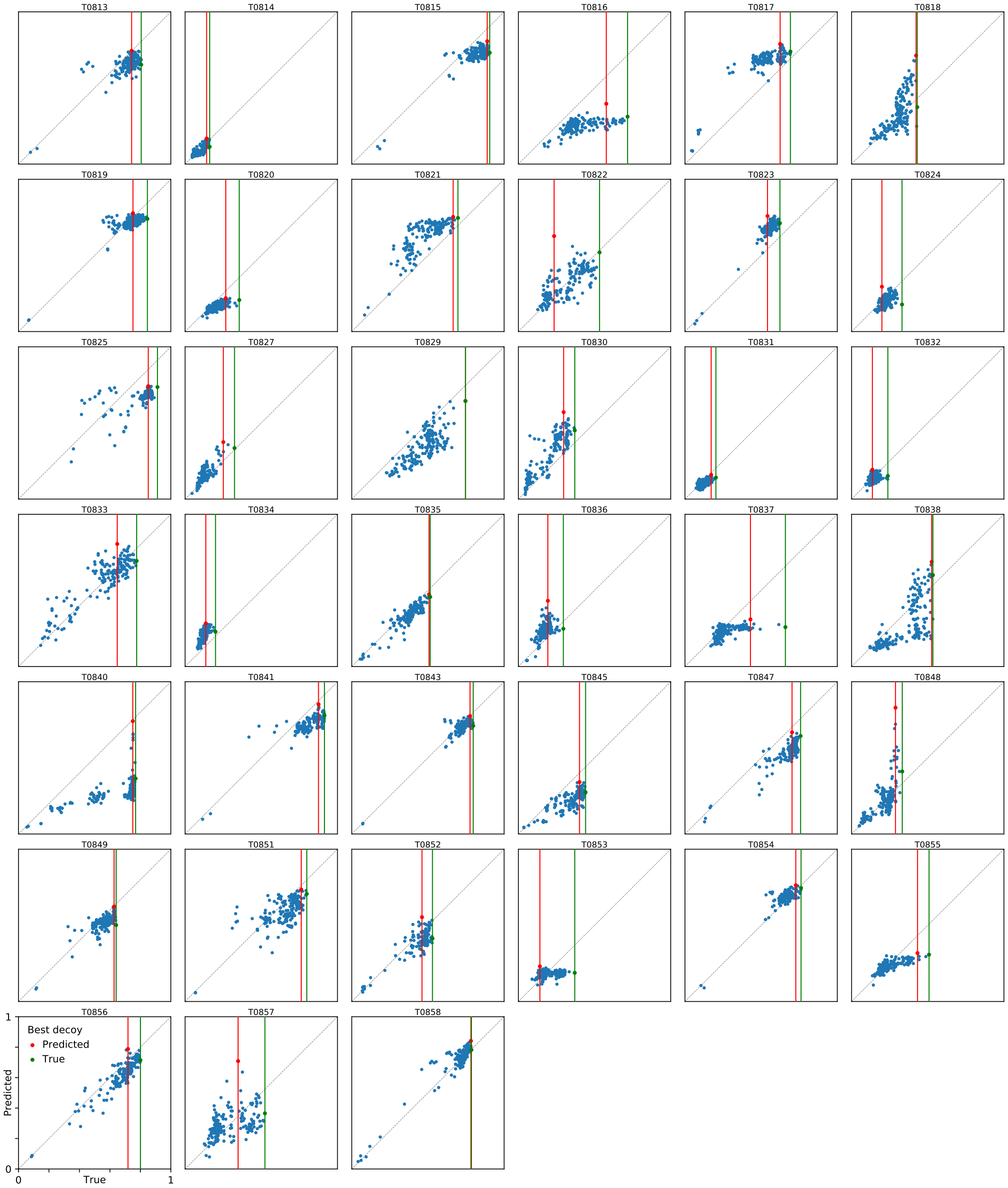


Figure 11: CASP 11: funnels (continued)

F.2 CASP 12

			GDT_TS								LDDT				
TestSet	Method	Source	FRL	R	R_{target}	RMSE	ρ	ρ_{target}	τ	τ_{target}	R	R_{model}	RMSE	ρ	ρ_{model}
CASP 12	ProQ3D	3D CNN	0.164		0.609			0.602		0.451					
	ProQ4	Ours									0.772	0.516		0.776	0.498
	VoroMQA	3D CNN	0.161		0.557			0.515		0.38					
	RWplus	3D CNN	0.192		0.313			0.355		0.257					
	3D CNN	3D CNN	0.146		0.607			0.521		0.381					
	AngularQA	AngularQA	0.138	0.651	0.439										
	GRAPHQA$_{\text{RAW}}$	Ours	0.092	0.816	0.673	0.149	0.814	0.606	0.624	0.448	0.796	0.501	0.139		
	GRAPHQA	Ours	0.089	0.843	0.745	0.137	0.834	0.66	0.684	0.503	0.843	0.565	0.124		
stage 1	ProQ3D	Ornate	0.086	0.671	0.705		0.478	0.636	0.335	0.482					
	VoroMQA	Ornate	0.085	0.456	0.611		0.381	0.554	0.263	0.414					
	RWplus	Ornate	0.132	-0.272	0.479		-0.538	0.465	-0.381	0.344					
	Ornate	Ornate	0.113	0.551	0.566		0.484	0.504	0.339	0.374					
	AngularQA	AngularQA	0.148		0.502										
	GRAPHQA$_{\text{RAW}}$	Ours	0.068	0.721	0.679	0.127	0.623	0.596	0.451	0.448	0.661	0.413	0.118		
	GRAPHQA	Ours	0.043	0.814	0.789	0.085	0.755	0.684	0.589	0.541	0.718	0.474	0.105		
stage 2	ProQ3D	Ornate	0.06	0.806	0.6		0.8	0.54	0.601	0.388					
	VoroMQA	Ornate	0.106	0.605	0.559		0.604	0.501	0.445	0.362					
	RWplus	Ornate	0.103	-0.096	0.417		-0.096	0.378	-0.067	0.265					
	Ornate	Ornate	0.072	0.67	0.491		0.657	0.458	0.472	0.322					
	AngularQA	AngularQA	0.128		0.377										
	GRAPHQA$_{\text{RAW}}$	Ours	0.094	0.807	0.614	0.151	0.807	0.545	0.618	0.395	0.793	0.507	0.141		
	GRAPHQA	Ours	0.08	0.832	0.707	0.141	0.828	0.61	0.679	0.456	0.842	0.573	0.125		

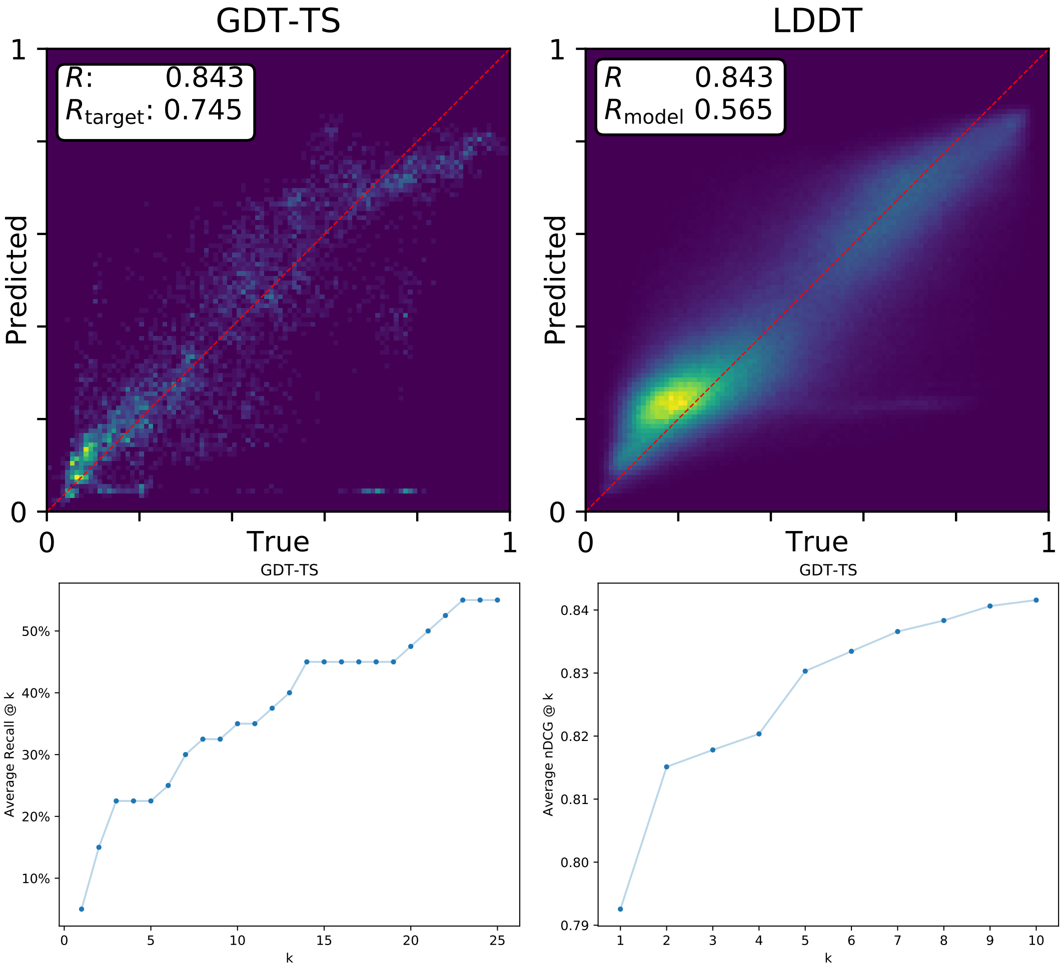


Figure 12: CASP 12: Histograms of true vs. predicted LDDT and GDT_TS scores, average recall @ k, average NDCG @ k

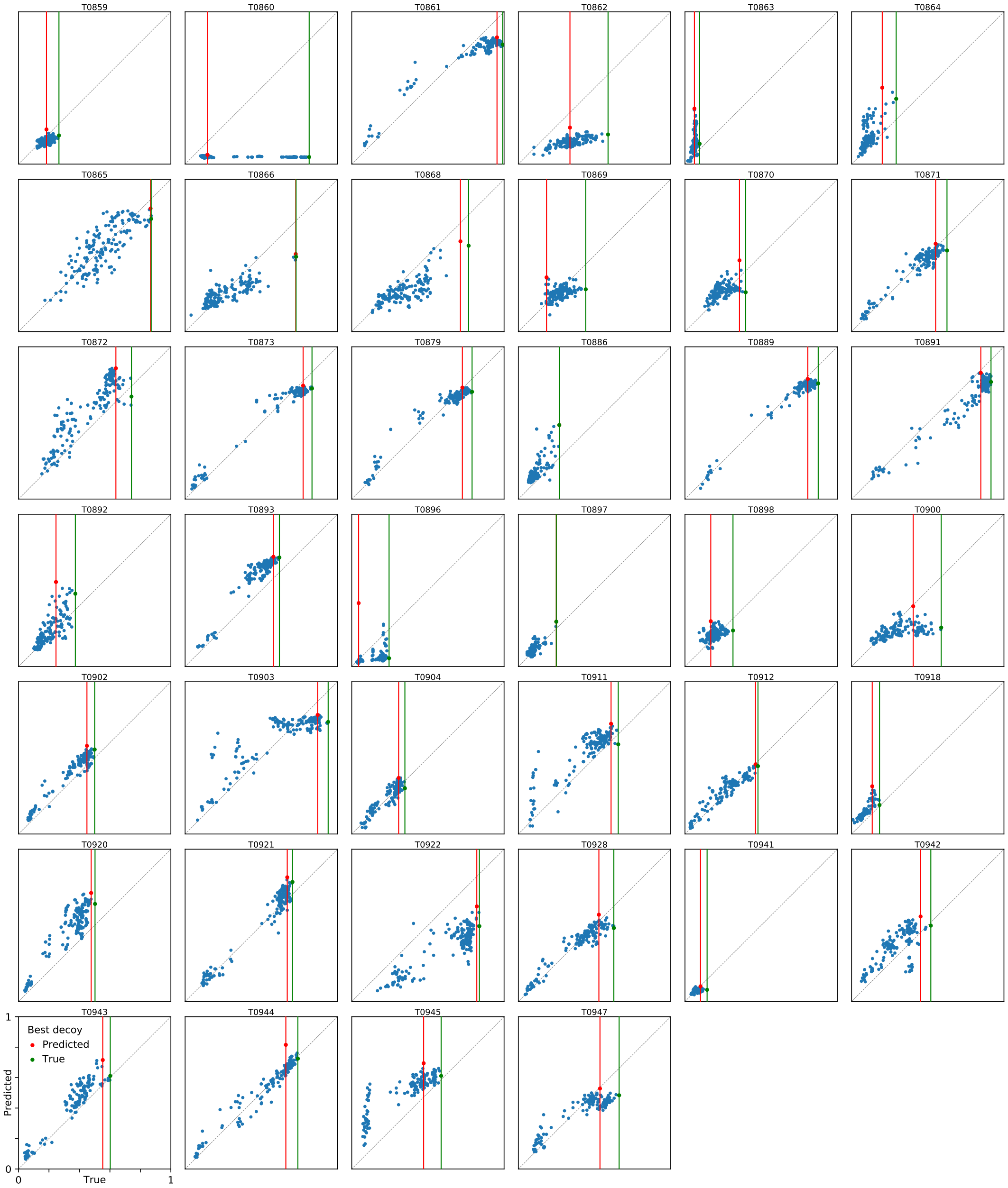


Figure 13: CASP 12: funnels