# When Geometric Deep Learning Meets Pretrained Protein Language Models

Fang Wu[1†], Yu Tao[3†], Dragomir Radev[2] and Jinbo Xu[1,4*]

[1]Institute of AI Industry Research, Tsinghua University, Haidian Street, Beijing, 100084, China.
[2]Department of Computer Science, Yale University, New Haven, 06511, Connecticut, United States.
[3]School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China.
[4]Toyota Technological Institute at Chicago, Chicago, 60637, Illinois, United States.

*Corresponding author(s). E-mail(s): jinboxu@gmail.com;
Contributing authors: fw2359@columbia.com;
taoyu928@sjtu.edu.cn; dragomir.radev@yale.edu;
†These authors contributed equally to this work.

**Abstract**

Geometric deep learning has recently achieved great success in non-Euclidean domains, and learning on 3D structures of large biomolecules is emerging as a distinct research area. However, its efficacy is largely constrained due to the limited quantity of structural data. Meanwhile, protein language models trained on substantial 1D sequences have shown burgeoning capabilities with scale in a broad range of applications. Nevertheless, no preceding studies consider combining these different protein modalities to promote the representation power of geometric neural networks. To address this gap, we make the foremost step to integrate the knowledge learned by well-trained protein language models into several state-of-the-art geometric networks. Experiments are evaluated on a variety of protein representation learning benchmarks, including protein-protein interface prediction, model quality assessment, protein-protein rigid-body docking, and binding affinity prediction, leading to an overall improvement of 20% over baselines and the new state-of-the-art performance. Strong evidence indicates that the incorporation

of protein language models' knowledge enhances geometric networks' capacity by a significant margin and can be generalized to complex tasks.

**Keywords:** Geometric Deep Learning, Language Model, Protein Representation Learning

# 1 Introduction

Macromolecules (*e.g.*, proteins, RNAs, or DNAs) are essential to biophysical processes. While they can be represented using lower-dimensional representations such as linear sequences (1D) or chemical bond graphs (2D), a more intrinsic and informative form is the three-dimensional geometry [97]. 3D shapes are critical to not only understanding the physical mechanisms of action but also answering a number of questions associated with drug discovery and molecular design [86]. As a consequence, tremendous efforts in structural biology have been devoted to deriving insights from their conformations [53, 55, 96].

With the rapid advances of deep learning (DL) techniques, it has been an attractive challenge to represent and reason about macromolecules' structures in the 3D space. In particular, different sorts of 3D information including bond lengths and dihedral angles play an essential role. In order to encode them, a number of 3D geometric graph neural networks (GGNNs) or CNNs [9, 42, 44, 45, 81, 94, 95] have been proposed, and simultaneously achieve several crucial properties of Euclidean geometry such as E(3) or SE(3) equivariance and symmetry. Notably, they are important constituents of geometric deep learning (GDL), an umbrella term that generalizes networks to Euclidean or non-Euclidean domains [5].

Meanwhile, the anticipated growth of sequencing promises unprecedented data on natural sequence diversity. The abundance of 1D amino acid sequences has spurred increasing interest in developing protein language models at the scale of evolution, such as the series of ESM [54, 60, 71] and ProtTrans [25]. These protein language models are capable of capturing information about secondary and tertiary structures and can be generalized across a broad range of downstream applications. To be explicit, they have recently been demonstrated with strong capabilities in uncovering protein structures [54], predicting the effect of sequence variation on function [60], learning inverse folding [40] and many other general purposes [71].

Despite the fruitful progress in protein language models, no prior studies have considered enhancing GGNNs' ability by leveraging the knowledge of those protein language models. This is nontrivial because compared to sequence learning, 3D structures are much harder to obtain and thus less prevalent. Consequently, learning on the structure of proteins leads to a reduced amount of training data. For example, the SAbDab database [23] merely has 3K antibody-antigen structures without duplicate. The SCOPe database [16]

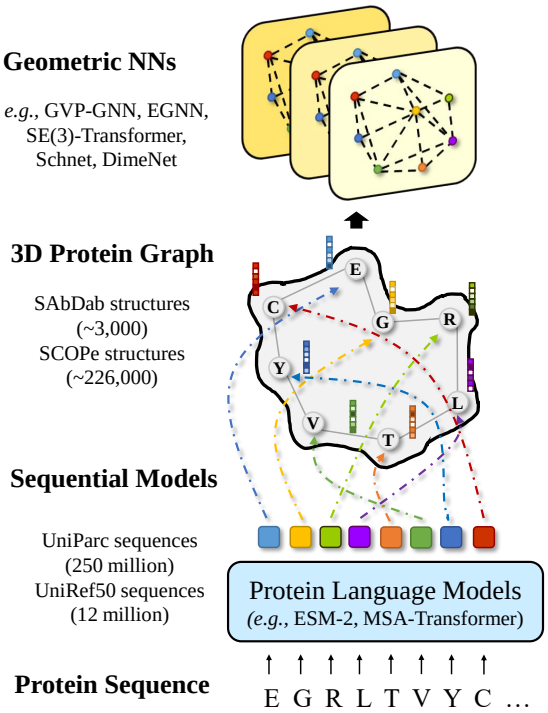


**Fig. 1**: Illustration of our proposed framework to strengthen GGNNs with knowledge of protein language models. The protein sequence is first forwarded into a pretrained protein language model to extract per-residue representations, which are then used as the node feature in 3D protein graphs for GGNNs.

has 226K annotated structures, and the SIFTS database [89] comprises around 220K annotated enzyme structures. These numbers are orders of magnitude lower than the data set sizes that can inspire major breakthroughs in the deep learning community. In contrast, while the Protein Data Bank (PDB) [12] possesses approximately 182K macromolecule structures, databases like Pfam [61] and UniParc [8] contains more than 47M and 250M protein sequences respectively.

In addition to the data size, the benefit of protein sequence to structure learning also has solid evidence and theoretical support. Remarkably, the idea that biological function and structures are documented in the statistics of protein sequences selected through evolution has a long history [2, 98]. The unobserved variables that decide a protein's fitness, including structure, function, and stability, leave a record in the distribution of observed natural sequences [35]. Those protein language models use self-supervision to unlock the information encoded in protein sequence variations, which is also beneficial for GGNNs.

Accordingly, in this paper, we propose to promote the capacity of GGNNs with the knowledge learned by protein language models (see Figure 1). The improvements come from two major lines. Firstly, GGNNs can benefit from the information that emerges in the learned representations of those protein language models on fundamental properties of proteins, including secondary structures, contacts, and biological activity. This kind of knowledge may be difficult for GGNNs to be aware of and learn in a specific downstream task. To confirm this claim, we conduct a toy experiment to demonstrate that conventional graph connectivity mechanisms prevent existing GGNNs from being cognizant of residues' absolute and relative positions in the protein sequence. Secondly and more intuitively, protein language models serve as an alternative way of enriching GGNNs' training data and allow GGNNs to be exposed to more different families of proteins, which thereby greatly strengthens GGNNs' generalization capability.

We examine our hypothesis across a wide range of benchmarks, containing model quality assessment, protein-protein interface prediction, protein-protein rigid-body docking, and ligand binding affinity prediction. Extensive experiments show that the incorporation and combination of pretrained protein language models' knowledge significantly improve GGNNs' performance for various problems, which require distinct domain knowledge. By utilizing the unprecedented view into the language of protein sequences provided by powerful protein language models, GGNNs promise to augment our understanding of a vast database of poorly understood protein structures. Our work provides a new perspective on protein representation learning and hopes to shed light on how to bridge the gap between the thriving geometric deep learning and mature protein language models, which independently leverage different modalities of proteins.

## 2 Experiments

Our toy experiments illustrate that existing GGNNs are unaware of the positional order inside the protein sequences. Taking a step further, we show in this section that incorporating knowledge learned by large-scale protein language models can robustly enhance GGNN's capacity in a wide variety of downstream tasks.

### 2.1 Tasks and Datasets

- **Model Quality Assessment** (MQA) aims to select the best structural model of a protein from a large pool of candidate structures and is an essential step in structure prediction [17]. For a number of recently solved but unreleased structures, structure generation programs produce a large number of candidate structures. MQA approaches are evaluated by their capability of predicting the GDT-TS score of a candidate structure compared to the experimentally solved structure of that target. Its

database is composed of all structural models submitted to the Critical Assessment of Structure Prediction (CASP) [52] over the last 18 years. The data is split temporally by competition year. MQA is similar to the Protein Structure Ranking (PSR) task introduced by Townshend et al. [86].

- **Protein-protein Rigid-body Docking** (PPRD) computationally predicts the 3D structure of a protein-protein complex from the individual unbound structures. It assumes that no conformation change within the proteins happens during binding. We leverage Docking Benchmark 5.5 (DB5.5) [91] as the database. It is a gold standard dataset in terms of data quality and contains 253 structures.
- **Protein-protein Interface** (PPI) investigates whether two amino acids will contact when their respective proteins bind. It is an important problem in understanding how proteins interact with each other, *e.g.*, antibody proteins recognize diseases by binding to antigens. We use the Database of Interacting Protein Structures (DIPS), a comprehensive dataset of protein complexes mined from the PDB [85], and randomly select 15K samples for evaluation.
- **Ligand Binding Affinity** (LBA) is an essential task for drug discovery applications. It predicts the strength of a candidate drug molecule's interaction with a target protein. Specifically, we aim to forecast $pK = -\log_{10} K$, where $K$ is the binding affinity in Molar units. We use the PDBbind database [57, 93], a curated database containing protein-ligand complexes from the PDB and their corresponding binding strengths. The protein-ligand complexes are split such that no protein in the test dataset has more than 30% or 60% sequence identity with any protein in the training dataset.

## 2.2 Experimental Setup

We evaluate our proposed framework on the instances of several state-of-the-art geometric networks, using Pytorch [66] and PyG [28] on four standard protein benchmarks. For MQA, PPI, and LBA, we use backbones that have been carefully described in Section 5, *i.e.*, GVP-GNN, EGNN, and Molformer. For PPRD, we utilize the state-of-the-art deep learning model, EquiDock [32], as the backbone. It approximates the binding pockets and obtains the docking poses using keypoint matching and alignment. For more experimental details, please refer to Appendix A.

**Table 1**: Results on MQA.

| Model | PLM | Model Quality Assessment | | | | | |
|---|---|---|---|---|---|---|---|
| | | Spearman Correlation↑ | | Pearson's Correlation↑ | | Kendall Rank↑ | |
| | | Mean | Global | Mean | Global | Mean | Global |
| GVP-GNN | ✗ | 0.4144 ± 0.010 | 0.6910 ± 0.008 | 0.5235 ± 0.013 | 0.6875 ± 0.006 | 0.2960 ± 0.010 | 0.4959 ± 0.004 |
| | ✓ | **0.6121 ± 0.017** | **0.8492 ± 0.015** | **0.7399 ± 0.017** | **0.8544 ± 0.009** | **0.4530 ± 0.008** | **0.6798 ± 0.014** |
| EGNN | ✗ | 0.4249 ± 0.016 | 0.7341 ± 0.015 | 0.5315 ± 0.008 | 0.7336 ± 0.018 | 0.3004 ± 0.013 | 0.5344 ± 0.011 |
| | ✓ | **0.5642 ± 0.013** | **0.8436 ± 0.012** | **0.6925 ± 0.006** | **0.8456 ± 0.015** | **0.4105 ± 0.014** | **0.6558 ± 0.006** |
| Molformer | ✗ | 0.1238 ± 0.011 | 0.3921 ± 0.004 | 0.1969 ± 0.004 | 0.3901 ± 0.012 | 0.0841 ± 0.010 | 0.2696 ± 0.005 |
| | ✓ | **0.2424 ± 0.015** | **0.6516 ± 0.009** | **0.3850 ± 0.011** | **0.6210 ± 0.014** | **0.1681 ± 0.012** | **0.4579 ± 0.007** |

## 2.3 Results

Results are reported with mean ± standard deviation over three repeated runs, where the best performance is in bold. The column of 'PLM' indicates whether the protein language model is used.

### 2.3.1 Single-protein Representation Task

For MQA, we document Spearman correlation ($R_S$), Pearson's correlation ($R_P$), and Kendall rank correlation ($K_R$) (see Table 1). The introduction of protein language models has brought a significant average increase of 32.63% and 55.71% in global and mean $R_S$, of 34.66% and 58.75% in global and mean $R_P$, and of 43.21% and 63.20% in global and mean $K_R$ respectively. With the aid of language models, GVP-GNN achieves the optimal global $R_S$, global $R_P$, and $K_R$ of 84.92%, 85.44%, and 67.98% separately.

Apart from that, we provide a full comparison with all existing approaches in Table 2. We elect RWplus [99], ProQ3D [88], VoroMQA [64], SBROD [47], 3DCNN [86], 3DGNN [86], 3DOCNN [65], DimeNet [51], GraphQA [24], and GBPNet [6] as the baselines. Performance is recorded in Table 2, where the second best is underlined. It can be concluded that even if GVP-GNN is not the best architecture, it can largely outperform existing methods including the state-of-the-art no-pretraining method set by Aykent and Xia [6] (*i.e.*, GBPNet) and the state-of-the-art pretraining results set by Jing et al. [45] and set the new state-of-the-art if it is enhanced by the protein language model.

### 2.3.2 Protein-protein Representation Tasks

For PPRD, we report three items as the measurements: the complex root mean squared deviation (RMSD), the ligand RMSD, and the interface RMSD (see Table 3). The interface is determined with a distance threshold less than 8Å. It is noteworthy that, unlike the EquiDock paper, we do not apply the Kabsch algorithm to superimpose the receptor and the ligand. Contrastingly, the receptor protein is fixed during evaluation. All three metrics decrease considerably with improvements of 11.61%, 12.83%, and 31.01% in complex, ligand, and interface median RMSD, respectively. Notably, we also report the result of EquiDock, which is first pretrained on DIPS and then fine-tuned on DB5. It can be discovered that DIPS-pretrained EquiDock still performs worse than

**Table 2**: Comparison of performance on MQA. Models are sorted by the year they are released.

| Model | PLM | Model Quality Assessment | | | | | |
|---|---|---|---|---|---|---|---|
| | | Spearman Correlation↑ | | Pearson's Correlation↑ | | Kendall Rank↑ | |
| | | Mean | Global | Mean | Global | Mean | Global |
| RWplus [99][1] | ✗ | 0.167 | 0.056 | 0.192 | 0.033 | 0.137 | 0.011 |
| ProQ3D [88][1] | ✗ | 0.432 | 0.772 | 0.444 | 0.796 | 0.304 | 594 |
| VoroMQA [64][1] | ✗ | 0.419 | 0.651 | 0.412 | 0.651 | 0.291 | 0.505 |
| SBROD [47][1] | ✗ | 0.413 | 0.569 | 0.431 | 0.551 | 0.291 | 0.393 |
| 3DOCNN [65][2] | ✗ | 0.432 | 0.796 | 0.444 | 0.772 | 0.304 | 0.594 |
| DimeNet [51][1] | ✗ | 0.351 | 0.625 | 0.302 | 0.614 | 0.285 | 0.431 |
| 3DCNN [86][2] | ✗ | $0.431 \pm 0.013$ | $0.789 \pm 0.017$ | $0.557 \pm 0.011$ | $0.780 \pm 0.016$ | $0.308 \pm 0.010$ | $0.592 \pm 0.016$ |
| 3DGNN [86][2] | ✗ | $0.411 \pm 0.006$ | $0.750 \pm 0.018$ | $0.500 \pm 0.012$ | $0.747 \pm 0.018$ | $0.278 \pm 0.005$ | $0.547 \pm 0.016$ |
| GVP-GNN [44][3] | ✗ | $0.414 \pm 0.010$ | $0.691 \pm 0.008$ | $0.523 \pm 0.013$ | $0.687 \pm 0.006$ | $0.296 \pm 0.010$ | $0.495 \pm 0.004$ |
| GraphQA [24][1] | ✗ | 0.379 | 0.820 | 0.357 | 0.821 | 0.331 | 0.618 |
| GBPNet [6][1] | ✗ | <u>0.517</u> | **0.856** | <u>0.612</u> | <u>0.853</u> | <u>0.372</u> | <u>0.656</u> |
| GVP-GNN | ✓ | $\mathbf{0.612 \pm 0.017}$ | <u>$0.849 \pm 0.015$</u> | $\mathbf{0.739 \pm 0.017}$ | $\mathbf{0.854 \pm 0.009}$ | $\mathbf{0.453 \pm 0.008}$ | $\mathbf{0.679 \pm 0.014}$ |

[1]These results are taken from Aykent and Xia [6].

[2]These results are taken from Townshend et al. [86].

[3]These results are re-produced.

**Table 3**: Performance of PPRD on DB5.5 Test Set. Models with ♣ are directly trained and tested on DB5, while EquiDock with ♠ is first pretrained on DIPS and then fine-tuned on the DB5 training set.

| Model | PLM | Protein-protein Rigid-body Docking | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Complex RMSD ↓ | | | Ligand RMSD ↓ | | | Interface RMSD ↓ | | |
| | | Median | Mean | Std | Median | Mean | Std | Median | Mean | Std |
| EquiDock | ✗♣ | 16.88 | 17.11 | 5.33 | 40.35 | 37.97 | **12.94** | 16.19 | 37.97 | 4.47 |
| | ✗♠ | 15.02 | 14.31 | 5.28 | 36.82 | 35.95 | 13.18 | 14.37 | 35.68 | **4.12** |
| | ✓♣ | **14.92** | **13.14** | **4.59** | **35.17** | **33.48** | 14.34 | **11.17** | **33.48** | 4.38 |

**Table 4**: Results on LBA.

| Model | PLM | Ligand Binding Affinity | | | |
|---|---|---|---|---|---|
| | | Sequence Identity (30%) | | | |
| | | RMSD↓ | Pearson's Correlation↑ | Spearman Correlation↑ | Kendall Rank↑ |
| GVP-GNN | ✗ | 1.6480 ± 0.014 | 0.2138 ± 0.013 | 0.1648 ± 0.009 | 0.1107 ± 0.012 |
| | ✓ | **1.4556 ± 0.011** | **0.5373 ± 0.010** | **0.5078 ± 0.005** | **0.3495 ± 0.009** |
| EGNN | ✗ | 1.4929 ± 0.012 | 0.4891 ± 0.017 | 0.4725 ± 0.008 | 0.3291 ± 0.014 |
| | ✓ | **1.4033 ± 0.013** | **0.5655 ± 0.016** | **0.5448 ± 0.005** | **0.3790 ± 0.007** |
| Molformer | ✗ | 1.9107 ± 0.018 | 0.4618 ± 0.014 | 0.4104 ± 0.011 | 0.2812 ± 0.019 |
| | ✓ | **1.6028 ± 0.020** | **0.5351 ± 0.017** | **0.5372 ± 0.015** | **0.3758 ± 0.016** |
| | | Sequence Identity (60%) | | | |
| GVP-GNN | ✗ | 1.5438 ± 0.015 | 0.6608 ± 0.012 | 0.6668 ± 0.0010 | 0.4797 ± 0.014 |
| | ✓ | **1.5137 ± 0.019** | **0.6680 ± 0.010** | **0.6716 ± 0.008** | **0.4786 ± 0.012** |
| EGNN | ✗ | 1.5928 ± 0.020 | 0.6274 ± 0.013 | 0.6271 ± 0.017 | 0.4498 ± 0.014 |
| | ✓ | **1.5595 ± 0.022** | **0.6445 ± 0.015** | **0.6463 ± 0.019** | **0.4656 ± 0.019** |
| Molformer | ✗ | 1.8610 ± 0.018 | 0.5528 ± 0.016 | 0.5309 ± 0.015 | 0.3738 ± 0.017 |
| | ✓ | **1.5926 ± 0.024** | **0.6524 ± 0.018** | **0.6528 ± 0.016** | **0.4367 ± 0.011** |

EquiDock equipped with pretrained language models. This strongly demonstrates that structural pretraining for GGNNs may not benefit GGNNs more than pretrained protein language models.

For PPI, we record AUROC as the metric in Figure 2. It can be found that AUROC increases for 6.93%, 14.01%, and 22.62% for GVP-GNN, EGNN, and Molformer respectively. It is worth noting that Molformer falls behind EGNN and GVP-GNN originally in this task. But after injecting knowledge learned by protein language models, Molformer achieves competitive or even better performance than EGNN or GVP-GNN. This indicates that protein language models can realize the potential of GGNNs to the full extent and greatly narrow the gap between different geometric deep learning architectures. The results mentioned above are amazing because, unlike MQA, PPRD and PPI study the geometric interactions between two proteins. Though existing protein language models are all trained on single protein sequences, our experiments show that

**Table 5**: Comparison of performance on LBA with 30% sequence identity. Models are sorted by the year they are released.

| Model | PLM | **Ligand Binding Affinity Sequence Identity (30%)** | | | |
|---|---|---|---|---|---|
| | | RMSD↓ | Pearson's Correlation↑ | Spearman Correlation↑ | Kendall Rank↑ |
| DeepAffinity [48][1] | ✗ | $1.893 \pm 0.650$ | 0.415 | 0.426 | — |
| Cormorant [4][2] | ✗ | $1.568 \pm 0.012$ | 0.389 | 0.408 | — |
| LSTM [11][3] | ✗ | $1.985 \pm 0.006$ | $0.165 \pm 0.006$ | $0.152 \pm 0.024$ | — |
| TAPE [68][3] | ✗ | $1.890 \pm 0.035$ | $0.338 \pm 0.044$ | $0.286 \pm 0.124$ | — |
| ProtTrans [25][3] | ✗ | $1.544 \pm 0.015$ | $0.438 \pm 0.053$ | $0.434 \pm 0.058$ | — |
| 3DCNN [86][1] | ✗ | $1.414 \pm 0.021$ | 0.550 | 0.553 | — |
| GNN [86][1] | ✗ | $1.570 \pm 0.025$ | 0.545 | 0.533 | — |
| MaSIF [31][3] | ✗ | $1.484 \pm 0.018$ | $0.467 \pm 0.020$ | $0.455 \pm 0.014$ | — |
| DGAT [63][2] | ✗ | $1.719 \pm 0.047$ | 0.464 | 0.472 | — |
| DGIN [63][2] | ✗ | $1.765 \pm 0.076$ | 0.426 | 0.432 | — |
| DGAT-GCN [63][2] | ✗ | $1.550 \pm 0.017$ | 0.498 | 0.496 | — |
| GVP-GNN [44][4] | ✗ | $1.648 \pm 0.014$ | $0.213 \pm 0.013$ | $0.164 \pm 0.009$ | $0.110 \pm 0.012$ |
| EGNN [73][4] | ✗ | $1.492 \pm 0.012$ | $0.489 \pm 0.017$ | $0.472 \pm 0.008$ | $\underline{0.329 \pm 0.014}$ |
| HoloProt [79][3] | ✗ | $1.464 \pm 0.006$ | $0.509 \pm 0.002$ | $0.500 \pm 0.005$ | — |
| GBPNet [6][2] | ✗ | $\underline{1.405 \pm 0.009}$ | $\underline{0.561}$ | **0.557** | — |
| EGNN | ✓ | $\mathbf{1.403 \pm 0.013}$ | $\mathbf{0.565 \pm 0.016}$ | $\underline{0.544 \pm 0.005}$ | $\mathbf{0.379 \pm 0.007}$ |

[1]These results are taken from Townshend et al. [86].

[2]These results are taken from Aykent and Xia [6].

[3]These results are copied from Somnath et al. [79].

[4]These results are re-produced.

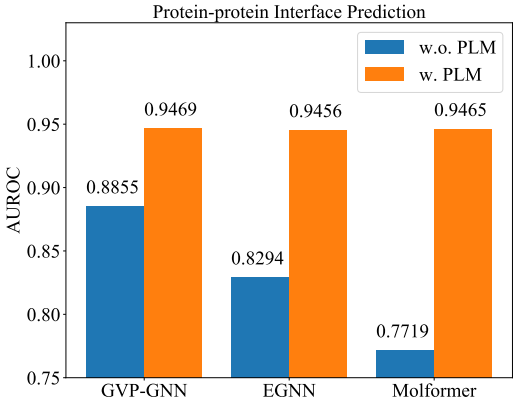


**Fig. 2**: Results of PPI.

the evolution information hidden in unpaired sequences can also be valuable to analyze the multi-protein environment.

### 2.3.3 Protein-molecules Representation Task

For LBA, we compare RMSD, $R_S$, $R_P$, and $K_R$ in Table 4. The incorporation of protein language models produces a remarkably average decline of 11.26% and 6.15% in RMSD for 30% and 60% identity, an average increase of 51.09% and 9.52% in $R_P$ for the 30% and 60% identity, an average increment of 66.60% and 8.90% in $R_S$ for the 30% and 60% identity, and an average increment of 68.52% and 6.70% in $K_R$ for the 30% and 60% identity. It can be seen that the improvements in the 30% sequence identity is higher than that in the less restrictive 60% sequence identity. This confirms that protein language models benefit GGNNs more when the unseen samples belong to different protein domains. Moreover, contrasting PPRD or PPI, LBA studies how proteins interact with small molecules. Our outcome demonstrates that rich protein representations encoded by protein language models can also contribute to the analysis of protein's reaction to other non-protein drug-like molecules.

In addition, we compare thoroughly with all existing approaches for LBA in Table 5. We select a broad range of models including DeepAffinity [48], Cormorant [4], LSTM [11], TAPE [68], ProtTrans [25], 3DCNN [86], GNN [86], MaSIF [31], DGAT [63], DGIN [63], DGAT-GCN [63], HoloProt [79], and GBPNet [6] as the baseline. We report the comparison in Table 5, where the second best is underlined. It is clear that even if EGNN is a median-level architecture, it can achieve the best RMSD and the best Pearson's correlation when enhanced by protein language models, beating a group of strong baselines including HoloProt [79] and GBPNet [6].

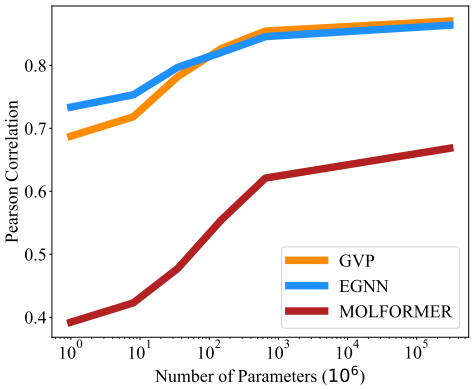


**Fig. 3**: Performance of GGNNs on MQA with ESM-2 at different scales.

### 2.4 Scale of Protein Language Models

It has been observed that as the size of the language model increases, there are consistent improvements in tasks like structure prediction [54]. Here we conduct an additional ablation study to investigate the effect of protein language models' sizes on GGNNs. Specifically, we explore different ESM-2 with the parameter numbers of 8M, 35M, 150M, 650M, and 3B. Results are plotted in Figure 3, which verifies that scaling the protein language model is advantageous for GGNNs.

## 3 Conclusion

In this study, we investigate a problem that has been long ignored by existing geometric deep learning methods for proteins. That is, how to employ the abundant protein sequence data for 3D geometric representation learning. To answer this question, we propose to leverage the knowledge learned by existing advanced pre-trained protein language models, and use their amino acid representations as the initial features. We conduct a variety of experiments such as protein-protein docking and model quality assessment to demonstrate the efficacy of our approach. Our work provides a simple but effective mechanism to bridge the gap between 1D sequential models and 3D geometric neural networks, and hope to throw light on how to combine information encoded in different protein modalities.

## 4 Method

As discussed before, learning on 3D structures cannot benefit from these large amounts of sequential data. Due to this fact, model sizes of those GGNNs are therefore limited or overfitting may occur [39]. On the contrary, it can be seen, comparing the number of protein sequences in the UniProt database [19] to the number of known structures in the PDB, over 1700 times more sequences

than structures. More importantly, the availability of new protein sequence data continues to far outpace the availability of experimental protein structure data, only increasing the need for accurate protein modeling tools.

Therefore, it is straightforward to assist GGNNs with pretrained protein language models. To this end, we feed amino acid sequences into those protein language models, where the state-of-the-art ESM-2 [54] is adopted in our case, and extract the per-residue representations, denoted as $\mathbf{h}' \in \mathbb{R}^{N \times \psi_{PLM}}$. Here $\psi_{PLM} = 1280$. Then $\mathbf{h}'$ can be added or concatenated to the per-atom feature $\mathbf{h}$. For residue-level graphs, $\mathbf{h}'$ immediately replaces the original $\mathbf{h}$ as the input node features.

Notably, incompatibility exists between the experimental structure and its original amino acid sequence. That is, structures stored in the PDB files are usually incomplete and some strings of residues are missing due to inevitable realistic issues [22]. They, therefore, do not perfectly match the corresponding sequences (*i.e.,* FASTA sequence). There are two choices to address this mismatch. On the one hand, we can simply use the fragmentary sequence as the substitute for the integral amino acid sequence and forward it into the protein language models. On the other hand, we can leverage a dynamic programming algorithm provided by Biopython [18] to implement pairwise sequence alignment and abandon residues that do not exist in the PDB structures. It is empirically discovered that no big difference exists between them, so we adopt the former processing mechanism for simplicity.

## 5 Sequence Recovery Analysis

It is commonly acknowledged that protein structures maintain much more information than their corresponding amino acid sequences. And for decades long, it has been an open challenge for computational biologists to predict protein structure from its amino acid sequence [49, 76]. Though the advancement of Alphafold (AF) [46], as well as RosettaFold [7], have made a huge step in alleviating the limitation brought by the number of available experimentally determined protein structures [40], neither AF nor its successors such as Alphafold-Multimer [26], IgFold [72], and HelixFold [92] are a panacea. Their predicted structures can be severely inaccurate when the protein is orphan and lacks multiple sequence alignment (MSA) as the template. As a consequence, it is hard to conclude that protein sequences can be perfectly transformed to the structure modality by current tools and be used as extra training resources for GGNNs.

Moreover, we argue that even if conformation is a higher-dimensional representation, the prevailing learning paradigm may forbid GGNNs from capturing the knowledge that is uniquely preserved in protein sequences. Recall that GGNNs are mainly diverse in their patterns to employ 3D geometries, the input features include distance [74], angles [50, 51], torsion, and terms of other orders [56]. The position index hidden in protein sequences, however, is usually neglected when constructing 3D graphs for GGNNs. Therefore, in this section,

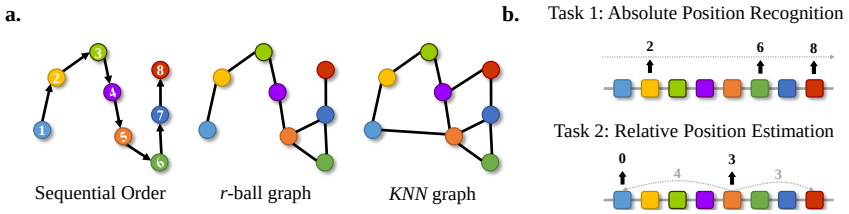


**Fig. 4**: (a) Protein residue graph construction. Here we draw graphs in 2D for better visualization but study 3D graphs for GGNNs. (b) Two sequence recovery tasks. The first requires GGNNs to predict the absolute position index for each residue in the protein sequence. The second aims to forecast the minimum distance of each amino acid to the two sides of the protein sequence.

**Table 6**: Results of two residue position identification tasks.

| Target | Metric | GVP | | EGNN | | Molformer |
|---|---|---|---|---|---|---|
| | | r-ball graph | KNN graph | r-ball graph | KNN graph | FC graph |
| Index | Accuracy (%) ↑ | 0.157 | 0.158 | 0.150 | 0.131 | 0.148 |
| Distance | RMSE ↓ | 392.38 | 392.38 | 412.70 | 403.86 | 270.69 |

we design a toy trial to examine whether GGNNs can succeed in recovering this kind of positional information.

## 5.1 Protein Graph Construction

Here the structure of a protein can be represented as an atom-level or residue-level graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E} = (e_{ij})$ correspond to the set of $N$ nodes and $M$ edges respectively. Nodes have their 3D coordinates as $\mathbf{x} \in \mathbb{R}^{N\times 3}$ as well as the initial $\psi_h$-dimension roto-translational invariant features $\mathbf{h} \in \mathbb{R}^{N\times\psi_h}$ (*e.g.*, atom types and electronegativity, residue classes). Normally, there are three types of options to construct connectivity for molecules: *r-ball* graphs, *fully-connected* (FC) graphs, and *K-nearest neighbors* (KNN) graphs. In our setting, nodes are linked to $K = 10$ nearest neighbors for KNN graphs, and edges include all atom pairs within a distance cutoff of 8Å for r-ball graphs.

## 5.2 Recovery from Graphs to Sequences

Since most prior studies choose to establish 3D protein graphs based on purely geometric information and ignore their sequential identities, it provokes the following position identity question:

*Can existing GGNNs identify the sequential position order only from geometric structures of proteins?*

To answer this question, we formulate two categories of toy tasks (see Figure 4). The first one is a classification task, where models are asked to directly predict the position index ranging from 1 to $N$, the residue number of each protein. This task adopts accuracy as the metric and expects models to

discriminate the absolute position of the amino acid within the whole protein sequence.

In addition to that, we propose the second task to focus on the relative position of each residue, where models are required to predict the minimum distance of residue to the two sides of the given protein. We use the root mean squared error (RMSE) as the metric. This task aims to examine the capability of GGNNs to distinguish which segment the amino acid belongs to (*i.e.*, the center section of the protein or the end of the protein).

## 5.3 Experimental Setting

We adopt three technically distinct and broadly accepted architectures of GGNNs for empirical verification. To be specific, **GVP-GNN** [44, 45] extends standard dense layers to operate on collections of Euclidean vectors, performing both geometric and relational reasoning on efficient representations of macromolecules. **EGNN** [73] is a translation, rotation, reflection, and permutation equivariant GNN without expensive spherical harmonics. **Molformer** [95] employs the self-attention mechanism for 3D point clouds while guarantees SE(3)-equivariance.

We exploit a small non-redundant subset of high-resolution structures from the PDB. To be specific, we use only X-ray structures with resolution $< 3.0$Å, and enforce a 60% sequence identity threshold. This results in a total of 2643, 330, and 330 PDB structures for the train, validation, and test sets, respectively. Experimental details, the summary of the database, and the description of these GGNNs are elaborated in Appendix A.

## 5.4 Results and Analysis

Table 6 documents the overall results, where metrics are labeled with ↑/↓ if higher/lower is better, respectively. It can be found that all GGNNs fail to recognize either the absolute or the relative positional information encoded in the protein sequences with an accuracy lower than 1% and a high RMSE.

This phenomenon stems from the conventional ways to build graph connectivity, which usually excludes sequential information. To be specific, unlike common applications of GNNs such as citation networks [75, 83], social networks [27, 36], knowledge graphs [15], molecules do not have explicitly defined edges or adjacency. On the one hand, r-ball graphs utilize a cut-off distance, which is usually set as a hyperparameter, to determine the particle connections. But it is hard to guarantee a cut-off to properly include all crucial node interactions for complicated and large molecules. On the other hand, FC graphs that consider all pairwise distances will cause severe redundancies, dramatically increasing the computational complexity especially when proteins consist of thousands of residues. Besides, GGNNs also easily get confused by excessive noise, leading to unsatisfactory performance. As a remedy, KNN becomes a more popular choice to establish graph connectivity for proteins [29, 32, 80].

However, all of them take no account of the sequential information and require GGNNs to learn this original sequential order during training.

The lack of sequential information can yield several problems. To begin with, residues are unaware of their relative positions in the proteins. For instance, two residues can be close in the 3D space but distant in the sequence, which can mislead models to find the correct backbone chain. Secondly, according to the characteristics of the MP mechanism, two residues in a protein with the same neighborhood are expected to share similar representations. Nevertheless, the role of those two residues can be significantly separate [62] when they are located at different segments of the protein. Thus, GGNNs may be incapable of differentiating two residues with the same 1-hop local structures. This restriction has already been distinguished by several works [42, 100], but none of them make a strict and thorough investigation. Admittedly, sequential order may only be necessary for certain tasks. But this toy experiment strongly indicates that the knowledge monopolized by amino acid sequences can be lost if GGNNs only learn from protein structures.

# 6 Related Work

## 6.1 Geometric Deep Learning for Proteins

Gigantic molecules (*i.e.*, macromolecules) populate a cell, providing it with irreplaceable functions for life. And past few years have witnessed growing attention in learning on their 3D structures. Early work adopts graph kernels and support vector machines to classify enzymes based on their conformations [14]. Later, inspired by the booming development of computer vision, protein tertiary structures are represented as 3D density maps with 3DCNNs to address a host of problems such as protein binding site prediction [43], enzyme classification [3], protein-ligand binding affinity [67], protein quality assessment [21], and protein-protein interaction interface identification [85].

At the same time, due to the fact that molecules can be naturally modeled as graphs in real-world studies, GGNNs have emerged as the mainstream line to learn directly from the protein spatial neighboring graphs. They show promising capacity in many tasks including protein interface prediction [29], protein design [42, 45, 81], protein quality assessment [9], function prediction [34], and in more challenging ones like protein folding [39].

Most existing GGNNs rely on the widely adopted message passing (MP) paradigm [20, 33, 56] to aggregate local neighborhood information for the update of node features. Their divergence mainly lies in how to exploit different types of 3D information such as bond lengths or dihedral angles [41, 50, 51, 56]. Moreover, equivariance is regarded as a ubiquitous property for molecular systems, and plenty of evidence has proven the effectiveness to integrate such inductive bias into GGNNs for modeling 3D geometry [4, 10, 30, 84]. Nevertheless, the potential of GGNNs is largely underestimated and cannot be fully released owing to the sparsity of structure data.

### 6.2 Protein Language Modeling

A large body of work has focused on protein language modeling in individual protein families, solving problems like functional nanobody design [78] and protein sequence generation [87]. This success has triggered a prospective trend to model large-scale databases of protein sequences rather than families of related sequences, where unsupervised learning becomes the preferred option. To be explicit, Bepler and Berger [11] combine unsupervised sequence pretraining with structural supervision to generate sequence embeddings. Alley et al. [1] and Heinzinger et al. [38] demonstrate that LSTM language models are able to capture certain biological properties. In the meantime, Rao et al. [68] evaluated a variety of protein language models across a panel of benchmarks concluding that small LSTMs and Transformers fall well short of features from the bioinformatics pipeline. Rives et al. [71] start to model protein sequences with self-attention, illustrating that Transformer-based protein language models can seize accurate information of structure and function in their representations.

A combination of model scale and architecture improvements has been critical to recent successes in protein language modeling. Elnaggar et al. [25] analyze a diversity of Transformer variants. Rives et al. [71] prove that large Transformer models can achieve state-of-the-art features across various tasks. Vig et al. [90] discover that specific attention heads of pretrained Transformers have immediate correlations with protein contacts. Moreover, Rao et al. [69] find that the combination of multiple attention heads is more accurate than Potts models in contact prediction, even if using a single sequence for inference.

For the sake of better capturing the biochemical knowledge, a group of typical pretraining objectives is explored including next amino acid prediction [1, 25], masked language modeling (MLM) [37], contrastive predictive coding [58], and conditional generation [59]. Besides that, Sturmfels et al. [82] and Sercu et al. [77] study alternative objectives with sets of sequences for supervision. Sturmfels et al. [82] extend the unsupervised language modeling to forecast the position-specific scoring matrix (PSSM). Apart from approaches that merely learn in the whole sequence space, multiple sequence alignment (MSA)-based methods leverage the sequences within a protein family to seize the conserved and variable regions of homologous sequences [13, 60, 70].

## Data Availability

The data of model quality assessment, protein-protein interface prediction, and ligand affinity prediction is available by https://www.atom3d.ai/. The data of protein-protein rigid-body docking can be downloaded directly from the official repository of Equidock https://github.com/octavian-ganea/equidock_public.

## Code availability

The code repository is stored at https://github.com/smiles724/bottleneck.

## Authors' contributions

F.W. and J.X. led the research. F.W. contributed technical ideas. F.W. and Y.T. developed the proposed method. F.W., D.R., and Y.T. performed the analysis. J.X. and D.R. provided evaluation and suggestions. All authors contributed to the manuscript.

## Acknowledgments

This work is supported in part by the Institute of AI Industry Research at Tsinghua University and the Molecule Mind.

## References

[1] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16 (12):1315–1322, 2019.

[2] DANIÈLE Altschuh, AM Lesk, AC Bloomer, and A Klug. Correlation of co-ordinated amino acid substitutions with function in viruses related to tobacco mosaic virus. *Journal of molecular biology*, 193(4):693–707, 1987.

[3] Afshine Amidi, Shervine Amidi, Dimitrios Vlachakis, Vasileios Megalooikonomou, Nikos Paragios, and Evangelia I Zacharaki. Enzynet: enzyme classification using 3d convolutional neural networks on spatial representation. *PeerJ*, 6:e4750, 2018.

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

[5] Kenneth Atz, Francesca Grisoni, and Gisbert Schneider. Geometric deep learning on molecular representations. *Nature Machine Intelligence*, 3 (12):1023–1032, 2021.

[6] Sarp Aykent and Tian Xia. Gbpnet: Universal geometric representation learning on protein structures. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4–14, 2022.

[7] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557): 871–876, 2021.

[8] Amos Bairoch, Rolf Apweiler, Cathy H Wu, Winona C Barker, Brigitte Boeckmann, Serenella Ferro, Elisabeth Gasteiger, Hongzhan Huang, Rodrigo Lopez, Michele Magrane, et al. The universal protein resource (uniprot). *Nucleic acids research*, 33(suppl_1):D154–D159, 2005.

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

[10] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):1–11, 2022.

[11] Tristan Bepler and Bonnie Berger. Learning protein sequence embeddings using information from structure. *arXiv preprint arXiv:1902.08661*, 2019.

[12] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[13] Surojit Biswas, Grigory Khimulya, Ethan C Alley, Kevin M Esvelt, and George M Church. Low-n protein engineering with data-efficient deep learning. *Nature methods*, 18(4):389–396, 2021.

[14] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1):i47–i56, 2005.

[15] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka, and Tom M Mitchell. Toward an architecture for never-ending language learning. In *Twenty-Fourth AAAI conference on artificial intelligence*, 2010.

[16] John-Marc Chandonia, Naomi K Fox, and Steven E Brenner. Scope: classification of large macromolecular structures in the structural classification of proteins—extended database. *Nucleic acids research*, 47(D1): D475–D481, 2019.

[17] Jianlin Cheng, Myong-Ho Choe, Arne Elofsson, Kun-Sop Han, Jie Hou, Ali HA Maghrabi, Liam J McGuffin, David Menéndez-Hurtado, Kliment Olechnovič, Torsten Schwede, et al. Estimation of model accuracy in casp13. *Proteins: Structure, Function, and Bioinformatics*, 87(12):1361–1377, 2019.

[18] Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009.

[19] UniProt Consortium. Uniprot: a hub for protein information. *Nucleic acids research*, 43(D1):D204–D212, 2015.

[20] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning based protein sequence design using proteinmpnn. *bioRxiv*, 2022.

[21] Georgy Derevyanko, Sergei Grudinin, Yoshua Bengio, and Guillaume Lamoureux. Deep convolutional networks for quality assessment of protein folds. *Bioinformatics*, 34(23):4046–4053, 2018.

[22] Kristina Djinovic-Carugo and Oliviero Carugo. Missing strings of residues in protein crystal structures. *Intrinsically disordered proteins*, 3(1):e1095697, 2015.

[23] James Dunbar, Konrad Krawczyk, Jinwoo Leem, Terry Baker, Angelika Fuchs, Guy Georges, Jiye Shi, and Charlotte M Deane. Sabdab: the structural antibody database. *Nucleic acids research*, 42(D1):D1140–D1146, 2014.

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

[25] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: towards cracking the language of life's code through self-supervised deep learning and high performance computing. *arXiv preprint arXiv:2007.06225*, 2020.

[26] Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Žídek, Russ Bates, Sam Blackwell, Jason Yim, et al. Protein complex prediction with alphafold-multimer. *BioRxiv*, pages 2021–10, 2022.

[27] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426, 2019.

[28] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[29] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using graph convolutional networks. *Advances in neural information processing systems*, 30, 2017.

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

[31] Pablo Gainza, Freyr Sverrisson, Frederico Monti, Emanuele Rodola, D Boscaini, MM Bronstein, and BE Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020.

[32] Octavian-Eugen Ganea, Xinyuan Huang, Charlotte Bunne, Yatao Bian, Regina Barzilay, Tommi Jaakkola, and Andreas Krause. Independent se (3)-equivariant models for end-to-end rigid protein docking. *arXiv preprint arXiv:2111.07786*, 2021.

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

[34] Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):1–14, 2021.

[35] Ulrike Göbel, Chris Sander, Reinhard Schneider, and Alfonso Valencia. Correlated mutations and residue contacts in proteins. *Proteins: Structure, Function, and Bioinformatics*, 18(4):309–317, 1994.

[36] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[37] Liang He, Shizhuo Zhang, Lijun Wu, Huanhuan Xia, Fusong Ju, He Zhang, Siyuan Liu, Yingce Xia, Jianwei Zhu, Pan Deng, et al. Pretraining co-evolutionary protein representation via a pairwise masked language model. *arXiv preprint arXiv:2110.15527*, 2021.

[38] Michael Heinzinger, Ahmed Elnaggar, Yu Wang, Christian Dallago, Dmitrii Nechaev, Florian Matthes, and Burkhard Rost. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC bioinformatics*, 20(1):1–17, 2019.

[39] Pedro Hermosilla and Timo Ropinski. Contrastive representation learning for 3d protein structures. *arXiv preprint arXiv:2205.15675*, 2022.

[40] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. *bioRxiv*, 2022.

[41] Ilia Igashov, Nikita Pavlichenko, and Sergei Grudinin. Spherical convolutions on molecular graphs for protein model quality assessment. *Machine Learning: Science and Technology*, 2(4):045005, 2021.

[42] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. *Advances in neural information processing systems*, 32, 2019.

[43] José Jiménez, Stefan Doerr, Gerard Martínez-Rosell, Alexander S Rose, and Gianni De Fabritiis. Deepsite: protein-binding site predictor using 3d-convolutional neural networks. *Bioinformatics*, 33(19):3036–3042, 2017.

[44] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael JL Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020.

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

[46] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

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

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

[49] Lisa N Kinch, R Dustin Schaeffer, Andriy Kryshtafovych, and Nick V Grishin. Target classification in the 14th round of the critical assessment of protein structure prediction (casp14). *Proteins: Structure, Function, and Bioinformatics*, 89(12):1618–1632, 2021.

[50] Johannes Klicpera, Shankari Giri, Johannes T Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv preprint arXiv:2011.14115*, 2020.

[51] Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.

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

[53] Jaechang Lim, Seongok Ryu, Kyubyong Park, Yo Joong Choe, Jiyeon Ham, and Woo Youn Kim. Predicting drug–target interaction using a novel graph neural network with 3d structure-embedded graph representation. *Journal of chemical information and modeling*, 59(9):3981–3988, 2019.

[54] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

[55] Yi Liu, Hao Yuan, Lei Cai, and Shuiwang Ji. Deep learning of high-order interactions for protein interface prediction. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 679–687, 2020.

[56] Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d molecular graphs. In *International Conference on Learning Representations*, 2021.

[57] Zhihai Liu, Yan Li, Li Han, Jie Li, Jie Liu, Zhixiong Zhao, Wei Nie, Yuchen Liu, and Renxiao Wang. Pdb-wide collection of binding data: current status of the pdbbind database. *Bioinformatics*, 31(3):405–412, 2015.

[58] Amy X Lu, Haoran Zhang, Marzyeh Ghassemi, and Alan Moses. Self-supervised contrastive learning of protein representations by mutual information maximization. *BioRxiv*, 2020.

[59] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

[60] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:29287–29303, 2021.

[61] Jaina Mistry, Sara Chuguransky, Lowri Williams, Matloob Qureshi, Gustavo A Salazar, Erik LL Sonnhammer, Silvio CE Tosatto, Lisanna Paladin, Shriya Raj, Lorna J Richardson, et al. Pfam: The protein families database in 2021. *Nucleic acids research*, 49(D1):D412–D419, 2021.

[62] Ryan Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Relational pooling for graph representations. In *International Conference on Machine Learning*, pages 4663–4673. PMLR, 2019.

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

[64] 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.

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

[66] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[67] Matthew Ragoza, Joshua Hochuli, Elisa Idrobo, Jocelyn Sunseri, and David Ryan Koes. Protein–ligand scoring with convolutional neural networks. *Journal of chemical information and modeling*, 57(4):942–957, 2017.

[68] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.

[69] Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners. *Biorxiv*, 2020.

[70] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In *International Conference on Machine Learning*, pages 8844–8856. PMLR, 2021.

[71] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[72] Jeffrey A Ruffolo and Jeffrey J Gray. Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies. *Biophysical Journal*, 121(3):155a–156a, 2022.

[73] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[74] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

[75] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

[76] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Protein structure prediction using multiple deep neural networks in the 13th critical assessment of protein structure prediction (casp13). *Proteins: Structure, Function, and Bioinformatics*, 87(12):1141–1148, 2019.

[77] Tom Sercu, Robert Verkuil, Joshua Meier, Brandon Amos, Zeming Lin, Caroline Chen, Jason Liu, Yann LeCun, and Alexander Rives. Neural potts model. *bioRxiv*, 2021.

[78] Jung-Eun Shin, Adam J Riesselman, Aaron W Kollasch, Conor McMahon, Elana Simon, Chris Sander, Aashish Manglik, Andrew C Kruse, and Debora S Marks. Protein design and variant prediction using

autoregressive generative models. *Nature communications*, 12(1):1–11, 2021.

[79] Vignesh Ram Somnath, Charlotte Bunne, and Andreas Krause. Multi-scale representation learning on proteins. *Advances in Neural Information Processing Systems*, 34:25244–25255, 2021.

[80] Hannes Stärk, Octavian Ganea, Lagnajit Pattanaik, Regina Barzilay, and Tommi Jaakkola. Equibind: Geometric deep learning for drug binding structure prediction. In *International Conference on Machine Learning*, pages 20503–20521. PMLR, 2022.

[81] Alexey Strokach, David Becerra, Carles Corbi-Verge, Albert Perez-Riba, and Philip M Kim. Fast and flexible protein design using deep graph neural networks. *Cell systems*, 11(4):402–411, 2020.

[82] Pascal Sturmfels, Jesse Vig, Ali Madani, and Nazneen Fatema Rajani. Profile prediction: An alignment-based pre-training task for protein sequence models. *arXiv preprint arXiv:2012.00195*, 2020.

[83] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998, 2008.

[84] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[85] Raphael Townshend, Rishi Bedi, Patricia Suriana, and Ron Dror. End-to-end learning on 3d protein structure for interface prediction. *Advances in Neural Information Processing Systems*, 32, 2019.

[86] Raphael JL Townshend, Martin Vögele, Patricia Suriana, Alexander Derry, Alexander Powers, Yianni Laloudakis, Sidhika Balachandar, Bowen Jing, Brandon Anderson, Stephan Eismann, et al. Atom3d: Tasks on molecules in three dimensions. *arXiv preprint arXiv:2012.04035*, 2020.

[87] Jeanne Trinquier, Guido Uguzzoni, Andrea Pagnani, Francesco Zamponi, and Martin Weigt. Efficient generative modeling of protein sequences using simple autoregressive models. *Nature communications*, 12(1):1–11, 2021.

[88] Karolis Uziela, David Menéndez Hurtado, Nanjiang Shu, Björn Wallner, and Arne Elofsson. Proq3d: improved model quality assessments using

deep learning. *Bioinformatics*, 33(10):1578–1580, 2017.

[89] Sameer Velankar, José M Dana, Julius Jacobsen, Glen Van Ginkel, Paul J Gane, Jie Luo, Thomas J Oldfield, Claire O'Donovan, Maria-Jesus Martin, and Gerard J Kleywegt. Sifts: structure integration with function, taxonomy and sequences resource. *Nucleic acids research*, 41 (D1):D483–D489, 2012.

[90] Jesse Vig, Ali Madani, Lav R Varshney, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. Bertology meets biology: interpreting attention in protein language models. *arXiv preprint arXiv:2006.15222*, 2020.

[91] Thom Vreven, Iain H Moal, Anna Vangone, Brian G Pierce, Panagiotis L Kastritis, Mieczyslaw Torchala, Raphael Chaleil, Brian Jiménez-García, Paul A Bates, Juan Fernandez-Recio, et al. Updates to the integrated protein–protein interaction benchmarks: docking benchmark version 5 and affinity benchmark version 2. *Journal of molecular biology*, 427(19): 3031–3041, 2015.

[92] Guoxia Wang, Xiaomin Fang, Zhihua Wu, Yiqun Liu, Yang Xue, Yingfei Xiang, Dianhai Yu, Fan Wang, and Yanjun Ma. Helixfold: An efficient implementation of alphafold2 using paddlepaddle. *arXiv preprint arXiv:2207.05477*, 2022.

[93] Renxiao Wang, Xueliang Fang, Yipin Lu, and Shaomeng Wang. The pdbbind database: Collection of binding affinities for protein- ligand complexes with known three-dimensional structures. *Journal of medicinal chemistry*, 47(12):2977–2980, 2004.

[94] Xiao Wang, Sean T Flannery, and Daisuke Kihara. Protein docking model evaluation by graph neural networks. *Frontiers in Molecular Biosciences*, page 402, 2021.

[95] Fang Wu, Qiang Zhang, Dragomir Radev, Jiyu Cui, Wen Zhang, Huabin Xing, Ningyu Zhang, and Huajun Chen. 3d-transformer: Molecular representation with transformer in 3d space. *arXiv preprint arXiv:2110.01191*, 2021.

[96] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[97] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

[98] Charles Yanofsky, Virginia Horn, and Deanna Thorpe. Protein structure relationships revealed by mutational analysis. *Science*, 146(3651):1593–1594, 1964.

[99] 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.

[100] Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. *arXiv preprint arXiv:2203.06125*, 2022.

# Appendix A Experimental Details

## A.1 Dataset Description

***Sequence position identification.***

We use a subset of 3243 high-resolution structures from the PDB and adopt a random split of 2643/330/330 for train/val/test. The distribution of the number of residues is plotted in Figure A1. For the train set, the maximum and the minimum number of residues are 6248 and 43 separately. The mean and standard deviation of the number of residues are 389.9 and 385.7. For the validation set, the maximum and the minimum number of residues are 9999 and 59 separately. The mean and standard deviation of the number of residues are 454.3 and 772.0. For the test set, the maximum and the minimum number of residues are 2326 and 59 separately. The mean and standard deviation of the number of residues are 383.2 and 286.2.

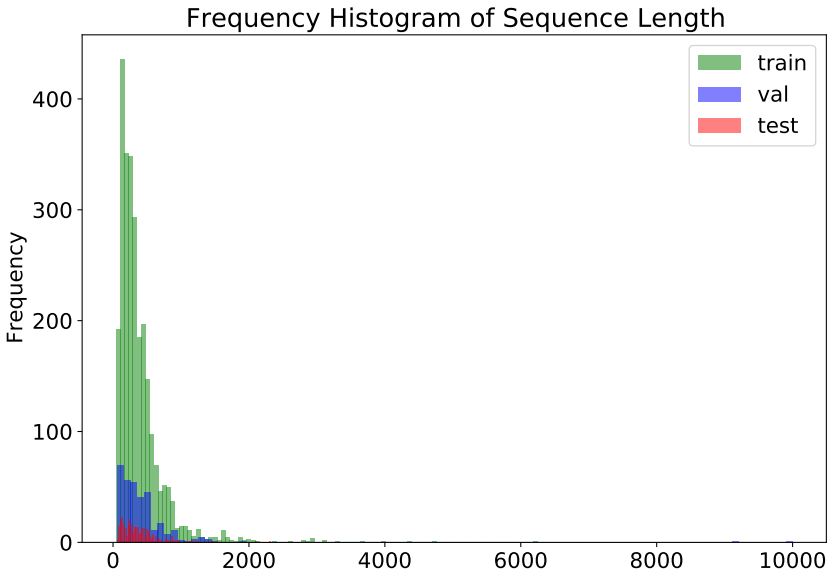


**Fig. A1**: The histogram of the sequence length

***Model quality assessment.***

The Critical Assessment of Structure Prediction (CASP) [52] is a long-running international competition held biennially, of which CASP13 is the most recent that addresses the protein structure prediction problem by withholding newly

solved experimental structures. Mirroring the setup of the competition, we follow Townshend et al. [86] and split the decoy sets based on target and released year. We choose CASP11 as test set, as the targets in CASP12-13 are not fully released yet. This leads to a dataset split of 25400/2800/16014 for train/val/test.

***Protein-protein rigid-body docking.***

We use the DB5.5 databases, which is obtained from https://zlab.umassmed.edu/benchmark/. It is randomly partitioned in train/val/test splits of sizes 203/25/25. It is worth mentioning that DB5.5 also includes unbound protein structures, however, which mostly show rigid structures.

***Protein-protein interface prediction.***

We adopt a part of the DIPS database and use a data split of 12216/1526/1526 for train/val/test. Each complex is an ensemble, with the bound ligand and bound receptor structures forming 2 distinct sub-units of said ensemble. We then define the neighboring amino acids as those with any $\alpha$-carbon within 8Å of one another. These neighbors are then included as the positive samples, with all other residues as negatives. At prediction time, we attempt to re-predict which possible residues are positive or negative. In other words, we desire to determine whether each residue is located at the binding pocket. AUROC of those predictions are used as the metric to evaluate performance.

***Ligand affinity prediction.***

PDBbind contains X-ray structures of proteins bound to small molecules and peptide ligands, We use the dataset mined from PBDbind by Townshend et al. [86], which has two splits based on 30% and 60% sequence identity thresholds, respectively. Splitting using 30% sequence identity results in train/-val/test of 3507/466/490, while splitting using 60% sequence identity results in train/val/test of 3678/460/460.

## A.2 Backbone Architecture

For tasks that only require the predictive model to output a scalar for the protein/complex or each residue including model quality assessment, protein-protein interface prediction, and ligand binding affinity prediction, we select GVP-GNN, EGNN, and Molformer as the backbone architecture. For more complicated tasks that require more complex computational processes such as protein-protein rigid-body docking, we use specific models to address them like Equidock.

***GVP-GNN.***

GVP-GNN [44, 45] is an equivariant GNN in which all node and edge embeddings are tuples $(\mathbf{s}, \mathbf{V})$ of scalar feature and geometric vector features.

Message and update functions are parameterized by *geometric vector perceptrons* (GVPs) – modules mapping between tuples $(\mathbf{s}, \mathbf{V})$ while preserving rotation equivariance. Its computational process is described in Algorithm 1, where $\mathbf{s}$ and $\mathbf{V}$ correspond to the node embedding $\mathbf{h}$ and coordinates $\mathbf{x}$ separately. :

**Algorithm 1** GVP-GNN

**Input:** Scalar and vector features $(\mathbf{s}, \mathbf{V}) \in \mathbb{R}^n \times \mathbb{R}^{\nu \times 3}$.
**Output:** Scalar and vector features $(\mathbf{s}', \mathbf{V}') \in \mathbb{R}^n \times \mathbb{R}^{\mu \times 3}$.
$h \leftarrow \max(\nu, \mu)$
$\mathbf{V}_h \leftarrow \mathbf{W}_h \mathbf{V} \quad \in \mathbb{R}^{h \times 3}$
$\mathrm{V}_\mu \leftarrow \mathbf{W}_\mu \mathbf{V}_h \in \mathbb{R}^{\mu \times 3}$
$\mathrm{s}_h \leftarrow \|\mathrm{V}_h\|_2$ (row-wise) $\in \mathbb{R}^h$
$\mathrm{v}_\mu \leftarrow \|\mathrm{V}_\mu\|_2$ ( row-wise) $\in \mathbb{R}^\mu$
$\mathrm{s}_{h+n} \leftarrow$ concat $(\mathrm{s}_h, \mathbf{s}) \in \mathbb{R}^{h+n}$
$\mathrm{s}_m \leftarrow \mathbf{W}_m \, \mathrm{s}_{h+n} + \mathbf{b} \in \mathbb{R}^m$
$\mathbf{s}' \leftarrow \sigma(\mathrm{s}_m) \in \mathbb{R}^m$
$\mathbf{V}' \leftarrow \sigma^+(\mathbf{v}_\mu) \odot \mathbf{V}_\mu$ (row-wise multiplication) $\in \mathbb{R}^{\mu \times 3}$
**Return:** $(\mathbf{s}', \mathbf{V}')$

At its core, GVP-GNN consists of two separate linear transformations $\mathbf{W}_m$ and $\mathbf{W}_h$ for the scalar and vector features, followed by nonlinearities $\sigma, \sigma^+$. An additional linear transformation $\mathbf{W}_\mu$ is inserted before the vector nonlinearity to control the output dimensionality independently of the number of norms extracted. We adopt a 5-layer GVP-GNN with a dropout rate of 0.7 and a ReLU activation function. The number of radial bases in the edge embedding is 16 and the node dimension is set as (100, 16). All implementation codes are downloaded from the official repository in https://github.com/drorlab/gvp.

***EGNN.***

EGNN [73] achieves equivariance without expensive high-order representations in intermediate layers and also realizes competitive performance. Its Equivariant Graph Convolutional Layer (EGCL) takes the set of node embedding $\mathbf{h}^l = \{\mathbf{h}_i^l\}_{i=1,\ldots,N}$, the coordinate embeddings $\mathbf{x}^l = \{\mathbf{x}_i^l\}_{i=1,\ldots,N}$ and edge information $\mathcal{E} = (e_{ij})$ as input, and then outputs a transformation on $\mathbf{h}^{l+1}$ and

$\mathbf{x}^{l+1}$. Concisely, The equations that define this layer are described as follows:

$$\begin{aligned}
\mathbf{m}_{ij} &= \phi_e\left(\mathbf{h}_i^l, \mathbf{h}_j^l, \left\|\mathbf{x}_i^l - \mathbf{x}_j^l\right\|^2, a_{ij}\right), \\
\mathbf{x}_i^{l+1} &= \mathbf{x}_i^l + C\sum_{j\neq i}\left(\mathbf{x}_i^l - \mathbf{x}_j^l\right)\phi_x\left(\mathbf{m}_{ij}\right), \\
\mathbf{m}_i &= \sum_{j\neq i}\mathbf{m}_{ij}, \\
\mathbf{h}_i^{l+1} &= \phi_h\left(\mathbf{h}_i^l, \mathbf{m}_i\right),
\end{aligned} \tag{A1}$$

where $\mathbf{h}_i^l \in \mathbb{R}^{nf}$ is the nf_dimensional embedding of node $v_i$ at layer $l$. $a_{ij}$ are the edge attributes. $\phi_e$ and $\phi_h$ are the edge and node operations respectively which are commonly approximated by Multi-layer Perceptrons (MLPs). $\phi_x : \mathbb{R}^{nf} \to \mathbb{R}^1$ is the function that takes the edge embedding $\mathbf{m}_{ij}$ as input from the previous edge operation and outputs a scalar value. $C$ is chosen to be $1/\|\mathcal{N}_i\|$, which divides the sum by its number of neighboring (connected) elements. We choose a 4-layer EGNN with a Siwsh activation function as a non-linearity. The number of the dimension for edges is 16 and residue connections are used. All implementation codes are downloaded from the official repository in https://github.com/vgsatorras/egnn.

***Molformer.***

Molformer [95] is a variant of Transformer that employs a heterogeneous self-attention layer to differentiate the interactions between multi-level nodes. Here we use a weaker version of Molformer. To be explicit, we do not extract any sort of motifs from either protein or small molecules. Besides, we abandon the multi-scale self-attention mechanism and the Attentive Farthest Point Sampling (AFPS) for more efficient computations and only use the global features. Even though we pick up a simplified form of Molformer, its performance is competitive with or even outperforms EGNN and GVP-GNN on all tasks. The Molformer architecture has 2 layers, 4 heads, a hidden-layer dimension of 1280, and a dropout rate of 0.1. All implementation codes are downloaded from the official repository in https://github.com/smiles724/molformer.

***EquiDock.***

Equidock [32] predicts the rotation and translation to place on of the proteins at the right docked position relative to the second protein. It adopts an Independent E(3)-Equivariant Graph Matching Network (IEGMN), which extends both Graph Matching Networks (GMN) and EGNN. It performs node coordinate and feature embedding updates for an input pair of protein graphs and uses inter- and inter-node messages, as well as E(3)-equivariant coordinate updates. The backbone IEGMN has 5 layers and no dropout. It uses LeakyReLU as the activation function. and does not use distance as an edge feature. All implementation codes are downloaded from the official repository in https://github.com/octavian-ganea/equidock_public.

### A.3 Training Details

We run all experiments on 2 A100 GPUs, each with a memory of 80G. For MQA, PPRD and PPI, we use residue-level graphs for protein representation learning. For PPRD, models are trained using Adam with a learning rate of 2e-4 and early stopping with patience of 30 epochs. For MQA, PPI, and LBA, models are trained using Adam with a learning rate of 1e-4 and early stopping with patience of 8 epochs. A Plateau learning rate scheduler is applied with a factor of 0.6, patience of 5, and a minimum learning rate of 5e-7. The batch size is 32 if no out-of-memory (OOM) error is not triggered, otherwise, we adopt a batch size of 16. The maximum epoch is set as 200.

For PPRD, we randomly assign the roles of ligands and receptors during training. For PPI, as mentioned before, we formulate interface as the residues whose least distances to their counterpart protein are within 8 Å. For LBA, we only use the residues within a distance of 6 Å from the ligand (*i.e.*, the pocket) following Townshend et al. [86]. We build heterogeneous molecular graphs where nodes for proteins are residue-level and nodes for ligands are atom-level. Here we do not distinguish different atom types and simply regard all atoms as the same group, which is denoted as the new 'LIG' pseudo residue class.

For the protein language model, we use the ESM-2 with a parameter size of 650M and 33 layers as the default one. It is trained on UR50/D2021_04 and has an embedding dimension of 1280. We extract per-residue representations as the input for each task. For the ablation study, we adopt ESM-2 with parameter sizes of 8M, 35M, 150M, and 3B that are all trained on the same UR50/D2021_04 dataset with different layers of 6, 12, 30, and 36 respectively. For more details, please visit the official website of ESM in https://github.com/facebookresearch/esm.

## Appendix B Limitations

In spite of our successful confirmation that protein language models can promote geometric deep learning, there are several limitations and extensions of our framework left open for future investigation. First, our work only examines the efficacy of the state-of-the-art protein language model, ESM-2. It should be more convincing if more types of protein language models like MSA-Transformer and ProtTrans are verified. Second, our 3D protein graphs are residue-level. We believe atom-level protein graphs also benefit from our approach, but its increase in performance needs further exploration.