Geometric Graph Representation Learning on Protein Structures

Anonymous Author(s)

ABSTRACT

Representation learning of protein 3D structures is challenging and essential for applications, e.g., computational protein design or protein engineering. Recently, geometric deep learning has achieved great success on non-Euclidean domains. Although protein can be represented as a graph naturally, it remains under-explored mainly due to the significant challenges in modeling the complex representations and capturing the inherent correlation in the 3D structure modeling. Several challenges include: 1) It is challenging to extract and preserve multi-level rotation and translation invariant information during learning. 2) Difficulty in developing appropriate tools to effectively leverage the input spatial representations to capture complex geometry across the spatial dimension. 3) Difficulty incorporating various geometric features and preserving the inherent geometry relations. In this work, we introduce *geometric bottleneck perceptron*, and a general SO(3)-equivariant message passing neural network built on top of it for protein geometric representation learning. The proposed *geometric bottleneck perceptron* can be incorporated into diverse network architecture backbones to process geometric data in different domains. This research shed new light on geometric deep learning in 3D structure studies. Empirically, we demonstrate the strength of our proposed approach on three core downstream tasks, where our model achieves significant improvements and outperform existing benchmarks. The implementation is available at https://github.com/knowledge-anonymous/rev

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

geometric representation learning, equivariant networks, neural networks

ACM Reference Format:
Anonymous Author(s). 2018. Geometric Graph Representation Learning on Protein Structures. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Proteins, as the building blocks for all livings organisms, play a critical role in fundamental biological processes and attract great interest from different domains. Investigating the geometric structure of these macromolecules is key to understanding protein reaction mechanisms in the biological process and enhancing drug design. The recent technical advance in deep learning, especially the successful application of GNNs to model graph structures [28], makes the learning from the protein structures have seen sharply growing popularity over the last few years. Several promising results achieved by convolutional neural networks and graph neural network-based methods to understand protein structure, including Computational Protein Design (CPD) [9, 11], Ligand Binding Affinity (LBA) [17, 25], and Protein Structure Ranking (PSR) [25].

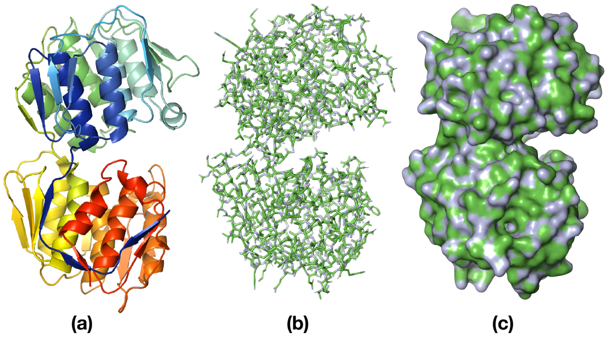


Figure 1: Different views of the 3D structure of a protein. a). cartoon view b). line view and c). surface view of a protein's structure.

The recent work demonstrated the potential ability of GNN to learn from the protein structure. Despite recent progress, there are still several challenges that remain under-explored. **First of all, effectively leveraging the spatial input information to capture complex geometry across the spatial dimension dynamically remains an open problem.** Although it is natural to model 3D protein structure representation as a graph, directly adopting the existing GNNs to handle protein 3D structures may not be sufficient enough to capture the ubiquitous multi-level structural information during learning. Therefore, it is not tuned to capture the interaction between amino acids that are spatially close but further in sequential location. Currently, the de facto choice for the geometric deep learning for 3D structures is graph neural networks [6, 11]. Message passing neural network learns the distant nodes' information by aggregating the messages from direct neighbors and stacking GNN layers. Many previous works have already identified some existing problems with the message passing paradigm, including the problem of over-smoothing when the GNN is deep with many layers and over-squashing when the message passing is relying on long-range interactions [1]. How to flow the information among the graph network efficiently without information distortion is crucial for geometric deep learning. There remains a need for an improved propagation method for GNNs to handle complex 3D geometric data.

Woodstock '18, June 03–05, 2018, Woodstock, NY
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

Secondly, difficulty in discovering and preserving different leveled geometric features between node/edge and graph spectra. The geometric representations are composed of mutually related edge and node features. The representation learning of the protein structure should not only rely on the node, but also on the edge features. Moreover, the scalar and vector features for nodes and edges are mutually related, so a module with the ability to learn scalar and vector features jointly is essential for the model to capture the geometric representation from the natural protein graph. For instance, the overall protein backbone is represented by a set of Cα-CO − NH − Cα coplanar units. Multiple conformations can be generated with the rotation of one coplanar unit because the chemical bonds around the coplanar unit will rotate accordingly. Thus learning the protein structure representations not only requires the network to process the geometric features simultaneously but also preserve both nodes/edges and graph level information jointly.

Lastly, it remains challenging to capture the non-local relations and abstract feature maps in large complex 3D protein structures. Existing works [9, 11] typically use GNN-based methods. As shown in Figure 1, one protein contains thousands of amino acids. The folding and intramolecular binding of the protein amino acid sequences forms the protein 3D geometric structures. The amino acids pairs that are sequentially far apart might be spatially in contact. Thus the problem radius, which is the required range of interaction between nodes in the graph to be solved, is comparably large for the protein geometric graph. Figure 1 (b) shows the 3D structure of one protein, where the line shows where bonds connect the atoms. Since the protein as a specific type of graph is very long in sequence and complex, the messages from non-adjacent nodes might need to propagate across the whole network. The large protein graphs have limited the approaches to model the protein 3D structures. The requirement to learn long-range dependencies and complex structural properties largely hardens our task.

We propose a novel graph neural network for geometric graph representation learning to solve the challenge of capturing complex geometry across the spatial dimension and integrating vector and scalar features. Specifically, a novel *Geometric Bottleneck Perceptron* (GBP) is proposed for integrating the scalar and vector features and strengthening the shared low-level representations with a reduced parameter space in the module. GBP is a general drop-in structure that applies to various domains where geometric information is present. Furthermore, we introduce a GBP-based Equivariant Message Passing Neural Network (GBP-GNN) for protein 3D structural representation learning. This model can aggregate the complex spatial information in feature space to capture the geometric pattern and increase the model's scalability. We summarize our main contribution as follows:

- **A new SO(3)-equivariant message passing neural network is proposed.** We propose a new versatile framework for protein geometric representation learning. Our SO(3)-equivariant message passing network supports a variety of geometric representation learning tasks.
- **A novel drop-in module for geometric representation learning is proposed.** We propose a novel *Geometric Bottleneck Perceptron* (GBP) to integrate the geometric features and capture the complex geometric relations in the 3D structure. The outputs of the GBP module are equivariant to graph rotation and translation. Most importantly, this design allows the model to scale up to stack more GNN layers, allowing the graph to learn representations from larger receptive fields.
- **Comprehensive experiments were conducted.** Comprehensive experiments on three datasets with three protein representation learning tasks validate that GBP-GNN is capable of learning geometric relations in protein structure for various downstream tasks and outperforms the state-of-the-art methods.

2 RELATED WORK

2.1 Geometric Deep Learning

The interest in processing graph data with Graph Neural Networks (GNNs) is increasing over the years [7, 14, 28]. Modeling the geometric structures as a graph without losing geometric properties attracted a significant amount of attention [6, 7]. In recent years, graph structures show impressive results on complex protein structure tasks such as rigid protein docking [6], protein interface prediction [25], mutation stability prediction [25], protein structure ranking [25]. The representative efficiency of graphs also allows researchers to investigate structures at a fine-grained level. While representing proteins with residue level resolution is effective, it is also possible to represent them as atoms. Using atoms to represent proteins have advantages including, generalized structure and ease of transferring information from other problems [10]. Therefore, studying the proteins with atom level graphs has been explored as ways to improve the representation power of the graph potentially [4, 8, 10]. [4] presented a hierarchical network that initially represents proteins with all atoms then aggregates the information to $C\alpha$ atoms. Similarly, [8] proposed to represent proteins as multi-level atom graphs.

2.2 Equivariant Graph Neural Networks

The properties of graph representations can change with graph translation, rotation, and permutation. The GNN is permutation equivalent networks by their design [14]. Although it is possible to introduce various transformations to the network by data augmentation, it is computationally inefficient. Recent works [5, 6, 23] tackle this problem with equivariant network architectures. [5] proposed high performance SE(3)-equivariant Transformer networks. More recently, [6] SE(3)-equivariant graph matching networks with competitive results and inspiring speedups on rigid protein docking problems.

2.3 Representation Learning on Protein Structure

Several core tasks rely on the structural information of the proteins and explore the representation of the protein structures. The primary purpose of Computational Protein Design (CPD) [9, 11] is to design a protein with desired structure and functions. [9] proposed Structured Transformer, a conditional generative model for generating sequences based on the structure and sequence priors. [11] explore the GNN approach to tackle this problem.

Another line of protein representation learning work is Ligand Binding Affinity (LBA) [17, 25], which aims to predict drug-target binding for potential drug discovery and drug screening. [21] proposed a convolutional neural network architecture that relies on sequential information to predict the affinity. The proceeding works [17, 19] show that learning from structure rather than the sequence is more effective. In [19] the author proposed a hybrid model that processes the protein sequence and ligand structure separately to predict the protein-ligand binding affinity. A more effective framework [17] takes advantage of the 3D geometric information of the drug-target interaction complex and shows superior performance.

The PSR task, also known as Model Quality Assessment, evaluates the generated protein structure. Over the years, many frameworks are adapted and proposed [3, 10, 12, 15, 20, 22, 26, 29] to tackle this problem. [12] introduced geometric structural scoring function for protein 3D structure. [22] propose a performant 3D Convolutional Networks (CNNs) with the local and global scoring scheme. [22] used residue-wise local orientations to learn the local structures for protein structure ranking.

3 PRELIMINARIES

The protein is a sequence of amino acids where each amino acid contains four backbone atoms and a set of side-chain atoms. The sequence of amino acids folds into a unique structure to form a protein tertiary structure, which defines its properties and functions. It is crucial to represent this structural information without handcraft features to avoid bias. Therefore, we map these geometric structures to graph representations invariant to rotation and translation transformations. In this section, we first introduce the notation and generation process for the SE(3)-equivariant protein representations, then formulate the three tasks used to evaluate the proposed representation learning model.

We formulate the protein structure into graph representation, where each input protein as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{E}, \mathbf{F})$. $\mathcal{V}$ is the set of N nodes in the graph representing amino acid residues, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{K}$ is the set of edges that connects the nodes. The $\mathcal{K}$ denotes the maximum number of edges per node used to generate the k-Nearest Neighbor (kNN) graph. The kNN graph is constructed based on the 3D coordinates of the set of points $K = \{(x_i, y_i, z_i) | x_i, y_i, z_i \in \mathbb{R}\}$ that corresponds to the coordinates of Cα atoms.

The node features $\mathbf{E}$ include scalars features and vector features (s_n, V_n). The scalar features $V_n \in \mathbb{R}^O$ include identity embedding and dihedral angles. The vector features of the nodes $V_n \in \mathbb{R}^{3\times P}$ includes orientation of a node with previous and next nodes in sequential orders.

Similarly, we denote the edge features $\mathbf{F}$ as scalar and vector features $(s_e, V_e) \in \mathbf{E}$. The scalar features, $s_e \in \mathbb{R}^R$, are the euclidean distance of two nodes. The vector features, $V_e \in \mathbb{R}^{3\times D}$, include information associated with 3D coordinates such as direction unit vectors. The direction unit vectors are produced by calculating the difference between two nodes then normalized to get a unit vector.

Geometric Graph Representation Learning for protein aims to design a model that could enhance the quality of spatial encodings and capture the hierarchical and geometrical patterns in the protein complex structures for downstream tasks. With this aim in mind, we introduce the three tasks first:

Definition 3.1. **Computational Protein Design (CPD).** Given a protein with 3D structure and sequential information until given position, our goal is to model the next token in the protein sequence. This operation is repeated iteratively from the first position to the last position. Hence, the model only predicts the full protein sequence from structural information as a multi-class classification task. We formulate this problem as $\mathcal{F}_{CPD} : p(S_i | S_{j:j<i}, \mathcal{G})$.

Definition 3.2. **Protein Structure Ranking (PSR).** Given a predicted/ generated 3D protein structure, our goal is predicting its' quality. The quality of a protein structure is defined as the probability of a given structure being a real protein structure. When the crystallized protein structure is not available, this task is crucial to estimate the accuracy of the predicted structure. Therefore, it is a regression task to estimate a single value from a protein 3D structure. We formulate this problem as $\mathcal{F}_{PSR} : p(r|\mathcal{G})$, where r is the quality score of the 3D structure by the model.

Definition 3.3. **Ligand Binding Affinity (LBA).** Given the structure of a protein and a ligand such as a drug or inhibitor, we predict the binding affinity of the protein-ligand complex. Predicting the binding strength of the complex is a classical graph regression task. This task is essential for drug discovery and drug screening purposes. We formulate this problem as $\mathcal{F}_{LBA} : p(a|\mathcal{G})$, where a denotes the binding affinity of the complex.

4 METHODOLOGY

This section presents the proposed GBP-GNN model for geometric representation learning of the protein 3D structures. We first describe the overall SO(3)-equivariant message passing framework to model the geometric properties in the graph. Then we introduce the details for the GBP module to incorporate structural scalar and vector features into the model. Finally, we show how to integrate our proposed GBP-GNN into the representation learning process on regression and classification tasks.

4.1 Model Construction

Learning the geometry properties of complex protein structures is a challenging problem. Our study's goal of representation learning is to learn the representation F of a graph embedded with the geometric structural information and satisfies several facets for analyzing and performing downstream tasks, including strong discriminative power and equivariant properties. Several challenges need to be tackled to achieve this goal, including: 1) Difficulty in leveraging the different types and levels of geometric information in nodes /edges and graph spectra. 2) Difficulty in capturing the complex geometry across the spatial dimensions. 3) Difficulty in designing a versatile framework for wide ranges of geometric representation learning tasks.

To rectify the above challenges, we propose a novel *Geometric Bottleneck Perceptron* - based Graph Neural Network (GBP-GNN) to model the protein 3D structure. As shown in Figure 2, the overall architecture is composed of three main components. We first construct the geometric graph in the embedding space by encoding the 3D structure as rotation invariant scalars s and geometric vectors V for both nodes and edges in the graph. Then we propose a novel GBP-GNN architecture as the second component in Figure 2-b. This geometric neural network can be stacked by N-th

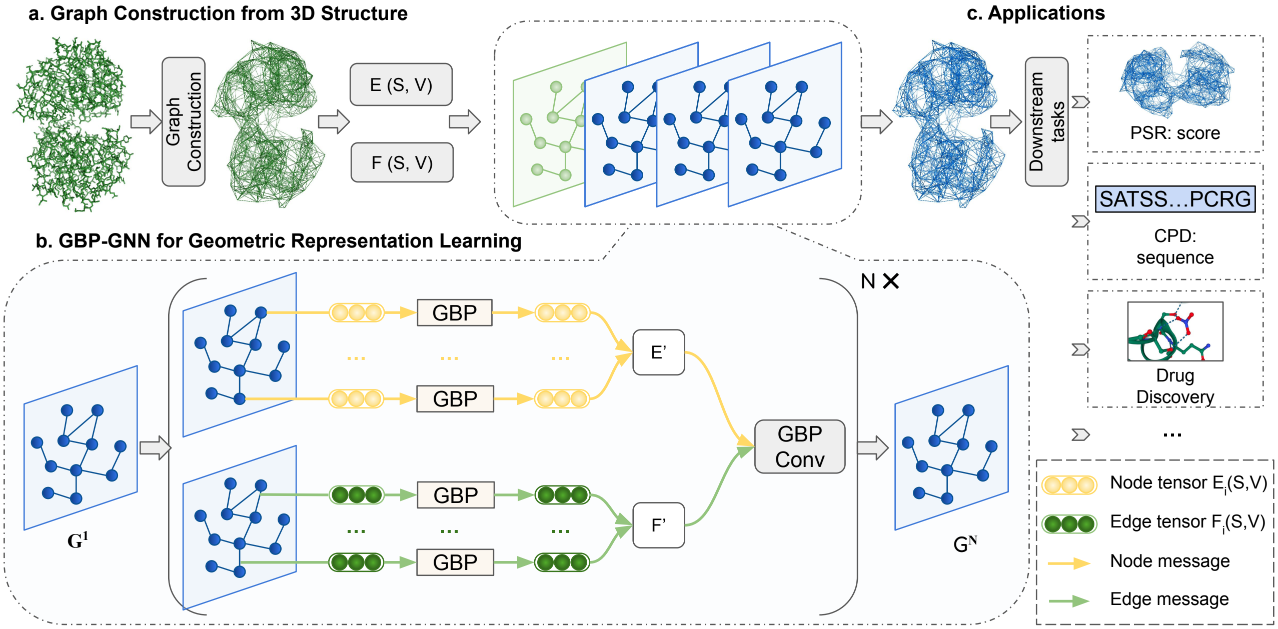


Figure 2: The architecture of the proposed GBP-GNN framework. It has three main components: (a) graph construction from a 3D protein structure, (b) a *Geometric Bottleneck Perceptron*-based GNN for 3D structural representation learning, and (c) downstream tasks for prediction.

layers and used as an encoder to learn the latent representation of the geometric features or decoder to decode the learned latent representations for downstream tasks in the network. For the last component, we present the downstream tasks' application by two categories: classification tasks and regression tasks.

4.2 Geometric Bottleneck Perceptron

In order to extract information from protein 3D structures and keep the native structure of the features intact, we model the 3D graph representations invariant to their orientations. For each node or edge in the 3D graphs, its multi-level geometry information is featured by a tuple (s, V). The scalar features s include relative distance and positional distance, and the vector features V include unit vectors of orientations, the direction of the side chains, and edge directions, as illustrated in Figure 3(a). Since the geometry information includes both scalar and vector features, it is natural to model them separately by different channels. However, each type of representation is mutually related to express the complete geometric information. Here we take advantage of the expressive power of our proposed GBP module to ensure that this property holds throughout the processing pipeline. The scalar and vector features (s, V) for both nodes and edges are processed separately by the scalar channel and vector channel and incorporated by the GBP module to return updated scalar and vector feature representations. The goal of the GBP is to learn a mapping function $\mathcal{F} : (s, V) \longrightarrow (s', V')$ with invariant constraints.

The architecture of the proposed GBP module is shown in Figure 3(b). For simplicity, we denote the GBP encoding process as $\mathrm{GBP}_\lambda(\cdot)$ with parameters λ, where λ is a downscaling hyperparameter for the GBP module. The expression of scalar and vector features is jointly generated by the two mapping function $\mathcal{F}_V : (V) \longrightarrow (z, V')$ and $\mathcal{F}_s : (s, z) \longrightarrow (s')$ in the GBP module.

4.2.1 Expression of vector representations V. The key point of the vector representation path is to aggregate vector features into a latent representation that is permutation and rotation equivariant to pass to the scalar path. For each node or edge vector feature, GBP generate the latent representation z as Eq. (1). Thus the input feature vector V with representation depth r is downscaled by λ.

$$z = \{v \cdot w_d | w_d \in \mathbb{R}^{r \times r/\lambda}\} \tag{1}$$

To generate the vector representation V', the GBP transform z into representation of depth r', which is the depth of the V'. The intermediate representation V_u is calculated by parameter w_u as Eq. (2).

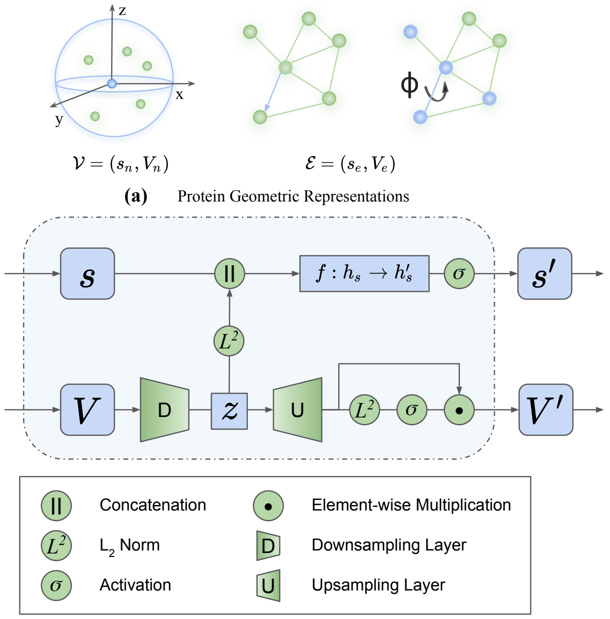
$$V_u = \{z \cdot w_u | w_u \in \mathbb{R}^{r/\lambda \times r'}\} \tag{2}$$

Then GBP updates the vector representation by gating operation defined in Eq. (3) to generate the updated vector representation V'.

$$V' = V_u \odot \sigma_v(||V_u||_2) \tag{3}$$

where σ_v is the non-linearity applied to L_2 norm of V_u and $\odot$ denotes the element-wise multiplication. When σ_v is a sigmoid function, this operation acts as a gating mechanism for the intermediate representation V_u.

4.2.2 Expression of scalar representations s. For geometric representation learning, it is essential to jointly integrate the vector and scalar features to aggregate the geometric information flow in the graph. Each node or edge scalar feature is updated by fusing



(a) Protein Geometric Representations

(b) Geometric Bottleneck Perceptron

Figure 3: The proposed *Geometric Bottleneck Perceptron* with illustration of feature representations. (a) shows the vertex and edge representations of protein graph, (b) the detailed architecture of *Geometric Bottleneck Perceptron.*

the latent vector feature z from the vector representation path to fully model the relation and geometry information in the graph. The aggregation operation of the scalar and transformed vector information in the scalar channel can be written as:

$$s_{(s,z)} = ||z||_2 \cup s \tag{4}$$

where $|| \cdot ||_2$ denotes the L_2 norm. We denote t as the representation depth of s. Then $s_{(s,z)} \in \mathbb{R}^{t+r/\lambda}$ with the depth of $(t + r/\lambda)$ is projected to s' with representation depth t' as shown in Eq. (5).

$$s' = \left\{ \sigma_s(s_{(s,z)} \cdot w_s) | w_s \in \mathbb{R}^{t+r/\lambda \times t'} \right\} \tag{5}$$

where σ_s denotes the non-linearity, and w_s is learnable parameter that performs the linear transformation.

The GBP module aggregates complex spatial and geometric properties and generates the updated tuple (s', V') with the expression of vector and scalar representations jointly. The GBP module can be stacked ω times to further enhance the feature blending and expressiveness of the geometric feature representations in the neural network.

4.3 GBP-GNN for 3D graphs

In this section, we propose a SO(3)-equivariant message passing neural network framework to work with GBP for the protein representation learning in detail. The main components of our proposed GBP-GNN are geometric graph convolution and geometric position-wise feed-forward operations. Both components process the scalar and vector channels by GBP with inter-channel communication throughout the network.

4.3.1 *Geometric Graph Convolution.*

Unlike defining the geometry of the 3D structure in terms of Cartesian coordinates of nodes, the GBP-GNN represents the geometry information by the orientation of the nodes and different types of the distance between the nodes to achieve the goal of node permutation invariant and graph rotation equivariance. Although such a choice of representation meets the equivariance constraints, a new neural network design is needed to maintain the expressive power without losing the geometric information during the graph message passing operations.

For node x_i, we denote $N(i)$ as the neighbor nodes of node x_i. The neighbors can be selected with different methods, including k-Nearest Neighbors and the nodes within a given radius. Since distance is an important factor in defining the influence between nodes, it effectively reduces the complexity. Thus a single layer of the proposed geometric graph convolution layer can be written as Eq. (6).

$$x_i^k = \phi^k \left(x_i^{k-1}, \mathcal{F}_{\forall j \in \mathcal{N}(i)} \Omega_\omega^k (x_i^{k-1}, x_j^{k-1}, e_{i,j}) \right) \tag{6}$$

where ϕ denotes the differentiable function, $\mathcal{F}$ denotes the permutation invariant aggregation function, k denotes the depth of the network and Ω_ω denotes the message generation function generated by ω-th GBP layers. The geometric graph convolution can be detailed into the following four steps.

First, for the neighbor nodes i and j with their connection information $e_{i,j}$, the message flow regarding to the target node i and the neighbor node j at the first layer can be written as:

$$\bar{\Omega}^k = \text{GBP}(e_{i,j}^{k-1} \cup x_i^{k-1} \cup x_j^{k-1}) \tag{7}$$

where $\cup$ denotes the concatenation operation. Both vector and scalar representations are concatenated then integrated by the GBP layer.

Next, for the messages generated by ω-th layer, $\bar{\Omega}^k$ is recursively generated by GBP until ω-th iteration. The output of the Ω_ω^k function is calculated with Eq. (8)

$$\Omega_\omega^k = \text{GBP}(\Omega_{\omega-1}^k) \tag{8}$$

where Ω_ω^k denotes output of recursion with ω-th iteration. The message function generates messages for each node in the graph.

Then, to get the updated node features $\hat{x}_i^k$, the generated messages are aggregated by $\mathcal{F}$ shown in Eq. (9).

$$\mathcal{F}\left(\mathcal{G}(\mathcal{V}, \mathcal{E})\right) = f\{\Omega_{\omega,v_i}^k | v_i \in \mathcal{V}\} \tag{9}$$

where f is a permutation invariant function including MEAN, SUM or MAX to aggregate the generated messages.

Lastly, the updated node features is calculated by Eq. (10) as:

$$\phi^k(\hat{x}^{k-1}) = \hat{x}^{k-1} + x^{k-1} \tag{10}$$

where $\phi^k(\cdot)$ denotes the output of geometric graph convolution block and the skip-connection is utilized among different geometric convolution blocks to allow the non-synchronicity of changes in the feature representation space.

4.3.2 *Position-Wise Feed-Forward Network.*

A position-wise feed-forward network follows each geometric graph convolution block.

For each node x a linear function with shared weights is applied to make the latent representation independent of the size of the input. At this step, we construct a position-wise representation interaction by applying linear functions with shared weights to all nodes. Then p layers of GBP are applied to get the updated vector and scalar representations as: $z_p^k = GBP(z_{p-1}^k)$.

The final output of the position-wise feed-forward network is calculated by applying $\phi(\cdot)$ to the updated feature representation z_p^k as:

$$x^k = \phi^k \left(GBP(z_{p-1}^k) \right) \tag{11}$$

4.4 Graph Classification Task

The overall framework for the classification task is based on an encoder-decoder structure composed of γ-th encoder layers and δ-th decoder layers. The encoder layers learn the latent representation of the geometric features, and the decoder layers decode the learned latent representations for classification tasks on either node /edge level or graph level.

For the CPD task, all structural information is available to the model regardless of the target node. However, the sequential information is restricted to use only for nodes before a given nodes' position. Therefore, this special case of a graph node classification task is auto-regressive based on the availability of the sequential information conditioned to the target's position [9] [11]. The model masked out the edges in the graph that connect the target node to the nodes after the target to prevent information leakage. The structural information is passed from the γ-th encoder layers to decoder layers with auto-regressive sequence information encoded to the edges. The stack δ layers of decoder updates the node features to $x^{\gamma+\delta}$ before prediction. In order to make the predictions, a fully-connected linear layer reduces the channel dimension to 20, where each dimension represents one of the 20 amino acids. Hence, the output of each node resembles the probability amino acid identity. CPD task uses cross-entropy loss function $\mathcal{L}(y, p) = -\sum_i^c y_i * log(p_i)$ to measure the model's performance for training, where c denotes the number of classes, y_i is one for the correct amino-acid class and p_i is the prediction of y_i. Since one residue can only have one identity, y contains only one true value per amino acid.

4.5 Graph Regression Task

For the graph regression task, the model aims to predict a single real-valued attribute of the input 3D graph. The overall framework for the graph regression task is composed of two components: the encoding layers and the prediction layers. The model first incorporates geometric scalar and vector features by γ number of geometric neural network layers as encoder layers. Then the updated latent scalar node representations s^γ will pass to the prediction layers as:

$$p = U\left(\frac{1}{||\mathcal{V}||} \sum\nolimits_{(s,V)\in\mathbf{E}} s^\gamma \right) \tag{12}$$

where the function $U(\cdot)$ can be implemented by the Multi-Layer Perceptrons (MLPs).

For the PSR task, the model predicts a score as the quality of the provided protein 3D structure. For the LBA task, the model predicts the binding strength based on the given protein and drug interaction complex. The mean squared error loss function $\mathcal{L}_r = (y - p)^2$ is used to evaluate the model's performance for both of the graph regression tasks, where y is the desired property value of the ground truth and p is the prediction of the model.

5 COMPLEXITY ANALYSIS

Our proposed model stacks the geometric graph convolution network with a position-wise feed-forward network. The time complexity of the geometric graph convolution network in terms of the forward passes is $\Theta(MD^2)$, where M is the number of edges, and D is the representation depth. The geometric graph convolution network is followed by the position-wise feed-forward network with $\Theta(ND^2)$ time complexity, where N is the number of nodes. Hence the overall time complexity is $\Theta(MD^2+ND^2)$. The proposed GBP block with bottleneck reduces the representation D by λ times.

6 EXPERIMENT

In this section, we evaluate our proposed GBP-GNN on three core tasks in geometric representation learning of protein 3D structures: CPD, PSR and LBA. In addition to their diversity in the required task output and real-world use cases, these three tasks span different use cases of our proposed geometric message passing framework: CPD is a classification task, and PSR and LBA are regression tasks. The experiments are conducted on 8x Nvidia A100 GPUs with 40 GB memory. The design choices and hyper-parameters are further discussed in the supplementary material.

6.1 Dataset

Classification of Protein Structures Dataset: The CATH dataset used in CPD is constructed based on the hierarchical classification of protein structure (CATH) [24]. All the chains from the test set with the same CATH topology classification are removed from train and validation splits to avoid imbalanced dataset bias. In the experiments, we used 80%, 10%, 10% splits following the previous work [9] to test our model. After the filtering, there were 18024 chains in the training set, 609 chains in the validation set, and 1120 chains in the test set.

Protein Structure Ranking Dataset: The PSR dataset is a collection of predicted 3D models submitted to CASP [16] competition. The predicted models are evaluated against experimentally observed native protein structures by the global distance test (GDT_TS). The dataset contains predictions and targets submitted in nine CASP competitions. We follow the same dataset splits as previous work [25].

Ligand-Binding Affinity Prediction Dataset: The LBA dataset is generated from the PDBBind database [18] with the corresponding binding strength of the protein and ligand interaction complex. PDBBind database is a collection of experimentally measured binding affinity data for drug and target interaction complexes. We follow the same splits used in previous work [25].

6.2 Benchmarks

The proposed methods is compared with the state-of-art methods for all three tasks.

Table 1: Comparison of the proposed GBP-GNN with the benchmarks on the CPD task.

	Perplexity ↓			Recovery ↑		
Method	Short	Single	All	Short	Single	All
STran [9]	8.54	9.03	6.85	28.30	27.60	36.40
SGNN [11]	8.31	8.88	6.55	28.40	28.10	37.30
GVP [11]	7.10	7.44	5.29	32.10	32.00	40.20
GBP-GNN	**6.14**	**6.46**	**5.03**	**33.22**	**33.22**	**42.70**

- To validate the significance of the proposed GBP for PSR task, the proposed model is compared with 3DCNN [10], ProQ3D [26], VoroMQA [20], RWplus [29], SBROD [12], Ornate [22], DimeNet [15], GraphQA [3], and GVP [10].
- For the LBA task, we compared the proposed model with Cormorant [2], 3DCNN [10], DeepAffinity [13], GIN [19], GAT [19] , GAT-GCN [19] and GVP [10].
- To validate the superiority of the proposed GBP for the classification task CPD, the proposed method is compared with STran [9], SGNN [9] and GVP [11].

6.3 Evaluation Metrics

A set of metrics are used to measure the performance of the models.

- For the CPD task, we use perplexity and recovery to evaluate the performance of the models. The perplexity is the exponential of the cross-entropy calculated from the model predictions. The recovery score is calculated as Eq. (13):

$$\text{Recovery} = \underset{S\in D}{\text{median}}\left(\frac{1}{||S||}\sum_{i\in S}||S_i - \hat{S}_i||\right) \tag{13}$$

  where D denotes the dataset, S denotes an amino acid sequence within the dataset, and $\hat{S}$ is the predicted amino acid. The absolute error is calculated for each position i, and averaged by dividing the length of the sequence $||S||$.
- For the PSR task, following the previous works [3, 12, 22] the evaluation metrics are mean and global statistical correlations. The mean correlation is the mean of all correlation scores computed for each sample. Furthermore, The global correlation is the correlation score over the whole test dataset. The statistical correlations are Pearson's correlation (p), Spearman's correlation (Sp), and Kendall's correlation (K).
- Following the previous works [13, 19, 25] the evaluation metrics for the LBA task are the root mean square error (RMSE), Pearson's correlation (P), and Spearman's correlation (Sp).

7 RESULT

7.1 Overall Performance

7.1.1 Computational Protein Design. Table 1 shows the comparisons of the GBP-GNN with the baselines on the CPD task. Our proposed method outperforms the baseline methods on both perplexity and recovery scores. Furthermore, the proposed model increases the perplexity of the Short and Single subset by over 15%. The improvement of the recovery of all structures is over 8%. On average, our model improves performance by 8%.

Table 2: Comparison of the proposed GBP-GNN with the benchmarks on the PSR task.

PSR	Local			Global		
Method	p ↑	Sp ↑	K ↑	p ↑	Sp ↑	K ↑
3DCNN [25]	0.491	0.431	0.272	0.643	0.769	0.481
ProQ3D [26]	0.444	0.432	0.304	0.796	0.772	0.594
VoroMQA [20]	0.412	0.419	0.291	0.688	0.651	0.505
RWplus [29]	0.192	0.167	0.137	0.033	0.056	0.011
SBROD [12]	0.431	0.413	0.291	0.551	0.569	0.393
Ornate [22]	0.393	0.371	0.256	0.625	0.669	0.481
DimeNet [15]	0.302	0.351	0.285	0.614	0.625	0.431
GraphQA [3]	0.357	0.379	0.251	0.821	0.820	0.618
GVP [10]	0.581	0.462	0.331	0.805	0.811	0.616
GBP-GNN	**0.612**	**0.517**	**0.372**	**0.856**	**0.853**	**0.656**

Table 3: Comparison of the proposed GBP-GNN with the baselines on the LBA task. The results are averaged over 3 independent runs.

	Method	RMSE ↓	p ↑	Sp ↑
ENN	Cormorant [2]	1.568 ± 0.012	0.389	0.408
CNN	3DCNN [25]	1.416 ± 0.021	0.550	0.553
	DeepAffinity [13]	1.893 ± 0.650	0.415	0.426
Graph	GCN [25]	1.570 ± 0.025	0.545	0.533
	DGAT [19]	1.632 ± 0.127	0.529	0.528
	DGIN [19]	1.659 ± 0.027	0.479	0.478
	DGAT-GCN [19]	1.662 ± 0.039	0.474	0.457
	GVP [10]	1.594 ± 0.073	0.434	0.432
Ours	GBP-GNN	**1.405 ± 0.009**	**0.561**	**0.557**

7.1.2 Protein Structure Ranking. Table 2 shows the performance comparisons of the GBP-GNN on the PSR task. We present the results in two sections, namely local and global. The local denotes the evaluation metric is computed per target, and the results are averaged to get final values. The global denotes evaluation metric applied to all samples in the test set. Our model achieved the best performance on every metric compared to all baseline methods. On average, an improvement over local metrics was over 5% compared to the best baseline method. Furthermore, similar improvements are observed in global metrics, where our model improved the baselines over 5%.

7.1.3 Ligand Binding Affinity. Table 3 shows the performance comparisons of the GBP-GNN on the LBA task. Our model is the only method that outperforms the 3DCNN method. GBP-GNN improves any other baseline approach in RMSE by more than 10%. Our model is also more reliable than any other model presented with more than two times lower standard deviation compared to any other model.

Table 4: Ablation study of proposed GBP-GNN.

				Perplexity ↓			Recovery ↑		
Type	Ablation	Parameters	Δt	Short	Single	All	Short	Single	All
Architecture	Scalar Only	1.588M	128s	7.281	7.567	7.247	28.71%	28.36%	31.99%
	- Vector Projection	1.780M	146s	6.794	6.493	5.273	31.24%	33.16%	40.90%
	- Vector Identity	1.677M	140s	6.157	6.486	5.026	32.20%	31.57%	42.11%
Node Features	- Scalar: Dihedral	1.677M	138s	7.855	8.301	5.386	29.87%	28.57%	39.87%
	- Vector: Orientation	1.677M	139s	7.124	7.392	6.206	29.55%	28.57%	36.28%
	- Vector: Sidechain	1.677M	139s	6.723	6.997	5.915	31.45%	29.96%	37.92%
Edge Features	- Scalar: Relative Distance	1.677M	137s	6.711	6.833	5.630	31.53%	30.43%	38.43%
	- Scalar: Relative Position	1.677M	138s	6.187	6.477	5.090	32.01%	31.23%	41.49%
	- Vector: Direction Unit	1.677M	141s	7.029	7.147	6.223	30.12%	29.42%	35.59%
	GBP	1.677M	145s	**6.144**	**6.458**	**5.025**	**33.22%**	**33.22%**	**42.70%**

7.2 Ablation Study

In this section, we perform ablation experiments over two types of variations of GBP to study the factors that affect the model performance. Table 4 presents the results of our evaluations. Δt denotes the average time to complete a training epoch in seconds.

7.2.1 Impact of the GBP expression paths. We compare the impact of the GBP expression paths for s and V in the GBP block, as shown in the first row of Table 4. The *Scalar Only* variation removes the (V) paths for nodes and edges. The model performance decreases severely on both perplexity and recovery scores. Thus the vector expression path that interacts with scalars is essential for geometric graph representation learning. The *Vector Projection* variation removes the bottleneck down-scaling and up-scaling operations on the V path when possible. The results show that the vector expression path with bottleneck successfully helps the model learn the protein geometric structures. The *Vector Identity* variation removes the element-wise multiplication is used to calculate V' in the GBP block. Although this variation performs similarly in terms of perplexity score with our proposed model, the recovery score decreased by 3% on average.

7.2.2 Impact of the geometric features. We compare the importance of the scalar and vector features by comparing the model's performance with removing one of the features. The second and third rows of Table 4 presents the results. The most salient feature to affect our model's performance is the *Direction Unit* vector. The *Direction Unit* feature provides relative orientation information of nodes, which is an essential indicator of the folding pattern for the protein. The absence of the *Direction Unit* feature reduced the model performance by 14%. Overall, the vector features are crucial for successfully interpreting the geometric representations. The improvements of the results prove the superiority of the novel design of the vector expression path in our GBP.

7.3 Sensitivity Analysis

This section investigates the model's hyper-parameter and dataset size sensitivity on the CPD task.

Table 5: The parameter sensitivity analysis of the proposed GBP-GNN model.

				Perplexity ↓			Recovery % ↑		
γ	δ	ω	PM	Short	Single	All	Short	Single	All
3	1	6	0.68	7.038	7.563	5.198	31.06	30.83	40.84
5	1	6	1.01	6.153	6.467	5.034	33.68	30.93	41.63
7	1	6	1.34	6.159	6.485	5.029	33.78	32.24	42.02
11	1	6	2.01	6.169	6.519	5.111	32.88	31.11	41.73
9	3	6	2.01	6.178	6.499	5.085	32.52	31.32	41.55
9	5	6	2.35	6.305	6.674	5.187	30.85	30.22	39.20
9	9	6	2.35	7.815	8.294	5.530	28.36	27.60	38.45
9	1	4	1.46	6.331	6.652	5.227	**33.81**	30.92	40.83
9	1	5	1.46	6.454	6.832	5.178	33.39	32.69	41.85
9	1	7	1.89	6.368	6.702	5.059	32.97	32.24	42.28
9	1	8	1.89	6.205	6.477	5.029	31.47	32.52	42.21
9	1	10	2.10	6.501	**6.209**	5.042	33.32	32.48	41.38
9	1	6	1.68	**6.144**	6.458	**5.025**	33.22	**33.22**	**42.70**

7.3.1 Parameter sensitivity. Table 5 shows the performance of the proposed model with different hyper-parameters. PM stands for the number of trainable parameters listed in millions. Each variant of the model only changes one parameter in $\{\gamma, \delta, \omega\}$ to compare with our base model. In the first row, the variance for different γ improves the model performance slightly until nine, and then the performance starts dropping. It is worth noting that the $\gamma = 3$ variant's performance is still competitive and could be a lightweight option for real-world applications. Although the change of the γ, δ, and ω decreases the model's performance slightly, the performances are still better than most of the benchmark results shown in Table 1, which confirms the robustness of the design of the GBP network architecture.

7.3.2 Dataset size sensitivity. Figure 4 presents the results for the sensitivity of size of dataset. Figure 4 shows both perplexity and recovery scores to explore further the relationship between the size

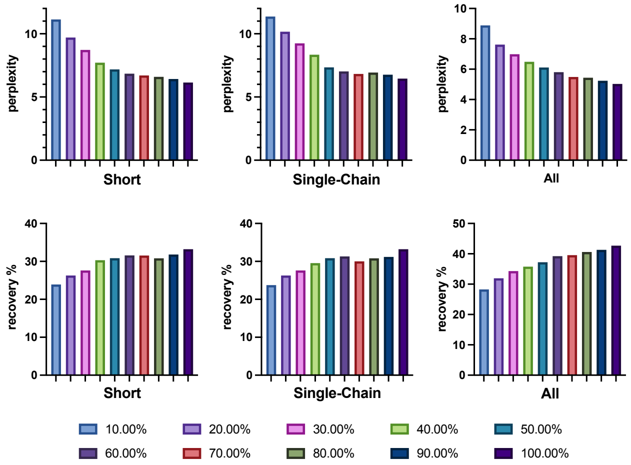


Figure 4: Sensitivity analysis of the size of the dataset

of the training dataset and the model's performance. The model scales well with the dataset size. It is evident in Figure 4 that there is a performance increase with the increase of the dataset size.

8 CONCLUSION

This paper focuses on learning geometric representations of the protein structures. We present GBP-GNN, a novel SO(3)-equivariant message passing neural network for learning the geometric representations of proteins. Moreover, we propose a drop-in module named *Geometric Bottleneck Perceptron* to integrate the geometric features and capture the complex geometric relations in the 3D structure. Our proposed GBP is a powerful and versatile module for learning and representing geometric features. We demonstrate the performance of GBP-GNN on three tasks, including Computational Protein Design, Protein Structure Ranking, and Ligand Binding Affinity task. The comparison with GBP-GNN and state-of-the-art methods validates the effectiveness of the proposed architecture.

REFERENCES

[1] Uri Alon and Eran Yahav. 2021. On the Bottleneck of Graph Neural Networks and its Practical Implications. In *International Conference on Learning Representations.*

[2] Brandon Anderson, Truong Son Hy, and Risi Kondor. 2019. Cormorant: Covariant molecular neural networks. *Advances in neural information processing systems* 32 (2019).

[3] Federico Baldassarre, David Menéndez Hurtado, Arne Elofsson, and Hossein Azizpour. 2021. GraphQA: protein model quality assessment using graph convolutional networks. *Bioinformatics* 37, 3 (2021), 360–366.

[4] Stephan Eismann, Raphael JL Townshend, Nathaniel Thomas, Milind Jagota, Bowen Jing, and Ron O Dror. 2021. Hierarchical, rotation-equivariant neural networks to select structural models of protein complexes. *Proteins: Structure, Function, and Bioinformatics* 89, 5 (2021), 493–501.

[5] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. 2020. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in Neural Information Processing Systems* 33 (2020), 1970–1981.

[6] Octavian-Eugen Ganea, Xinyuan Huang, Charlotte Bunne, Yatao Bian, Regina Barzilay, Tommi S. Jaakkola, and Andreas Krause. 2022. Independent SE(3)-Equivariant Models for End-to-End Rigid Protein Docking. In *International Conference on Learning Representations.*

[7] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*. PMLR, 1263–1272.

[8] Pedro Hermosilla, Marco Schäfer, Matej Lang, Gloria Fackelmann, Pere-Pau Vázquez, Barbora Kozlikova, Michael Krone, Tobias Ritschel, and Timo Ropinski. 2021. Intrinsic-Extrinsic Convolution and Pooling for Learning on 3D Protein Structures. In *International Conference on Learning Representations.*

[9] John Ingraham, Vikas K Garg, Regina Barzilay, and Tommi Jaakkola. 2019. Generative Models for Graph-Based Protein Design. In *Advances in Neural Information Processing Systems.*

[10] Bowen Jing, Stephan Eismann, Pratham N Soni, and Ron O Dror. 2021. Equivariant graph neural networks for 3D macromolecular structure. *arXiv preprint arXiv:2106.03843* (2021).

[11] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. 2021. Learning from Protein Structure with Geometric Vector Perceptrons. In *International Conference on Learning Representations.*

[12] Mikhail Karasikov, Guillaume Pagès, and Sergei Grudinin. 2019. Smooth orientation-dependent scoring function for coarse-grained protein quality assessment. *Bioinformatics* 35, 16 (2019), 2801–2808.

[13] Mostafa Karimi, Di Wu, Zhangyang Wang, and Yang Shen. 2019. DeepAffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics* 35, 18 (2019), 3329–3338.

[14] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR).*

[15] Johannes Klicpera, Janek Groß, and Stephan Günnemann. 2020. Directional Message Passing for Molecular Graphs. In *International Conference on Learning Representations.*

[16] Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moult. 2019. Critical assessment of methods of protein structure prediction (CASP)—Round XIII. *Proteins: Structure, Function, and Bioinformatics* 87, 12 (2019), 1011–1020.

[17] Shuangli Li, Jingbo Zhou, Tong Xu, Liang Huang, Fan Wang, Haoyi Xiong, Weili Huang, Dejing Dou, and Hui Xiong. 2021. Structure-aware Interactive Graph Neural Networks for the Prediction of Protein-Ligand Binding Affinity. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 975–985.

[18] Zhihai Liu, Yan Li, Li Han, Jie Li, Jie Liu, Zhixiong Zhao, Wei Nie, Yuchen Liu, and Renxiao Wang. 2014. PDB-wide collection of binding data: current status of the PDBbind database. *Bioinformatics* 31, 3 (10 2014), 405–412. https://doi.org/10.1093/bioinformatics/btu626

[19] Thin Nguyen, Hang Le, Thomas P Quinn, Tri Nguyen, Thuc Duy Le, and Svetha Venkatesh. 2021. GraphDTA: Predicting drug–target binding affinity with graph neural networks. *Bioinformatics* 37, 8 (2021), 1140–1147.

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

[21] Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. 2018. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics* 34, 17 (2018), i821–i829.

[22] Guillaume Pagès, Benoit Charmettant, and Sergei Grudinin. 2019. Protein model quality assessment using 3D oriented convolutional neural networks. *Bioinformatics* 35, 18 (2019), 3313–3319.

[23] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. 2021. E (n) equivariant graph neural networks. In *International Conference on Machine Learning*. PMLR, 9323–9332.

[24] Ian Sillitoe, Nicola Bordin, Natalie Dawson, Vaishali P Waman, Paul Ashford, Harry M Scholes, Camilla S M Pang, Laurel Woodridge, Clemens Rauer, Neeladri Sen, Mahnaz Abbasian, Sean Le Cornu, Su Datt Lam, Karel Berka, Ivana Hutařová Varekova, Radka Svobodova, Jon Lees, and Christine A Orengo. 2020. CATH: increased structural coverage of functional space. *Nucleic Acids Research* 49, D1 (11 2020), D266–D273. https://doi.org/10.1093/nar/gkaa1079

[25] Raphael John Lamarre Townshend, Martin Vögele, Patricia Adriana Suriana, Alexander Derry, Alexander Powers, Yianni Laloudakis, Sidhika Balachandar, Bowen Jing, Brandon M. Anderson, Stephan Eismann, Risi Kondor, Russ Altman, and Ron O. Dror. 2021. ATOM3D: Tasks on Molecules in Three Dimensions. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1).*

[26] Karolis Uziela, David Menéndez Hurtado, Nanjiang Shu, Björn Wallner, and Arne Elofsson. 2017. ProQ3D: improved model quality assessments using deep learning. *Bioinformatics* 33, 10 (2017), 1578–1580.

[27] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations.*

[28] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How Powerful are Graph Neural Networks?. In *International Conference on Learning Representations.*

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

A PROOF: ROTATIONAL TRANSFORMATION EQUIVARIANCE

The equivariance to a graph transformation achieved by satisfying the following property:

$$Q(f(g)) = f(Q(g)) | \forall g \in \mathcal{G} \tag{14}$$

where Q denotes a transformation operation, f is a linear or non-linear equivariant function, and g denotes a graph. In other words, if the result of applying Q before or after the f we call f equivariant to transformation Q. Hence, we prove our model is rotation invariant by defining any rotation function $Q_r(g)$:

$$Q_r(g) = \mathcal{R}g | \forall \mathcal{R} \in \mathbb{R}^{3\times 3} \tag{15}$$

where the rotation is applied by multiplying the input with rotation matrix $\mathcal{R}$. We construct edge and node features from g, and represent them with scalar and vectors (s, v). Therefore, we will prove that GBP is SO(3)-equivariant, showing that both s' and v' are equivariant. First, s' is affected by applying the rotation matrix to (s, v) from (5).

$$s' | (\mathcal{R}s, \mathcal{R}v) = \sigma_s(||\mathcal{R}v \cdot w_d||_2 \cup \mathcal{R}s \cdot w_s) \tag{16}$$

The s is invariant to rotations $\mathcal{R}s = s$ by definition since they are constructed from the local frame. Which reduces the problem to show $||\mathcal{R}v \cdot w_d||_2$ is equivariant. Since $||\mathcal{R}v \cdot w_d||_2 = \mathcal{R}||v \cdot w_d||_2 = ||v \cdot w_d||_2$, s' is invariant and therefore equivariant. Next, we show v' is equivariant by applying the $\mathcal{R}$ to v:

$$v' | (\mathcal{R}s, \mathcal{R}v) = (\mathcal{R}v * w_d * w_u) \odot \sigma_v(||\mathcal{R}v * w_d * w_u||_2) \tag{17}$$

Again, we see $||\mathcal{R}v * w_d * w_u||_2 = ||v * w_d * w_u||_2$ is invariant. Furthermore, $(\mathcal{R}v * w_d * w_u)$ is equivariant since $(\mathcal{R}v * w_d * w_u) = \mathcal{R}(v * w_d * w_u)$. Hence, we show both s' and v' is equivariant to the $Q_r(g)$.

B RESEARCH METHODS

B.1 Protein Geometric Representation

Residue Level Node Features:

- Scalar: Dihedral angles: The dihedral angles of the protein backbone structures are essential to parameterize the protein folding. Therefore, we computed the three dihedral angles (ϕ, ψ, ω) from C, N and Cα atoms of each three consecutive amino acids. To keep the circular property of the angles, the *sin* and *cos* values for each angle were calculated. Hence, dihedral angles are represented with six features per amino acid.
- Vector: Orientations: We calculate two unit vectors for forward and backwards to capture local orientations based on Cα positions. The forward unit vector is normalized result of Cα_{i+1} - Cα_i and backward unit vector is normalized result of Cα_{i-1} - Cα_i.
- Vector: Sidechains: The direction of Cβ_i atom with respect to Cα_i represented by a unit vector.

$$\sqrt{\frac{1}{3}}\frac{(n \times c)}{||n \times c||_2} - \sqrt{\frac{2}{3}}\frac{(n + c)}{||n + c||_2} \tag{18}$$

where $n = \mathrm{N}_i - \mathrm{C}\alpha_i$ and $c = \mathrm{C}_i - \mathrm{C}\alpha_i$. This unit vector in conjunction with the reverse and forward unit vectors are sufficient to define the orientation of the amino acids.

Residue Level Edge Features:

- Scalar: Relative Distance Embeddings: Euclidian distances between amino acids are essential for understanding the interactions. However, if the absolute values are provided to a network, it tends to memorize than learn the structure. Therefore, we provided the approximations of these values by using RBF kernels. The number of RBF kernels can change based on the problem. For that purpose, we leave them as hyper-parameter.
- Scalar: Relative Positional Embeddings: The relative distance in amino acid location provided by relative positonal embeddings. The positional embeddings for each sequence location is generated with a sinusoidal function as in Transformers paper. Hence, for each position i and j there is a positional embedding value P_i and P_j. For each edge, relative positional information is provided by computing $P_i - P_j$.
- Vector: Direction Unit Vectors: The direction of the edge is represented by the unit vectors calculated with normalization of Cα_i - Cα_j.

C PARAMETERS

We present the parameters used in GBP-GNN in this section. All the networks trained over 300 epochs if not stated otherwise. Furthermore, the one with the highest validation accuracy is selected for testing. The network was trained with Adam optimizer with an initial learning rate of 0.0001.

For the CPD task, we used nine encoder blocks with three decoder blocks. Ω calculation of encoder and decoder used eight GBP blocks stacked recursively with $\lambda = 4$. Five encoder blocks were used for the PSR task with $m = 2$. Eight encoder blocks were used for the LBA task on the experiments with $m = 2$.

D BENCHMARK DETAILS

- **ProQ3D** [26] uses the large set of manually annotated protein structures with their deep learning framework.
- **VoroMQA**[20] constructs Voronoi tessellation-based estimates the model quality by processing the contact areas with statistical potentials.
- **RWplus** [29] uses ideal random-walk chain as the reference state of high-resolution 3D structures.
- **SBROD** [12] proposes a scoring function that feeds four hand-picked features to Ridge Regression model.
- **Ornate** [22] evaluates the 3D structure in local and global stages. The local stage transforms the residue based on the local backbone topology.
- **DimeNet** [15] introduces directional message passing by using directional embeddings to achieve rotational equivariance. Furthermore, the authors show an effective way to represent distance and angles.
- **GraphQA** [3] adopts graph convolutional network to protein structure ranking task. Authors Incorporated the the domain-specific information into node and edge features.

- **GVP** [11] introduces different representation shapes for geometric structures and a paradigm to process them. The proposed architecture is flexible and applicable to tasks with 3D structures.
- **DeepAffinity** [13] proposes an RNN-CNN hybrid that takes structurally annotated protein sequences along with ligand SMILE sequences.
- **GraphDTA** [19] proposes architecture to predict affinity using CNNs and GNNs. It processes the protein structures with CNNs and the ligand structures with the variation of GNNs. The GNN variants uses GCN [14], GAT [27], GIN [28], GAT-GCN. The models' variations are denoted with 'D' as follows DGAT, DGIN, DGAT-GCN.
- **STrans** [9] proposes a transformer architecture with the ability to process structural information. The attention mechanism is limited to the neighbors for a given node. The neighbors were derived using the k-Nearest Neighbors graph.