EURNET: EFFICIENT MULTI-RANGE RELATIONAL MODELING OF PROTEIN STRUCTURE

Minghao Xu[1,4], Yuanfan Guo[3], Yi Xu[3],
Jian Tang[1,5,6], Xinlei Chen[2], Yuandong Tian[2]
Mila - Québec AI Institute[1], Meta AI (FAIR)[2], Shanghai Jiao Tong University[3]
Université de Montréal[4], HEC Montréal[5], CIFAR AI Chair[6]
minghao.xu@mila.quebec, {gyfastas,xuyi}@sjtu.edu.cn
jian.tang@hec.ca, {xinleic,yuandong}@meta.com

ABSTRACT

Modeling the 3D structures of proteins is critical for obtaining effective protein structure representations, which further boosts protein function understanding. Existing protein structure encoders mainly focus on modeling short-range interactions within protein structures, while they neglect modeling the interactions at multiple length scales that are actually complete interactive patterns in protein structures. To attain complete interaction modeling with efficient computation, we introduce the **EurNet** for **E**fficient m**u**lti-range **r**elational modeling. In EurNet, we represent the protein structure as a multi-relational residue-level graph with different types of edges for modeling short-range, medium-range and long-range interactions. To efficiently process these different interactive relations, we propose a novel modeling layer, called **G**ated **R**elational **M**essage **P**assing (**GRMP**), as the basic building block of EurNet. GRMP can capture multiple interactive relations in protein structures with little extra computational cost. We verify the state-of-the-art performance of EurNet on EC and GO protein function prediction benchmarks, and the proposed GRMP layer is proved to achieve better efficiency-performance trade-off than the widely-used relational graph convolution.

1 INTRODUCTION

Proteins play important roles in governing various biological processes and life itself, boosting a broad range of applications in drug discovery (Teague, 2003) and healthcare (Organization & University, 2007). Proteins are composed of a chain of amino acids (*a.k.a.*, residues) that fold into specific 3D conformations, and these 3D structures further determine diverse functions of proteins. Machine learning models (Hermosilla et al., 2021; Jing et al., 2021; Zhang et al., 2022) have shown great promise in extracting informative representations from protein structures, which further boost protein function prediction (Gligorijević et al., 2021) and protein design (Hsu et al., 2022).

Existing protein structure encoders (Hermosilla et al., 2021; Jing et al., 2021; Zhang et al., 2022) mainly focus on capturing the short-range interactions within protein structures (*e.g.*, peptide and hydrogen bonds), and some impressive performance on function prediction (Gligorijević et al., 2021; Zhang et al., 2022) is gained based on such modeling manner. However, it is worth noticing that residues in protein structures can interact beyond short range towards medium and long ranges by non-bond interactions like hydrophobic interaction. These non-local interactions are not explicitly modeled by previous methods, which limits their effectiveness. This limitation motivates us to study protein structure modeling that captures the interactions at multiple spatial ranges.

To attain this goal, we propose a novel protein structure encoder called **EurNet** for **E**fficient m**u**lti-range **r**elational modeling. Specifically, EurNet represents the protein structure as a multi-relational residue-level graph in which different types of edges are constructed to separately model short-range, medium-range and long-range interactions in the whole structure. To efficiently process these various interactive relations, we introduce the **G**ated **R**elational **M**essage **P**assing (**GRMP**) layer as EurNet's basic building block. GRMP separately performs (1) relational message aggregation on each individual feature channel and (2) node-wise aggregation of different feature channels.

Compared to the classical relational graph convolution (RGConv) (Schlichtkrull et al., 2018), GRMP enjoys lower computational cost when more relations are to be modeled, and thus can better perform multi-range relational modeling of protein structures given the same computational budget.

We study the effectiveness and efficiency of EurNet on EC and GO protein function prediction benchmarks. Under the fair comparison with comparable model size, EurNet consistently outperforms the previous SOTA GearNet with a clear margin in terms of protein-centric maximum F-score (EC: 0.768 *v.s.* 0.730; GO-BP: 0.437 *v.s.* 0.356; GO-MF: 0.563 *v.s.* 0.503; GO-CC: 0.421 *v.s.* 0.414). These performance improvements remain when edge-level message passing is involved. In addition, we demonstrate the superior efficiency-performance trade-off of the proposed GRMP layer over the broadly-used relational graph convolution, when they serve as the building block for multi-relational modeling of protein structures.

2 Related Work

Protein structure modeling. Under the guidance of the principle that "protein structures are determinants of their functions", a variety of protein structure encoders have been developed to acquire informative protein representations on different structural granularities, including residue-level structures (Gligorijević et al., 2021; Zhang et al., 2022), atom-level structures (Jing et al., 2021; Hermosilla et al., 2021) and protein surfaces (Gainza et al., 2020; Sverrisson et al., 2021). This work focuses on the residue-level protein structure modeling. GearNet (Zhang et al., 2022) relates to our work by exploring multi-relational modeling of residue-level structures with short-range linking and relational graph convolution (RGConv). By comparison, the proposed EurNet models a broader range of interactions including short, medium and long ranges, and it studies the gated relational message passing (GRMP) as a more efficient and equally effective alternative of RGConv.

Multi-relational data modeling. Multi-relational data are ubiquitous in the real world, *e.g.*, knowledge graphs (Toutanova & Chen, 2015), customer-product networks (Li et al., 2014) and proteins (Zhang et al., 2022). To effectively model multiple types of relations/interactions, existing works have explored embedding-based methods (Bordes et al., 2013; Sun et al., 2019), multi-headed attention (Vaswani et al., 2017) and different relational graph neural networks (GNNs) (Schlichtkrull et al., 2018; Vashishth et al., 2019; Busbridge et al., 2019; Zhu et al., 2021). Previous relational GNNs mainly focus on the model expressivity, and few works (Li et al., 2021) study the computational efficiency for multi-relational modeling at scale. EurNet is designed to model the multi-relational interactions within protein structures in a computationally efficient way.

3 Method

3.1 Problem Definition

We consider the alpha carbon (*i.e.*, Cα) graph as the representation of protein structure, which is an informative and light-weight summary of the overall protein 3D structure and is widely used in the literature (Gligorijević et al., 2021; Baldassarre et al., 2021; Zhang et al., 2022). To be specific, we extract all Cαs of a protein as the node set $\mathcal{V}$ of our graph. Based on these nodes, we construct three groups of edges $\mathcal{E}_{\text{short}} = \{\{(u, v, r)\}|r \in \mathcal{R}_{\text{short}}\}$, $\mathcal{E}_{\text{medium}} = \{\{(u, v, r)\}|r \in \mathcal{R}_{\text{medium}}\}$ and $\mathcal{E}_{\text{long}} = \{\{(u, v, r)\}|r \in \mathcal{R}_{\text{long}}\}$ to represent short-, medium- and long-range interactions in the protein structure, where (u, v, r) denotes an edge from node u to node v with relation r, and $\mathcal{R}_{\text{short}}$/$\mathcal{R}_{\text{medium}}$/$\mathcal{R}_{\text{long}}$ is the set of relations for short-/medium-/long-range interactions. To capture the interactions on different spatial ranges, all these edges are gathered into the edge set $\mathcal{E} = \mathcal{E}_{\text{short}} \cup \mathcal{E}_{\text{medium}} \cup \mathcal{E}_{\text{long}} = \{\{(u, v, r)\}|r \in \mathcal{R}\}$ with the integrated relation set $\mathcal{R} = \mathcal{R}_{\text{short}} \cup \mathcal{R}_{\text{medium}} \cup \mathcal{R}_{\text{long}}$. In this way, the protein structure is represented as a multi-relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ that is aware of diverse types of interactions within the structure.

3.2 Relational Edge Construction for Short-, Medium- and Long-range Interactions

In this section, we elucidate the construction scheme of multi-range relational edges. See Fig. 1 for a graphical illustration.

Relational edges for short-range interactions ($|\mathcal{R}_{\text{short}}| = 6$). We construct short-range edges inspired by two kinds of bond interactions in the short range, *i.e.*, peptide and hydrogen bonds.

- *Peptide bonds* link the residues that are consecutive on protein sequence, which motivates us to connect residues close on sequence. In specific, we build *sequential edges* to connect the Cα nodes within the distance of 2 on the protein sequence, where each of the sequential distances {-2,-1,0,1,2} is regarded as a single relation (*i.e.*, 5 relations in total).
- *Hydrogen bonds* are formed between spatially close residues, holding the secondary structures of proteins. To capture hydrogen bonds with different lengths, based on the length distribution (Egli & Sarkhel, 2007), we construct *spatial edges* to connect the Cα nodes within the Euclidean distance of 10 angstroms. All spatial edges own the same relation.

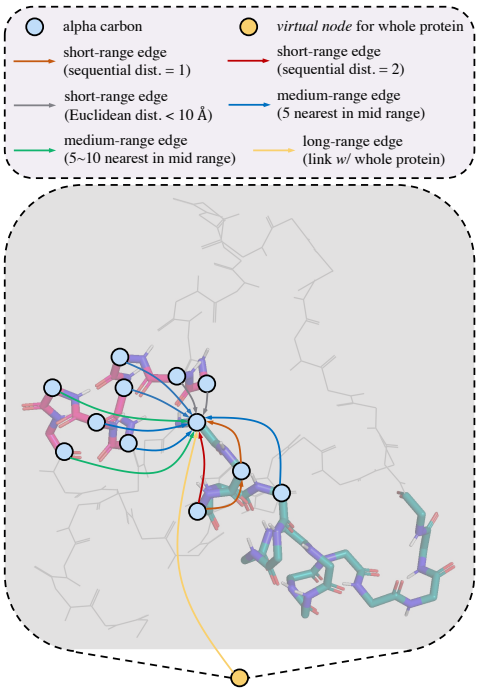

Figure 1: Illustration of multi-range relational edges. *Abbr.*, dist.: distance.

Relational edges for medium-range interactions ($|\mathcal{R}_{\text{medium}}| = 2$). Medium-range interactions are performed between the residues that are not close both on sequence and structure (Gromiha & Selvaraj, 2004). According to the statistics (Gromiha & Selvaraj, 2004), for each Cα node, we first filter out all its neighbors within the sequential distance of 5 or within the Euclidean distance of 10 angstroms. We then connect it with the remaining nodes that are *5 nearest* and *5∼10 nearest* to it (measured by Euclidean distance), and the connections with these two sets of medium-range neighbors are regarded as two different relations for strong and weak medium-range interactions, respectively.

Relational edges for long-range interactions ($|\mathcal{R}_{\text{long}}| = 1$). Inspired by the long-range interaction modeling of small molecules (Gilmer et al., 2017; Ying et al., 2021), we introduce the *virtual node* to model long-range interactions within protein structures, which guarantees that all pairs of residues are within 2 hops in our constructed graph. Specifically, we employ a virtual node as the whole protein representation and link it to all Cα nodes with a single relation. These edges make each residue aware of the status of all other residues, and thus the long-range interactions beyond short and medium ranges can be captured.

We gather all these edges with 9 different interactive relations into the edge set $\mathcal{E}$ and the relation set $\mathcal{R}$, which, together with the Cα node set $\mathcal{V}$, derive the full graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ for multi-range relational protein structure modeling.

3.3 Gated Relational Message Passing

Relational Graph Convolution (RGConv) (Schlichtkrull et al., 2018) is a typical method to model the constructed multi-relational graph $\mathcal{G}$. It employs a unique convolutional kernel matrix W_r to aggregate the messages of relation r, leading to $|\mathcal{R}|$ different kernel matrices in total for neighborhood aggregation. The RGConv layer updates the representation of node v from z_v to z'_v as below:

$$z_v^{\text{aggr}} = \sum_{r\in\mathcal{R}} \sum_{u\in\mathcal{N}_r(v)} \frac{1}{|\mathcal{N}_r(v)|} W_r z_u, \quad z'_v = W^{\text{self}} z_v + z_v^{\text{aggr}}, \tag{1}$$

where z_v^{aggr} is the aggregated message for node v, $\mathcal{N}_r(v) = \{u|(u, v, r) \in \mathcal{E}\}$ are v's neighbors with relation r, and W^{self} is the weight matrix for self-update (we omit all bias terms for brevity).

We assume that, when introducing a new relation, the in-degree of each node will increase by $\bar{d}$ on average. By taking the efficient implementation of RGConv with sparse matrix multiplication, it can be shown that the floating-point operations (FLOPs) of RGConv with C-dimensional input and output node features has the following form (see Appendix A for proof):

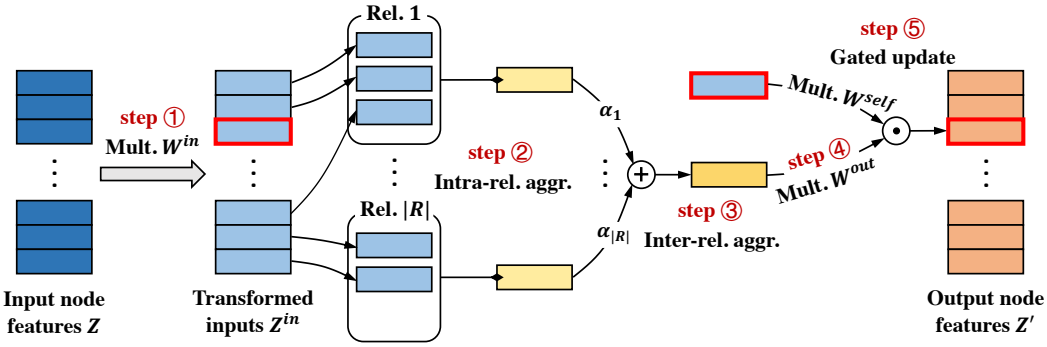

Figure 2: Graphical illustration for node representation update in the GRMP layer. We specifically show the neighborhood aggregation and representation update procedure of the node denoted in red. *Abbr.*, Multi.: multiply with; Rel.: relation; aggr.: aggregation.

$$\text{FLOPs}(\text{RGConv}) = |\mathcal{R}| \cdot (2\bar{d}|\mathcal{V}|C + 2|\mathcal{V}|C^2) + 2|\mathcal{V}|C^2 + |\mathcal{V}|C. \tag{2}$$

Therefore, the computational cost will scale with the relation number $|\mathcal{R}|$ by the factor of $2\bar{d}|\mathcal{V}|C + 2|\mathcal{V}|C^2$. Considering both the node (*i.e.*, residue) number $|\mathcal{V}|$ and the feature dimension C could be large when modeling large protein structures with large models, the $2|\mathcal{V}|C^2$ term will be the main obstacle of exploring more relations with moderate extra computation.

For more efficient multi-relational modeling, we aim at an approach that (1) can effectively model the interactions among relational messages and among feature channels, and (2) owns a gentle scaling behavior when modeling increasing number of relations within protein structures. To attain this goal, we propose the Gated Relational Message Passing (GRMP) layer. Inspired by light-weight separable graph convolution methods (Balcilar et al., 2020; Li et al., 2021) that aggregate neighborhood features in a channel-wise way, GRMP decomposes the relation-channel entangled aggregation of RGConv into (*i*) the aggregation of intra- and inter-relation messages on each individual channel and (*ii*) the aggregation of different feature channels. Specifically, it consecutively performs following steps: ① a pre-layer node-wise channel aggregation with the weight matrix W^{in}, ② an intra-relation message aggregation through channel-wise graph convolution, ③ an inter-relation message aggregation by node-adaptive weighted summation, ④ a post-layer node-wise channel aggregation with the weight matrix W^{out}, and ⑤ the final node representation update by *regarding the aggregated neighborhood information as gate*. Formally, GRMP updates the representation of node v from z_v to z'_v as below (see Fig. 2 for a graphical illustration):

$$z_v^{\text{aggr}} = \overbrace{W^{\text{out}}\Bigg(\overbrace{\sum_{r\in\mathcal{R}} \alpha_r(v) \cdot \underbrace{\sum_{u\in\mathcal{N}_r(v)} \frac{1}{|\mathcal{N}_r(v)|} w_r \odot \underbrace{(W^{\text{in}} z_u)}_{\text{step ①}}}_{\text{step ②}}}^{\text{step ③}}\Bigg)}^{\text{step ④}}, \quad z'_v = \underbrace{W^{\text{self}} z_v \odot z_v^{\text{aggr}}}_{\text{step ⑤}}, \tag{3}$$

where $\alpha(v) = W^{\alpha} z_v \in \mathbb{R}^{|\mathcal{R}|}$ are the attentive weights assigned to all relations on node v (W^{α} is the weight matrix for node-adaptive relation weighting), w_r is the channel-wise convolutional kernel vector for relation r (with the same shape as the node feature vector after step ①), and $\odot$ denotes the Hadamard product. The definitions of z_v^{aggr}, $\mathcal{N}_r(v)$ and W^{self} follow Eq. (1), and all bias terms are omitted for brevity.

Under the efficient implementation with sparse matrix multiplication, GRMP consumes the FLOPs as below when taking C-dimensional input and output node features (see Appendix A for proof):

$$\text{FLOPs}(\text{GRMP}) = |\mathcal{R}| \cdot (2\bar{d} + 7)|\mathcal{V}|C + 6|\mathcal{V}|C^2. \tag{4}$$

Therefore, the relation number $|\mathcal{R}|$ scales the computational cost of GRMP with the scaling factor $(2\bar{d} + 7)|\mathcal{V}|C$. Compared to the scaling factor $2\bar{d}|\mathcal{V}|C + 2|\mathcal{V}|C^2$ of RGConv, this factor gets rid of the quadratic reliance on feature dimension and thus leads to a gentler scaling behavior when modeling increasing number of relations. This merit enables GRMP-based models to achieve superior efficiency-performance trade-off against RGConv-based models, as studied in Sec. 4.3.

Table 1: Benchmark results on EC and GO protein function prediction.

Model	EC		GO-BP		GO-MF		GO-CC	
	F_{max}	AUPR	F_{max}	AUPR	F_{max}	AUPR	F_{max}	AUPR
CNN	0.545	0.526	0.244	0.159	0.354	0.351	0.287	0.204
3DCNN_MQA	0.077	0.029	0.240	0.132	0.147	0.075	0.305	0.144
GCN	0.320	0.319	0.252	0.136	0.195	0.147	0.329	0.175
GAT	0.368	0.320	0.284	0.171	0.317	0.329	0.385	0.249
GVP	0.489	0.482	0.326	**0.224**	0.426	0.458	0.420	**0.279**
GraphQA	0.509	0.543	0.308	0.199	0.329	0.347	0.413	0.265
GearNet	0.730	0.751	0.356	0.211	0.503	0.490	0.414	**0.276**
EurNet	**0.768**	**0.756**	**0.437**	**0.223**	**0.563**	**0.499**	**0.421**	0.267
GearNet-Edge	0.810	0.872	0.403	0.251	0.580	0.570	0.450	**0.303**
EurNet-Edge	**0.829**	**0.876**	**0.456**	**0.262**	**0.592**	**0.576**	**0.453**	0.298

3.4 Architecture of EurNet

EurNet follows the general architecture of GearNet (Zhang et al., 2022) to have a fair comparison with this SOTA model. To be specific, based on the input node features (*i.e.*, the one-hot encoding of amino acid types), six GRMP layers with 512 hidden dimensions are stacked for multi-range relational modeling. After each layer, the sum pooling over all Cα representations is deemed as the whole-protein representation, and these per-layer protein representations are concatenated to produce the final output. To perform a downstream task, a task-specific prediction head is appended.

Note that, in EurNet, all graph construction and message passing operations rely only on the quantities (*e.g.*, sequential and Euclidean distance) that are invariant to translation, rotation and reflection. Therefore, EurNet satisfies **E(3)-invariance** (Mumford et al., 1994).

4 Experiments

4.1 Experimental Setups

Datasets. Two standard protein function prediction benchmarks are used in our experiments:

- **Enzyme Commission (EC) number prediction** (Gligorijević et al., 2021) requires the model to predict the EC numbers of a protein based on its tertiary structure, where the EC numbers describe the protein's catalysis of biochemical reactions. This task involves the binary prediction of 538 different EC numbers, forming 538 binary classification problems.
- **Gene Ontology (GO) term prediction** (Gligorijević et al., 2021) seeks to predict the GO terms owning by a protein based on its tertiary structure. This benchmark is further split into three branches based on three types of ontologies: biological process (BP), molecular function (MF) and cellular component (CC). Each branch is formed by multiple binary classification problems.

Training and evaluation. An AdamW (Loshchilov & Hutter, 2017) optimizer (betas: $[0.9, 0.999]$, weight decay: 0) is used to train the model for 200 epochs with the binary cross entropy loss. We adopt a cosine learning rate scheduler to linearly increase the learning rate from 1.0×10^{-7} to 1.0×10^{-4}, and the learning rate is decayed to 1.0×10^{-6} in the rest epochs with a cosine rate. All models are trained with batch size 16 on 4 Tesla-V100-32GB GPUs (*i.e.*, four proteins per GPU).

For all models on all tasks, we select the checkpoint for evaluation based on the validation set performance, and we report all results on the seed 1024 following Zhang et al. (2022). Following the original benchmark (Gligorijević et al., 2021), we use two evaluation metrics, the protein-centric maximum F-score, *i.e.*, F_{max}, and the pair-centric area under precision-recall curve, *i.e.*, AUPR.

4.2 Benchmark Results

Baselines. We compare with the SOTA GearNet (Zhang et al., 2017) under two settings, *i.e.*, with and without edge message passing ("-Edge" in Tab. 1). More details about edge message passing are provided in Appendix B. We also include other baselines, *i.e.*, CNN (Shanehsazzadeh et al., 2020), 3DCNN_MQA (Derevyanko et al., 2018), GCN (Kipf & Welling, 2016), GAT (Veličković et al., 2017), GVP (Jing et al., 2021) and GraphQA (Baldassarre et al., 2021), for complete comparisons.

Table 2: Efficiency-Performance trade-off comparison between GRMP and RGConv. *Throughput* measures the number of proteins that the model can process in one second. *Abbr.*, dim.: dimension.

Layer	Hidden dim.	EC		GO-BP		GO-MF		GO-CC	
		Throughput	AUPR	Throughput	AUPR	Throughput	AUPR	Throughput	AUPR
RGConv	422	34.4	0.740	32.8	0.209	32.6	0.478	32.8	0.260
GRMP	512	34.6	**0.756**	32.9	**0.223**	32.7	**0.499**	33.0	**0.267**
RGConv	512	31.2	0.753	29.7	0.221	29.6	0.493	29.6	0.270
GRMP	592	31.5	**0.770**	30.1	**0.233**	29.9	**0.508**	30.0	**0.279**

Results. All benchmark results are reported in Tab. 1. For the F_{max} metric, EurNet consistently outperforms GearNet with a clear margin on all four tasks, and the performance gains remain after involving edge message passing. For the AUPR metric, EurNet outperforms GearNet on 6 out of 8 benchmark settings. Since EurNet follows the general architecture of GearNet for fair comparison, we can conclude the effectiveness of *medium- and long-range interaction modeling* and *GRMP-based multi-relational modeling*, which are novel modeling mechanisms in EurNet.

4.3 Study of Efficiency-Performance Trade-off

Setups. In this experiment, we compare the efficiency-performance trade-off of the proposed GRMP layer with that of the RGConv layer. We employ the *throughput* metric (*i.e.*, the number of proteins that the model can process in one second) to measure the *efficiency*, and use the *AUPR* metric to measure the *performance*. GRMP and RGConv layers are respectively embedded into the EurNet architecture so as to have a fair comparison.

Results. In Tab. 2, we compare between GRMP and RGConv under the comparable throughput. (1) We first set the hidden dimension of GRMP as 512. Under the comparable throughput, RGConv can only have the dimension of 422, and its AUPR scores are clearly lower than those of GRMP on all four tasks. (2) We then increase RGConv's hidden dimension to 512. At this time, RGConv can achieve comparable AUPR scores against GRMP under the same hidden dimension, while the throughput of RGConv is decreased by 3.0$\sim$3.2 across four tasks. Under the comparable throughput, GRMP can have the hidden dimension of 592, which leads to clearly higher AUPR scores on all four tasks. These results demonstrate that GRMP owns a better efficiency-performance trade-off than RGConv in terms of protein structure modeling.

4.4 Ablation Study

Table 3: Ablation study of multi-range edges on EC.

short	medium	long	F_{max}	AUPR
✓			**0.750**	**0.730**
	✓		0.708	0.688
		✓	0.647	0.615
✓	✓		0.755	0.739
✓		✓	**0.760**	**0.747**
	✓	✓	0.720	0.699
✓	✓	✓	**0.768**	**0.756**

Tab. 3 shows the performance of EurNet on EC by using different ranges of edges. When a single range of edges are employed, the model with short-range edges obtains the highest performance. This result illustrates the importance of capturing short-range interactions for protein structure modeling. By adding medium-range or long-range edges, both F_{max} and AUPR scores are improved, where the extra modeling of non-local interactions (*e.g.*, hydrophobic interactions) contributes to this improvement. By using all three ranges of edges, the full model of EurNet achieves the best F_{max} and AUPR scores, which demonstrates the necessity of modeling each range of interactions.

5 Conclusions and Future Work

This work proposes the EurNet to model residue-level protein structures. EurNet constructs relational edges on short, medium and long ranges to capture the interactions on multiple length scales. The novel gated relational message passing (GRMP) layer serves as the basic building block of EurNet to efficiently perform multi-relational protein structure modeling. We verify the effectiveness and efficiency of EurNet on EC and GO protein function prediction benchmarks.

In future works, we will extend EurNet to model atom-level protein structures and protein surfaces, and we will also explore protein structure pre-training with EurNet.

REFERENCES

Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. Spectral-designed depthwise separable graph neural networks. In *ICML Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020.

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.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in Neural Information Processing Systems*, 2013.

Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y Hammerla. Relational graph attention networks. *arXiv preprint arXiv:1904.05811*, 2019.

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

Martin Egli and Sanjay Sarkhel. Lone pair-aromatic interactions: To stabilize or not to stabilize. *Accounts of chemical research*, 40(3):197–205, 2007.

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.

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*, pp. 1263–1272. PMLR, 2017.

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.

M Michael Gromiha and S Selvaraj. Inter-residue interactions in protein folding and stability. *Progress in biophysics and molecular biology*, 86(2):235–277, 2004.

Frank Harary and Robert Z Norman. Some properties of line digraphs. *Rendiconti del circolo matematico di palermo*, 9(2):161–168, 1960.

Pedro Hermosilla, Marco Schäfer, Matěj Lang, Gloria Fackelmann, Pere Pau Vázquez, Barbora Kozlíková, Michael Krone, Tobias Ritschel, and Timo Ropinski. Intrinsic-extrinsic convolution and pooling for learning on 3d protein structures. *International Conference on Learning Representations*, 2021.

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.

Bowen Jing, Stephan Eismann, Pratham N. Soni, and Ron O. Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=1YLJDvSx6J4`.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Jing Li, Lingling Zhang, Fan Meng, and Fenhua Li. Recommendation algorithm based on link prediction and domain knowledge in retail transactions. *Procedia Computer Science*, 31:875–881, 2014.

Qimai Li, Xiaotong Zhang, Han Liu, Quanyu Dai, and Xiao-Ming Wu. Dimensionwise separable 2-d graph convolution for unsupervised and semi-supervised learning on graphs. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

David Mumford, John Fogarty, and Frances Kirwan. *Geometric invariant theory*, volume 34. Springer Science & Business Media, 1994.

World Health Organization and United Nations University. *Protein and amino acid requirements in human nutrition*, volume 935. World Health Organization, 2007.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, 2018.

Amir Shanehsazzadeh, David Belanger, and David Dohan. Is transfer learning necessary for protein landscape prediction? *arXiv preprint arXiv:2011.03443*, 2020.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.

Freyr Sverrisson, Jean Feydy, Bruno E Correia, and Michael M Bronstein. Fast end-to-end learning on protein surfaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15272–15281, 2021.

Simon J Teague. Implications of protein flexibility for drug discovery. *Nature reviews Drug discovery*, 2(7):527–541, 2003.

Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, pp. 57–66, 2015.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

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.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems*, 2021.

A FLOPS OF RGCONV AND GRMP

For FLOPs computation, we consider the multi-relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ with node set $\mathcal{V}$, edge set $\mathcal{E}$ and relation (*i.e.*, edge type) set $\mathcal{R}$, and both input and output node features are with C feature channels. In addition, we assume that, when introducing a new relation, the in-degree of each node will increase by $\bar{d}$ on average.

Proposition 1. *To process the assumed multi-relational graph, the Relational Graph Convolution (RGConv) consumes the FLOPs as below under the efficient implementation with sparse matrix multiplication:*

$$\mathrm{FLOPs}(\mathrm{RGConv}) = |\mathcal{R}| \cdot (2\bar{d}|\mathcal{V}|C + 2|\mathcal{V}|C^2) + 2|\mathcal{V}|C^2 + |\mathcal{V}|C.$$

Proof. We divide the computation of RGConv into three steps and compute the FLOPs of each step:

① In the first step, the adjacency of all node pairs on $|\mathcal{R}|$ different relations are summarized in the adjacency matrix $A \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{R}||\mathcal{V}|}$, where the element $A_{i,(j-1)|\mathcal{R}|+k}$ indicates the weight of the edge from the i-th node to the j-th node with the k-th relation:

$$A_{i,(j-1)|\mathcal{R}|+k} = \begin{cases} \frac{1}{|\mathcal{N}_{r_k}(v_j)|} & \text{there is an edge from } i\text{-th node to } j\text{-th node with } k\text{-th relation,} \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where $\mathcal{N}_{r_k}(v_j) = \{u|(u, v_j, r_k) \in \mathcal{E}\}$ is the neighborhood set of node v_j with relation r_k. Using this adjacency matrix, each node will have $|\mathcal{R}|$ different slots to receive the relational messages passed to it. All relational message passing operations can be realized by a sparse matrix multiplication:

$$\tilde{Z} = A^\top Z, \tag{6}$$

where $Z \in \mathbb{R}^{|\mathcal{V}|\times C}$ denotes input node features, and $\tilde{Z} \in \mathbb{R}^{|\mathcal{R}||\mathcal{V}|\times C}$ denotes the relational slots of all nodes after message passing. By utilizing the sparsity of the adjacency matrix, this step consumes following FLOPs:

$$\mathrm{FLOPs}(\mathrm{RGConv}-①) = 2|\mathcal{E}|C = 2\bar{d}|\mathcal{R}||\mathcal{V}|C. \tag{7}$$

② In the second step, we first integrate the relational slots of each node to get the reshaped $\tilde{Z} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{R}|C}$. At this time, each node is represented by a $|\mathcal{R}|C$-dimensional vector, *i.e.*, the aggregated messages of all relations. Next, we concatenate the convolutional kernel matrices of all relations to produce $W^{\mathrm{conv}} \in \mathbb{R}^{|\mathcal{R}|C\times C}$, and this matrix is applied upon $\tilde{Z}$ to combine the messages in the same relational slot and aggregate messages across different relations:

$$Z^{\mathrm{aggr}} = \tilde{Z}W^{\mathrm{conv}}, \tag{8}$$

where $Z^{\mathrm{aggr}} \in \mathbb{R}^{|\mathcal{V}|\times C}$ denotes the aggregated neighborhood information for each node. This step has the FLOPs as below:

$$\mathrm{FLOPs}(\mathrm{RGConv}-②) = 2|\mathcal{R}||\mathcal{V}|C^2. \tag{9}$$

③ In the final step, a self-update with matrix $W^{\mathrm{self}} \in \mathbb{R}^{C\times C}$ is first performed on the input feature of each node, and the self-updated node feature is further added with the aggregated neighborhood information:

$$Z' = ZW^{\mathrm{self}} + Z^{\mathrm{aggr}}, \tag{10}$$

where $Z' \in \mathbb{R}^{|\mathcal{V}|\times C}$ denotes output node features. This step has the FLOPs as below:

$$\mathrm{FLOPs}(\mathrm{RGConv}-③) = 2|\mathcal{V}|C^2 + |\mathcal{V}|C. \tag{11}$$

Therefore, by summing up the computational cost of three steps, the RGConv consumes the following FLOPs in total:

$$\mathrm{FLOPs}(\mathrm{RGConv}) = |\mathcal{R}| \cdot (2\bar{d}|\mathcal{V}|C + 2|\mathcal{V}|C^2) + 2|\mathcal{V}|C^2 + |\mathcal{V}|C.$$

□

Proposition 2. *To process the assumed multi-relational graph, the Gated Relational Message Passing (GRMP) consumes the FLOPs as below under the efficient implementation with sparse matrix multiplication:*

$$\mathrm{FLOPs}(\mathrm{GRMP}) = |\mathcal{R}| \cdot (2\bar{d} + 7)|\mathcal{V}|C + 6|\mathcal{V}|C^2.$$

Proof. Following the steps of GRMP stated in Eq. (3), we compute the FLOPs of each step:

① In the first step, we conduct a pre-layer node-wise channel aggregation with the weight matrix $W^{\mathrm{in}} \in \mathbb{R}^{C \times C}$:

$$Z^{\mathrm{in}} = ZW^{\mathrm{in}}, \tag{12}$$

where $Z \in \mathbb{R}^{|\mathcal{V}| \times C}$ denotes the input node features, and $Z^{\mathrm{in}} \in \mathbb{R}^{|\mathcal{V}| \times C}$ denotes the channel-aggregated node features. This step has the FLOPs consumption as below:

$$\mathrm{FLOPs}(\mathrm{GRMP}-①) = 2|\mathcal{V}|C^2. \tag{13}$$

② In the second step, we first gather the messages within the same relation for each node, which is realized by the sparse matrix multiplication between Z^{in} and the adjacency matrix $A \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{R}||\mathcal{V}|}$ (A is identically defined as in the step ① of Proposition 1):

$$\tilde{Z}^{\mathrm{in}} = A^{\top} Z^{\mathrm{in}}, \tag{14}$$

where $\tilde{Z}^{\mathrm{in}} \in \mathbb{R}^{|\mathcal{R}||\mathcal{V}| \times C}$ represents the relational slots of all nodes after message passing. The relational slots of each node are then integrated to get the reshaped $\tilde{Z}^{\mathrm{in}} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{R}|C}$. By concatenating the convolutional kernel vectors of all relations, we have $w_{\mathrm{conv}} \in \mathbb{R}^{|\mathcal{R}|C \times 1}$, and this vector is broadcast to all nodes to perform channel-wise message aggregation via Hadamard product:

$$\tilde{Z}^{\mathrm{aggr}} = (\mathbf{1}_{\mathrm{conv}} w_{\mathrm{conv}}^{\top}) \odot \tilde{Z}^{\mathrm{in}}, \tag{15}$$

where $\mathbf{1}_{\mathrm{conv}} \in \mathbb{R}^{|\mathcal{V}| \times 1}$ is the all-one vector for broadcasting, and $\tilde{Z}^{\mathrm{aggr}} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{R}|C}$ denotes the relational slots of all nodes after intra-relation message aggregation.

To conduct the operations in Eqs. (14) and (15), this step consumes the following FLOPs:

$$\mathrm{FLOPs}(\mathrm{GRMP}-②) = 2|\mathcal{E}|C + 2|\mathcal{R}||\mathcal{V}|C = 2\bar{d}|\mathcal{R}||\mathcal{V}|C + 2|\mathcal{R}||\mathcal{V}|C. \tag{16}$$

③ In the third step, we first compute the attentive weights assigned to all relations on each node:

$$M^{\alpha} = ZW^{\alpha}, \tag{17}$$

where $W^{\alpha} \in \mathbb{R}^{C \times |\mathcal{R}|}$ is the weight matrix for node-adaptive relation weighting, and $M^{\alpha} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{R}|}$ denotes the relation weights on all nodes. After that, a weighted summation is performed to aggregate the messages of different relations in $\tilde{Z}^{\mathrm{aggr}}$ (in this operation, we use the reshaped $\tilde{Z}^{\mathrm{aggr}} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{R}| \times C}$ and the reshaped $M^{\alpha} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{R}| \times 1}$):

$$\widehat{Z}^{\mathrm{aggr}} = \sum_{i=1}^{|\mathcal{R}|} (M^{\alpha}_{:,i,:} \, \mathbf{1}_{\alpha}^{\top}) \odot \tilde{Z}^{\mathrm{aggr}}_{:,i,:}, \tag{18}$$

where $\mathbf{1}_{\alpha} \in \mathbb{R}^{C \times 1}$ is the all-one vector for broadcasting relation weights to all feature channels, and $\widehat{Z}^{\mathrm{aggr}} \in \mathbb{R}^{|\mathcal{V}| \times C}$ denotes the per-node neighborhood representations after inter-relation message aggregation.

To perform Eqs. (17) and (18), this step has the following FLOPs consumption:

$$\mathrm{FLOPs}(\mathrm{GRMP}-③) = 2|\mathcal{R}||\mathcal{V}|C + |\mathcal{R}| \cdot 2|\mathcal{V}|C + (|\mathcal{R}| - 1)|\mathcal{V}|C = 5|\mathcal{R}||\mathcal{V}|C - |\mathcal{V}|C. \tag{19}$$

④ The fourth step conducts a post-layer node-wise channel aggregation with the weight matrix $W^{\mathrm{out}} \in \mathbb{R}^{C \times C}$:

$$Z^{\mathrm{aggr}} = \widehat{Z}^{\mathrm{aggr}} W^{\mathrm{out}}, \tag{20}$$

where $Z^{\mathrm{aggr}} \in \mathbb{R}^{|\mathcal{V}| \times C}$ denotes the channel-aggregated neighborhood representations. This step consumes the FLOPs as below:

$$\mathrm{FLOPs}(\mathrm{GRMP}-④) = 2|\mathcal{V}|C^2. \tag{21}$$

⑤ In the final step, the input feature of each node first performs self-update with the weight matrix $W^{\text{self}} \in \mathbb{R}^{C\times C}$, and the self-updated node feature is further updated by its neighborhood representation via a gating mechanism:

$$Z' = ZW^{\text{self}} \odot Z^{\text{aggr}}, \tag{22}$$

where $Z' \in \mathbb{R}^{|\mathcal{V}|\times C}$ denotes output node features. This step has the FLOPs as below:

$$\text{FLOPs}(\text{GRMP}-⑤) = 2|\mathcal{V}|C^2 + |\mathcal{V}|C. \tag{23}$$

Therefore, by summing up the computational cost of five steps, the GRMP has the following FLOPs consumption in total:

$$\text{FLOPs}(\text{GRMP}) = |\mathcal{R}| \cdot (2\bar{d} + 7)|\mathcal{V}|C + 6|\mathcal{V}|C^2. \tag{24}$$

□

B More Experimental Setups

Edge message passing. Zhang et al. (2022) proposes to enhance the GearNet by edge-level message passing, which well captures the interactions between edges. To compare with the GearNet-Edge model enhanced in this way, we adapt the same edge message passing scheme to our EurNet.

Specifically, based on the constructed multi-relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, we further construct a *line graph* (Harary & Norman, 1960) $\mathcal{G}_{\text{line}} = (\mathcal{V}_{\text{line}}, \mathcal{E}_{\text{line}}, \mathcal{R}_{\text{line}})$. In this graph, each node $v \in \mathcal{V}_{\text{line}}$ corresponds to an edge in the original graph $\mathcal{G}$. There will an edge (u, v, r) between nodes $u, v \in \mathcal{V}_{\text{line}}$ if the corresponding edges of u and v are adjacent in the original graph, and the edge type $r \in \{0, 1, \cdots, 7\}$ is determined by the angle $\angle_{(u,v)}$'s allocation in 8 equally-divided bins of $[0, \pi]$ ($\angle_{(u,v)}$ denotes the angle between the corresponding edges of u and v in the original graph). Based on this multi-relational line graph, we employ the GRMP layer (Sec. 3.3) to propagate information between the nodes in $\mathcal{G}_{\text{line}}$ and thus between the edges in the original graph $\mathcal{G}$. Readers are referred to Zhang et al. (2022) for more details. We name the EurNet equipped with such an edge message passing scheme as EurNet-Edge.