EURNET: EFFICIENT MULTI-RANGE RELATIONAL MODELING OF SPATIAL MULTI-RELATIONAL DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Modeling spatial relationship in the data remains critical across many different tasks, such as image classification, semantic segmentation and protein structure understanding. Previous works often use a unified solution like relative positional encoding. However, there exists different kinds of spatial relations, including short-range, medium-range and long-range relations, and modeling them separately can better capture the focus of different tasks on the multi-range relations (*e.g.*, short-range relations can be important in instance segmentation, while long-range relations should be upweighted for semantic segmentation). In this work, we introduce the **EurNet** for **E**fficient m**u**lti-range **r**elational modeling. EurNet constructs the multi-relational graph, where each type of edge corresponds to short-, medium- or long-range spatial interactions. In the constructed graph, EurNet adopts a novel modeling layer, called ***g**ated **r**elational **m**essage **p**assing* (**GRMP**), to propagate multi-relational information across the data. GRMP captures multiple relations within the data with little extra computational cost. We study EurNets in two important domains for image and protein structure modeling. Extensive experiments on ImageNet classification, COCO object detection and ADE20K semantic segmentation verify the gains of EurNet over the previous SoTA FocalNet. On the EC and GO protein function prediction benchmarks, EurNet consistently surpasses the previous SoTA GearNet. Our results demonstrate the strength of EurNets on modeling spatial multi-relational data from various domains.

1 INTRODUCTION

This work studies the data that lie in the 2D/3D space and incorporate interacting relations on different spatial ranges. A representative example is the image data, where an object in the image can interact with other adjacent objects via the direct touch, and it can also interact with those distantly relevant ones via gazing, waving hands or pointing. In protein science, the protein 3D structure is another typical example, in which different amino acids can interact in short range by peptide/hydrogen bonds, and they can also interact in medium and long ranges by hydrophobic interaction. We summarize such kind of data as **spatial multi-relational data**.

In various domains, a lot of previous efforts have been made to model the spatial multi-relational data. For image modeling, multi-head self-attention mechanisms (Dosovitskiy et al., 2020; Liu et al., 2021b), convolutional operations with large receptive fields (Ding et al., 2022; Yang et al., 2022) and MLPs for mixing full spatial information (Tolstikhin et al., 2021; Touvron et al., 2021a) are explored to capture multi-range spatial interactions within an image. For protein structure modeling, Zhang et al. (2022) builds multiple groups of edges for different short-range interactions and employs relational graph convolution (Schlichtkrull et al., 2018) for multi-relational modeling. These works either implicitly treat different kinds of spatial relations (*i.e.*, short-range, medium-range and long-range relations) (Tolstikhin et al., 2021; Yang et al., 2022) or handle them by a unified scheme like relative positional encoding (Dosovitskiy et al., 2020; Liu et al., 2021b). However, considering the relative importance of these spatial relations could vary across different tasks (*e.g.*, the great importance of short-range relations in instance segmentation, and the upgraded importance of long-range relations in semantic segmentation), separately modeling each spatial relation is a better solution to capture different tasks' focus. Such a separate modeling approach remains to be explored, and, especially, the approach is expected to have efficient adaptation to large data and model scales.

To attain this goal, we propose the **EurNet** for **E**fficient m**u**lti-range **r**elational modeling. In general, EurNets are a series of relational graph neural networks equipped with graph construction layers, where relational edges are constructed by the layers for capturing multi-range spatial interactions. When instantiated with different domain knowledge (*e.g.*, computer vision or protein science), EurNets can be specialized to tackle important problems like image classification, image segmentation and protein function prediction. To be specific, upon the raw data, EurNet first uses the graph construction layers to build different types of edges that respectively capture the short-, medium- and long-range spatial interactions within the data. For efficient multi-relational modeling over the constructed graph, we next introduce the *gated relational message passing* (**GRMP**) layer as the basic modeling module of EurNet. 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 handle more types of spatial interactions given the same computational budget. EurNet also supports dynamic graph construction and multi-stage modeling that are used in domains like image modeling.

We demonstrate EurNets in image and protein structure modeling. To model image patches with different granularity, we build EurNets with hierarchical graph construction layers and multiple modeling stages and derive a model series with increasing capacity, *i.e.*, **EurNet-T**, **EurNet-S** and **EurNet-B**. These models enjoy comparable or better top-1 accuracy (82.3% *v.s.* 82.3%; 83.6% *v.s.* 83.5%; 84.1% *v.s.* 83.9%) against the previous SoTA FocalNet$_{\text{(LRF)}}$ series (Yang et al., 2022) on ImageNet-1K classification (resolution: 224×224). Similar performance gains are preserved on COCO object detection and ADE20K semantic segmentation. To model protein alpha carbons, we build EurNet with a single-stage model architecture as GearNet Zhang et al. (2022). Under this fix-architecture comparison, EurNet consistently outperforms the SoTA GearNet on standard protein function prediction benchmarks 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. Our results demonstrate that EurNet could be a strong candidate for modeling spatial multi-relational data in various domains.

2 Related Work

Multi-relational data modeling. Multi-relational data are ubiquitous in the real world, *e.g.*, knowledge graphs (Toutanova & Chen, 2015) and customer-product networks (Li et al., 2014). 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 model expressivity and parameter efficiency, and few works (Li et al., 2021) study the computational efficiency for relational modeling at scale. In addition, they can hardly model the spatial multi-relational data whose relational linking structures at different spatial ranges are not originally given (*e.g.*, image patches). EurNet is designed to model such kind of data in a computationally efficient way.

Image modeling. After the dominance of convolutional vision backbones (Simonyan & Zisserman, 2014; He et al., 2016; Tan & Le, 2019) in 2010s, researchers rethink the architectures for more effective image modeling in 2020s. Vision Transformers (Dosovitskiy et al., 2020; Liu et al., 2021b; Wang et al., 2021; Fan et al., 2021) replace convolutions with the self-attention mechanism (Vaswani et al., 2017) to better capture non-local interactions, which gains SoTA performance on various tasks. Following such successes, modern convolutional architectures (Liu et al., 2022; Ding et al., 2022; Yang et al., 2022), all-MLP architectures (Tolstikhin et al., 2021; Touvron et al., 2021a) and vision GNNs (Han et al., 2022) are designed to aggregate long-range spatial context and achieve competitive performance against vision Transformers. By comparison, EurNet captures the multi-range spatial interactions within an image from a novel perspective, *i.e.*, multi-relational modeling.

Protein structure modeling. A variety of protein structure encoders have been developed to acquire informative protein representations on different structural granularity, 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) is a closely

related work which explores multi-relational modeling of residue-level structures with short-range linking and relational graph convolution (RGConv). By comparison, our 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.

3 EURNET FOR EFFICIENT MULTI-RANGE RELATIONAL MODELING

3.1 PROBLEM DEFINITION

This work studies the data $\mathcal{V} = \{v_i\}_{i=1}^N$ with N data units (*e.g.*, patches in an image, alpha carbons in a protein, *etc.*) with the following structure: (1) **spatial interaction on multiple ranges**: data units can interact with each other across diverse spatial ranges; (2) **multi-relational interaction**: multiple interaction types (*i.e.*, relations) exist between different units; (3) **no canonical linking structure**: the linking structures of multi-range interactions are not specified in the raw data.

To effectively model such *spatial multi-relational data*, the model is expected to own following capabilities: (1) **dynamic multi-range linking**: the model can link relevant data units across different spatial ranges, and the linking structure can change along the whole model if desired; (2) **multi-relational linking**: the model divides all links into multiple groups based on their interaction types; (3) **efficient multi-relational modeling**: the model can propagate information among interacting units by taking their interaction types into consideration, and it will not introduce too much extra computation when involving more relations. Keeping all these requirements in mind, we next introduce the high-level designs of EurNet, and we present its detailed instantiations in Sec. 4.

3.2 MULTI-RANGE RELATIONAL GRAPH CONSTRUCTION

We regard each data unit $v \in \mathcal{V}$ as a node in the graph. For the lack of canonical linking structure among the nodes, we therefore seek to build edges among them, especially with considering their interactions on multiple spatial ranges and dynamically adjusting the graph structure if desired.

Multi-range relational edge construction. Given the concepts of spatial and semantic adjacency in a specific domain (*e.g.*, computer vision or protein science), 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 spatial interactions, 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 $\mathcal{R} = \mathcal{R}_{\text{short}} \cup \mathcal{R}_{\text{medium}} \cup \mathcal{R}_{\text{long}}$. Now, the raw data $\mathcal{V}$ is structured as a multi-relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ that is aware of diverse types of interactions within the data.

Dynamic edge construction. A model can focus on different levels of semantics at different modeling stages. For example, for the image modeling problem we consider, a typical hierarchical image encoder (He et al., 2016; Liu et al., 2021b) is split into multiple stages, and it tends to encode low-level features in shallower stages and encode high-level semantics in deeper stages. To accommodate such a hierarchical modeling manner, our graph construction scheme will be dynamically performed before each modeling stage based on the input features (*e.g.*, node coordinates or representations) of the stage, so that each modeling stage can explore its specific neighborhood structures of data units.

3.3 GATED RELATIONAL MESSAGE PASSING

To perform multi-relational modeling over the constructed graph $\mathcal{G}$, the typical method Relational Graph Convolution (RGConv) (Schlichtkrull et al., 2018) 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 the message aggregation from neighborhoods. Taking node v as an example, the RGConv layer updates its representation 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):

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

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 number $|\mathcal{V}|$ and the feature dimension C could be large in many applications, the $2|\mathcal{V}|C^2$ term will be the main obstacle of exploring more relations with moderate extra computation. This scaling behavior will hurt the model capacity in the applications with strict limitations on computational cost, like the image modeling problems we consider.

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 the data. To attain this goal, we propose the Gated Relational Message Passing (GRMP). 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:

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

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} are identical to those in Eq. (1), and all bias terms are omitted for brevity. We analyze the key components of GRMP in Appendix E.1.

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 increasing the number of considered relations. In Fig. 1, we compare the FLOPs of RGConv and GRMP when they respectively serve as the building block of EurNet-T for image modeling (image resolution: 224×224; "T" denotes the tiny-scale model). In this illustrative comparison, we simply connect each node (*i.e.*, image patch) with its K-nearest neighbors in terms of representation similarity, and the connection with the k-th nearest neighbor is regarded as the k-th relation, leading to K relations in total. We can observe that, when increasing the number of neighbors and thus the number of relations, the computational cost of GRMP-based model increases much more gently than the RGConv-based one. This merit of GRMP enables it to explore more relations within the data under the given computational budget, which enhances the model capacity and performance, as shown in Tab. 6.

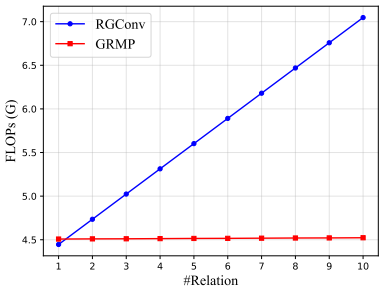


Figure 1: FLOPs trend of RGConv and GRMP under different relation numbers, evaluated on EurNet-T for image modeling.

4 INSTANTIATIONS OF EURNET

In this work, we focus on two application domains, *i.e.*, computer vision and protein science, where the modeling of spatial multi-relational data can solve important problems. We will introduce how to instantiate EurNets to model images and protein structures effectively and efficiently.

4.1 EURNET FOR IMAGE MODELING

4.1.1 RELATIONAL EDGES FOR SHORT-, MEDIUM- AND LONG-RANGE INTERACTIONS

Following previous practices (Dosovitskiy et al., 2020; Liu et al., 2021b), we split an image into local patches and regard these patches as the node set $\mathcal{V}$ of our multi-relational graph. Upon these patches, we construct following relational edges to capture different ranges of spatial interactions within an image (see Fig. 2 for a graphical illustration):

- **Edges for short-range interactions** ($|\mathcal{R}_{\text{short}}| = 4$). We connect each patch with its up, down, left and right patches and regard each direction of adjacency as a relation. These edges capture the one-hop spatial neighbors and thus shortest-range spatial interactions of each image patch.
- **Edges for medium-range interactions** ($|\mathcal{R}_{\text{medium}}| = 1$). In the medium range, a patch can interact with other patches sharing similar semantics (*e.g.*, different body parts of deer in Fig. 2). Therefore, we connect each patch with its K-nearest neighbors in terms of hidden representation similarity (we use the negative Euclidean distance as the similarity metric), and these edges are with the same relation. All edges connecting two patches within the same 2×2 window are removed to avoid short-range linking.

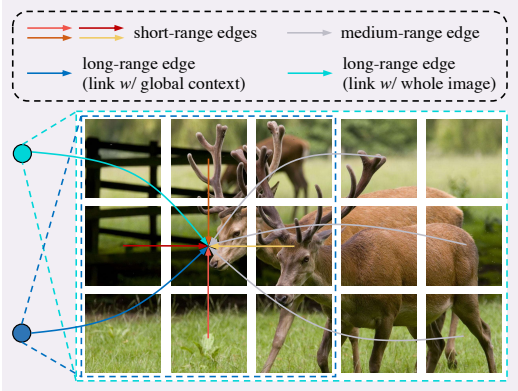


Figure 2: Multi-range relational edges for image.

- **Edges for long-range interactions** ($|\mathcal{R}_{\text{long}}| = 2$). To model long-range interactions, we introduce two kinds of *virtual nodes* and the associated edges. (1) *A virtual node for whole-image representation* is derived by global average pooling over all patch representations, and this virtual node is linked to all patches. (2) *Per-patch virtual nodes for surrounding global context* are got by a stack of depth-wise 2D convolutions (Yang et al., 2022) that aggregate each patch's contextual information with large receptive field and low computation; an edge links each of these virtual nodes to its corresponding patch with a different long-range interacting relation against that in (1), due to the different global context levels represented by two kinds of virtual nodes.

By gathering all these edges representing 7 different relations, we have the edge set $\mathcal{E}$, the relation (*i.e.*, edge type) set $\mathcal{R}$ and the full graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ for multi-relational image modeling.

4.1.2 MODEL ARCHITECTURE

General architecture. In general, we follow the hierarchical image modeling architecture proposed by Swin Transformer (Liu et al., 2021b), which is verified to be a superior architecture and is applied to many vision backbones (Liu et al., 2022; Yang et al., 2021; 2022). Specifically, the whole model is divided into four stages that (1) reduce the number of patches (*i.e.*, nodes in our graph) to a quarter across consecutive stages, and (2) use increasing number of feature channels $[C, 2C, 4C, 8C]$ for all stages. Each stage contains multiple modeling blocks, where we construct each block with a GRMP layer (Sec. 3.3) for relational message passing and a feed-forward network (FFN) (Vaswani et al., 2017) for feature transformation. We adjust the number of feature channels and the number of blocks in each stage to get a model series with increasing capacity, *i.e.*, **EurNet-T**, **EurNet-S** and **EurNet-B**. The detailed architectures of these models are displayed in Appendix B.

Graph construction layers. To adapt the multi-stage modeling manner, we put a graph construction layer before each modeling stage of EurNet-T/S/B. In this way, based on the locations and representations of the patches fed into each stage, the multi-relational graph $\mathcal{G}$ will be reconstructed to adapt

these stage inputs. In particular, along the modeling stages, the edges for medium-range interactions are expected to capture the semantic neighbors on different semantic levels (*e.g.*, from the relevance of low-level features to the relevance of high-level semantics), as studied in Sec. 5.4.

4.2 EURNET FOR PROTEIN STRUCTURE MODELING

4.2.1 RELATIONAL EDGES FOR SHORT-, MEDIUM- AND LONG-RANGE INTERACTIONS

In this work, 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) (see Appendix C for a preliminary introduction to protein structure). In specific, we extract all Cαs as the node set $\mathcal{V}$ of our graph, which, at this time, is actually a set of separate points in the 3D space, since there is no chemical bond among Cαs. To describe the multi-range spatial interactions within a protein, we build following relational edges (see Fig. 3 for a graphical illustration):

- **Edges for short-range interactions** ($|\mathcal{R}_{\text{short}}| = 6$). We adopt two kinds of short-range edges proposed by Zhang et al. (2022). (1) *Sequential edges* connect the Cα nodes that are 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). (2) *Radius edges* connect the Cα nodes within the Euclidean distance of 10 angstroms, and all radius edges have the same relation.
- **Edges for medium-range interactions** ($|\mathcal{R}_{\text{medium}}| = 2$). To capture medium-range interactions exclusively, 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 neighbors are regarded as two different relations.
- **Edges for long-range interactions** ($|\mathcal{R}_{\text{long}}| = 1$). To capture the interactions beyond short- and medium-range interactions above, we introduce a *virtual node* representing the whole protein by taking global average pooling over all Cα representations, and this virtual node is linked to all Cα nodes with a single relation. These edges make each Cα aware of the status of all other Cαs, and thus the long-range interactions beyond short and medium ranges can be captured.

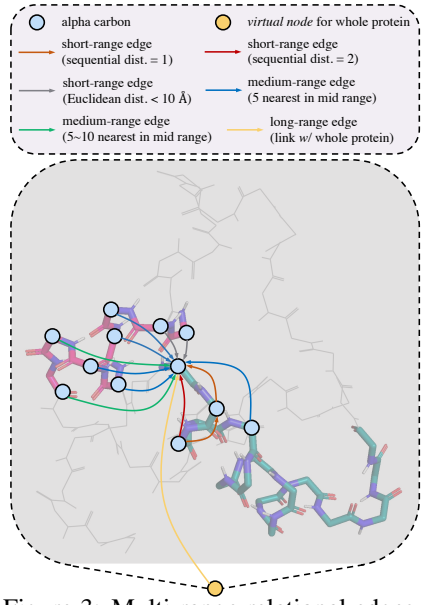


Figure 3: Multi-range relational edges for protein Cαs. *Abbr.*, dist.: distance.

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

4.2.2 MODEL ARCHITECTURE

This work focuses on comparing the graph construction and message passing schemes of EurNet against the SoTA GearNet (Zhang et al., 2022), and we thus follow its single-stage model architecture for fair comparison. Specifically, EurNet performs graph construction once before this only modeling stage, and the input node feature is the one-hot encoding of each Cα's corresponding amino acid. Upon these inputs, six GRMP layers (Sec. 3.3) are stacked for 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. Upon this output, EurNet performs a downstream task by appending a task-specific prediction head. We leave the design of the protein structure encoder with multiple modeling stages as a future work.

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).

5 EXPERIMENTS

5.1 PERFORMANCE COMPARISON ON IMAGE MODELING

5.1.1 BASELINE METHODS

We do point-by-point comparisons between our EurNet series, the SoTA ConvNeXt (Liu et al., 2022) and FocalNet (Yang et al., 2022) series, and other standard series including Swin Transformer (Liu et al., 2021b), FocalAtt (Yang et al., 2021) and ViG (Han et al., 2022). For completeness, we also report the results of EffNet (Tan & Le, 2019), EffNetV2 (Tan & Le, 2021), ViT (Dosovitskiy et al., 2020), DeiT (Touvron et al., 2021b), PVT (Wang et al., 2021), Mixer (Tolstikhin et al., 2021), gMLP (Liu et al., 2021a) and ResMLP (Touvron et al., 2021a) in applicable cases.

5.1.2 IMAGE CLASSIFICATION ON IMAGENET-1K

Setups. In this set of experiments, we benchmark the classification performance of different backbones on ImageNet-1K (Deng et al., 2009) (with 1.28M training and 50K validation images from 1,000 classes) in terms of top-1 accuracy. We consider both (1) training on ImageNet-1K from scratch and (2) pretraining on ImageNet-22K followed by ImageNet-1K fine-tuning. For fair comparison, we follow the standard training configurations of Swin Transformer (Liu et al., 2021b) with minor changes. Detailed model and training configurations are stated in Appendix D.1.

Results. In Tab. 1, for training on ImageNet-1K from scratch, EurNets outperform or align previous SoTA baselines on 224^2 image size, *i.e.*, EurNet-T *v.s.* FocalNet$_{\text{(LRF)}}$-T: **82.3**% *v.s.* **82.3**%; EurNet-S *v.s.* FocalNet$_{\text{(LRF)}}$-S: **83.6**% *v.s.* 83.5%; EurNet-B *v.s.* FocalNet$_{\text{(LRF)}}$-B: **84.1**% *v.s.* 83.9%. Following Swin Transformer, we lift the resolution to 384^2 for 30 epochs fine-tuning after training EurNet-B for 300 epochs on the 224^2 resolution, this model gains **85.4**% top-1 accuracy, outperforming ConvNeXt-B. These results demonstrate the effectiveness of EurNets on modeling images with different resolutions.

For ImageNet-22K pre-training, EurNet-B is on par with the previous SoTA ConvNeXt-B and clearly outperforms Swin-B. The pre-training on large scale is widely regarded as the strength of the models with few inductive biases like Swin-B; while our results show that the well-designed EurNet-B with more inductive biases could also be effective, aligning with ConvNeXt's finding.

Table 1: ImageNet-1K classification results. We measure throughput on a V100 GPU. † denotes the model pre-trained on ImageNet-22K. 224^2 and 384^2 denote the image size. "↑384" means fine-tuning on 384×384 images for 30 epochs.

Model	#Params. (M)	FLOPs (G)	Throughput (imgs/s)	Top-1 Acc (%)
Train on ImageNet-1K from scratch				
EffNet-B7	66	37.0	55	84.3
EffNetV2-L	120	53.0	84	85.7
ViT-S/16	22	4.6	939	79.9
ViT-B/16	87	17.6	330	81.8
DeiT-S/16	22	4.6	979	79.8
DeiT-S/16	87	17.6	302	81.8
PVT-Small	25	3.8	794	79.8
PVT-Medium	44	6.7	517	81.2
PVT-Large	61	9.8	352	81.7
Mixer-B/16	60	12.7	455	76.4
gMLP-S	20	4.5	785	79.6
gMLP-B	73	15.8	301	81.6
ResMLP-S24	30	6.0	871	79.4
ResMLP-B24	129	23.0	61	81.0
Swin-T	28	4.9	760	81.2
Pyramid ViG-S	27	4.6	-	82.1
FocalAtt-T	29	4.9	319	82.2
FocalNet$_{\text{(LRF)}}$-T	29	4.5	696	**82.3**
ConvNeXt-T	29	4.5	775	82.1
EurNet-T	29	4.6	530	**82.3**
Swin-S	50	8.7	435	83.1
Pyramid ViG-M	52	8.9	-	83.1
FocalAtt-S	51	9.4	192	83.5
FocalNet$_{\text{(LRF)}}$-S	50	8.7	406	83.5
ConvNeXt-S	50	8.7	447	83.1
EurNet-S	50	8.8	314	**83.6**
Swin-B	88	15.4	291	83.5
Pyramid ViG-B	93	16.8	-	83.7
FocalAtt-B	90	16.4	138	83.8
FocalNet$_{\text{(LRF)}}$-B	89	15.4	269	83.9
ConvNeXt-B	89	15.4	292	83.8
EurNet-B	89	15.6	224	**84.1**
Swin-B↑384	88	47.1	85	84.5
ConvNeXt-B↑384	89	45.0	96	85.1
EurNet-B↑384	90	46.6	69	**85.4**
Pre-train on ImageNet-22K & Fine-tune on ImageNet-1K				
Swin-B† (224^2)	88	15.4	291	85.2
FocalNet$_{\text{(SRF)}}$-B† (224^2)	88	15.3	280	85.6
ConvNeXt-B† (224^2)	89	15.4	292	**85.8**
EurNet-B† (224^2)	89	15.6	224	85.7
Swin-B† (384^2)	88	47.0	85	86.4
FocalNet$_{\text{(SRF)}}$-B† (384^2)	88	44.8	94	86.5
ConvNeXt-B† (384^2)	89	45.1	96	86.8
EurNet-B† (384^2)	90	46.6	69	**87.0**

Throughput analysis. The throughput of EurNet is higher than FocalAtt while lower than FocalNet and ConvNeXt. We point out that 2D convolutions (*i.e.*, the core of FocalNet and ConvNeXt) are well supported by CUDA kernels, while such supports are still ongoing for graph operations (Chen et al., 2020; Min et al., 2021). EurNet's further speedup is expected under maturer CUDA supports.

5.1.3 OBJECT DETECTION ON COCO

Setups. This experiment benchmarks the object detection and instance segmentation performance on COCO 2017 (Lin et al., 2014). All models are trained on 118K training images and evaluated on

Table 2: COCO object detection and instance segmentation results with Mask R-CNN (He et al., 2017).

Model	#Params. (M)	FLOPs (G)	Mask R-CNN 1×						Mask R-CNN 3×					
			AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
PVT-Small	44.1	245	40.4	62.9	43.8	37.8	60.1	40.3	43.0	65.3	46.9	39.9	62.5	42.8
Swin-T	47.8	264	43.7	66.6	47.7	39.8	63.3	42.7	46.0	68.1	50.3	41.6	65.1	44.9
FocalAtt-T	48.8	291	44.8	67.7	49.2	41.0	64.7	44.2	47.2	69.4	51.9	42.7	**66.5**	45.9
FocalNet$_{(LRF)}$-T	48.9	268	**46.1**	68.2	**50.6**	41.5	65.1	44.5	**48.0**	**69.7**	**53.0**	**42.9**	**66.5**	**46.1**
EurNet-T	49.8	271	**46.1**	**68.7**	50.5	**41.6**	**65.5**	**44.6**	47.8	69.5	52.3	**42.9**	**66.5**	**46.1**
PVT-Medium	63.9	302	42.0	64.4	45.6	39.0	61.6	42.1	44.2	66.0	48.2	40.5	63.1	43.5
Swin-S	69.1	354	46.5	68.7	51.3	42.1	65.8	45.2	48.5	70.2	53.5	43.3	67.3	46.6
FocalAtt-S	71.2	401	47.4	69.8	51.9	42.8	66.6	46.1	48.8	70.5	53.6	43.8	67.7	47.2
FocalNet$_{(LRF)}$-S	72.3	365	48.3	**70.5**	53.1	43.1	**67.4**	**46.2**	49.3	**70.7**	54.2	43.8	**67.9**	47.4
EurNet-S	72.8	364	**48.4**	**70.5**	**53.2**	**43.2**	**67.4**	**46.2**	**49.4**	**70.7**	**54.5**	**44.0**	67.6	**47.5**
PVT-Large	81.0	364	42.9	65.0	46.6	39.5	61.9	42.5	44.5	66.0	48.3	40.7	63.4	43.7
Swin-B	107.1	497	46.9	69.2	51.6	42.3	66.0	45.5	48.5	69.8	53.2	43.4	66.8	46.9
FocalAtt-B	110.0	533	47.8	70.2	52.5	43.2	67.3	46.5	49.0	70.1	53.6	43.7	67.6	47.0
FocalNet$_{(LRF)}$-B	111.4	507	49.0	70.9	53.9	43.5	67.9	46.7	49.8	70.9	54.6	44.1	68.2	47.2
EurNet-B	112.1	506	**49.3**	**71.8**	**54.0**	**43.9**	**68.2**	**47.2**	**50.1**	**71.5**	**55.1**	**44.5**	**68.7**	**47.8**

5K validation images. Two standard training schedules, *i.e.*, the 1× schedule with 12 epochs and the 3× schedule with 36 epochs, are used for benchmarking. Detailed setups are stated in Appendix D.2.

Results. In Tab. 2, EurNet performs comparably to FocalNet$_{(LRF)}$ on the tiny and small model scales. We can observe the superiority of EurNet-B over FocalNet$_{(LRF)}$-B on the base model scale (better performance on all 12 metrics). The base-scale EurNet-B owns $[2, 2, 18, 2]$ modeling blocks (more than EurNet-T) and $[128, 256, 512, 1024]$ feature channels (more than EurNet-S) for four modeling stages. Therefore, *larger message passing hops* (achieved by more modeling blocks) coupled with larger model width favor EurNet's performance on high-resolution dense prediction tasks.

5.1.4 Semantic Segmentation on ADE20K

Setups. In this experiment, we benchmark the semantic segmentation performance of different backbones on ADE20K (Zhou et al., 2017) which contains 20K training, 2K validation and 3K test images. The mIoU metrics under both single- and multi-scale (MS) evaluation are reported. We provide more details in Appendix D.3.

Results. Tab. 3 reports all results. It can be observed that EurNet-T, EurNet-S and EurNet-B achieve the best performance on their corresponding model scales under both evaluation metrics. Such consistent performance gains verify the effectiveness of EurNet on the dense prediction tasks that require to model fine-grained semantics and long-range interactions.

Table 3: ADE20K semantic segmentation results with UperNet (Xiao et al., 2018).

Model	#Params. (M)	FLOPs (G)	mIoU	+MS
Swin-T	60	941	44.5	45.8
FocalAtt-T	62	998	45.8	47.0
ConvNeXt-T	60	939	-	46.7
FocalNet$_{(LRF)}$-T	61	949	46.8	47.8
EurNet-T	62	948	**47.2**	**48.4**
Swin-S	81	1038	47.6	49.5
FocalAtt-S	85	1130	48.0	50.0
ConvNeXt-S	82	1027	-	49.6
FocalNet$_{(LRF)}$-S	84	1044	49.1	50.1
EurNet-S	85	1042	**49.8**	**50.8**
Swin-B	121	1188	48.1	49.7
FocalAtt-B	126	1354	49.0	50.5
ConvNeXt-B	122	1170	-	49.9
FocalNet$_{(LRF)}$-B	126	1192	50.5	51.4
EurNet-B	126	1190	**50.7**	**51.8**

5.2 Performance Comparison on Protein Structure Modeling

5.2.1 Baseline Methods

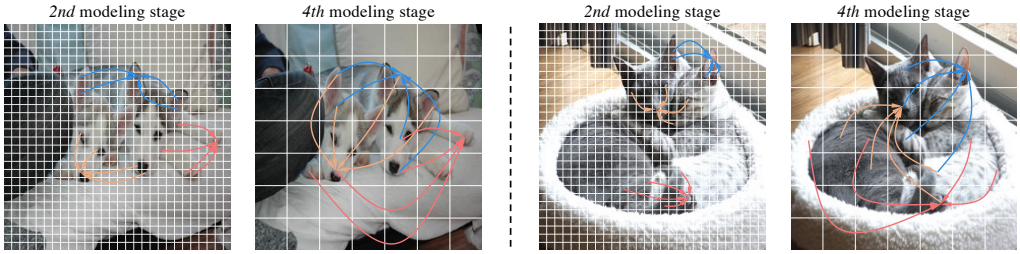
We compare with the SoTA GearNet (Zhang et al., 2017) under two settings, *i.e.*, with and without edge message passing ("-Edge" in Tab. 4). We also include other baselines, *i.e.*, 3DCNN_MQA (Derevyanko et al., 2018), GCN (Kipf & Welling, 2016), GAT (Veličković et al., 2017), GVP (Jing et al., 2021), GraphQA (Baldassarre et al., 2021) and New IEConv (Hermosilla & Ropinski, 2022), for complete comparisons.

Table 4: F_{max} results on EC and GO protein function prediction benchmarks.

Model	EC	GO-BP	GO-MF	GO-CC
3DCNN_MQA	0.077	0.240	0.147	0.305
GCN	0.320	0.252	0.195	0.329
GAT	0.368	0.284	0.317	0.385
GVP	0.489	0.326	0.426	0.420
GraphQA	0.509	0.308	0.329	0.413
New IEConv	0.735	0.374	0.544	**0.444**
GearNet	0.730	0.356	0.503	0.414
EurNet	**0.768**	**0.437**	**0.563**	**0.421**
GearNet-Edge	0.810	0.403	0.580	0.450
EurNet-Edge	**0.829**	**0.456**	**0.592**	**0.453**

5.2.2 Protein Function Prediction

Setups. This set of experiments compare different protein structure encoders on the EC (Gligorijević et al., 2021) and GO (Gligorijević et al., 2021) protein function prediction benchmarks. We follow GearNet to report the protein-centric maximum F-score F_{max}, a commonly-used metric in CAFA challenges (Radivojac et al., 2013). More dataset, model and training details are in Appendix D.4.



(a) (b)

Figure 4: Medium-range edges built by EurNet-T (we use different colors for different selected target nodes).

Results. In Tab. 4, we can observe that EurNet consistently outperforms GearNet on all four tasks, and the performance gains preserve after involving edge message passing (details of edge message passing are stated in Appendix D.4). Since EurNet follows the single-stage model architecture of GearNet, 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.

5.3 Ablation Study

Effect of multi-range relational edges. In Tab. 5, we evaluate EurNet-T on ImageNet-1K with different ranges of edges. When using a single range, the model with long-range edges achieves the highest accuracy 81.7%, which verifies the importance of capturing long-range interactions in image classification. By further adding short- or medium-range edges, the performance is promoted to 82.0%, where more fine-grained local interactions are captured. By using all three ranges of edges, the full model of EurNet-T obtains the 82.3% accuracy, which proves the complementarity of short-, medium- and long-range edges. Ablation study for protein structure is in Appendix E.2.

Table 5: Ablation study of multi-range edges on ImageNet-1K with EurNet-T.

short	medium	long	Top-1 Acc (%)
✓			80.7
	✓		79.3
		✓	**81.7**
✓	✓		81.5
✓		✓	**82.0**
	✓	✓	**82.0**
✓	✓	✓	**82.3**

Effect of GRMP layer. In Tab. 6, we compare between RGConv and GRMP under the comparable parameter number, FLOPs and throughput. (1) GRMP's dimensions are first set as $[96, 192, 384, 768]$ in four stages. To reach comparable cost, RGConv can only have the dimensions of $[84, 168, 336, 672]$ and achieves a lower accuracy 81.5% than GRMP's 82.3%. (2) After increasing RGConv's dimensions to $[96, 192, 384, 768]$, it aligns GRMP's performance while introduces more cost (1.3G more FLOPs). Under comparable cost, GRMP can have $[108, 216, 432, 864]$ dimensions, leading to a higher accuracy 82.7%. These results demonstrate the better efficiency-performance trade-off gained by GRMP. Ablation study for protein structure modeling is in Appendix E.3.

Table 6: Ablation study of multi-relational modeling layer on ImageNet-1K with EurNet-T.

Layer	Hidden Dimensions	#Params. (M)	FLOPs (G)	Throughput (imgs/s)	Top-1 Acc (%)
RGConv	$[84, 168, 336, 672]$	28.8	4.6	541.8	81.5
GRMP	$[96, 192, 384, 768]$	28.8	4.6	530.3	**82.3**
RGConv	$[96, 192, 384, 768]$	37.3	5.9	451.2	82.2
GRMP	$[108, 216, 432, 864]$	36.3	5.8	444.5	**82.7**

5.4 Visualization

Fig. 4 displays some medium-range edges built by the EurNet-T trained on ImageNet-1K. The edges for the 2nd stage connect the patches with similar low-level features (*e.g.*, the patches of red dog ears in Fig. 4(a)), while the edges for the 4th stage connect semantically relevant patches (*e.g.*, different body parts of two dogs in Fig. 4(a)), which shows EurNet-T's hierarchical image modeling ability.

6 Conclusions and Future Work

This work proposes the EurNet to model spatial multi-relational data like image patches and protein alpha carbons. It builds relational edges on multiple spatial ranges to describe the interactions in the data. It uses the gated relational message passing layer to model the built multi-relational graph, which can efficiently adapt to large data and model scales. The instantiations of EurNet have gained superior performance on various image and protein structure modeling tasks.

In future works, we will adapt EurNet to more tasks of other domains like 3D point cloud modeling for object and scene understanding, and we will explore a general hierarchical multi-relational modeling method for the data from various domains.

REPRODUCIBILITY STATEMENT

For the sake of reproducibility, we use Tab. 7 to provide detailed architectures of EurNet-T, EurNet-S and EurNet-B for image modeling, state the detailed single-stage architecture of EurNet for protein structure modeling in Sec. 4.2.2, and describe the model configurations for each specific task in Sec. D. We state the detailed training configurations of all considered tasks in Sec. D. The derivation of the FLOPs of RGConv (Eq. (2)) and GRMP (Eq. (4)) are provided in Sec. A. In the supplementary material, we submit all source code for reproducing the results of ImageNet classification experiments, and the source code for COCO object detection, ADE20K semantic segmentation and EC and GO protein function prediction experiments will be released to public upon acceptance.

REFERENCES

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.

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.

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.

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.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

Zhaodong Chen, Mingyu Yan, Maohua Zhu, Lei Deng, Guoqi Li, Shuangchen Li, and Yuan Xie. fusegnn: accelerating graph convolutional neural network training on gpgpu. In *IEEE/ACM International Conference On Computer Aided Design*, 2020.

MMSegmentation Contributors. Mmsegmentation: Openmmlab semantic segmentation toolbox and benchmark. *Availabe online: https://github. com/open-mmlab/mmsegmentation (accessed on 18 May 2022)*, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.

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

Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *IEEE/CVF International Conference on Computer Vision*, 2021.

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.

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.

Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision gnn: An image is worth graph of nodes. *arXiv preprint arXiv:2206.00272*, 2022.

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

Michael J Harms and Joseph W Thornton. Analyzing protein structure and function using ancestral gene reconstruction. *Current Opinion in Structural Biology*, 20(3):360–366, 2010.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE/CVF International Conference on Computer Vision*, 2017.

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

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.

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`.

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.

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.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. *Advances in Neural Information Processing Systems*, 2021a.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision*, 2021b.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

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

Seung Won Min, Kun Wu, Sitao Huang, Mert Hidayetoğlu, Jinjun Xiong, Eiman Ebrahimi, Deming Chen, and Wen-mei Hwu. Large graph convolutional network training with gpu-oriented data communication architecture. *arXiv preprint arXiv:2103.03330*, 2021.

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

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NeurIPS Workshop*, 2017.

Predrag Radivojac, Wyatt T Clark, Tal Ronnen Oron, Alexandra M Schnoes, Tobias Wittkop, Artem Sokolov, Kiley Graim, Christopher Funk, Karin Verspoor, Asa Ben-Hur, et al. A large-scale evaluation of computational protein function prediction. *Nature methods*, 10(3):221–227, 2013.

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.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019.

Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, 2021.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 2021.

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.

Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021a.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 2021b.

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.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *IEEE/CVF International Conference on Computer Vision*, 2021.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision*, 2018.

Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal attention for long-range interactions in vision transformers. *Advances in Neural Information Processing Systems*, 2021.

Jianwei Yang, Chunyuan Li, and Jianfeng Gao. Focal modulation networks. *arXiv preprint arXiv:2203.11926*, 2022.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *IEEE/CVF International Conference on Computer Vision*, 2019.

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.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

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:*

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

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:

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

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 $W^{\text{conv}} \in \mathbb{R}^{|\mathcal{R}|C \times C}$, and this matrix is applied upon $\tilde{Z}$ to combine the messages in the same relational slot and aggregate messages across different relations:

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

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

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

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

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

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

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

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

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

□

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:*

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

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 $W^{\text{in}} \in \mathbb{R}^{C\times C}$:

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

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

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

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):

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

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

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

where $\mathbf{1}_{\text{conv}} \in \mathbb{R}^{|\mathcal{V}|\times 1}$ is the all-one vector for broadcasting, and $\tilde{Z}^{\text{aggr}} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{R}|C}$ 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:

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

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 $\tilde{Z}^{\text{aggr}}$ (in this operation, we use the reshaped $\tilde{Z}^{\text{aggr}} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{R}|\times C}$ and the reshaped $M^\alpha \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{R}|\times 1}$):

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

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

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

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

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

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

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

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

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 Detailed Model Architecture for Image Modeling

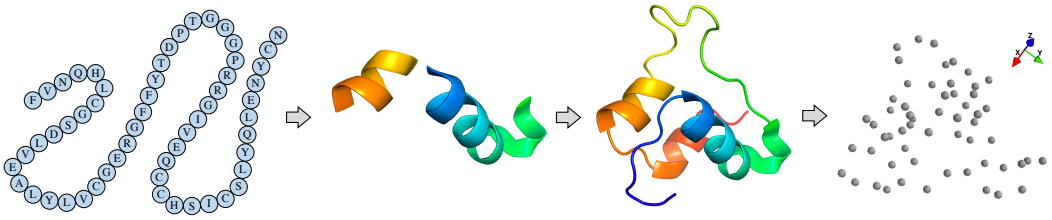
Table 7: Detailed architectures of EurNet-T/S/B for ImageNet-1K classification (#parameters and FLOPs are computed under the resolution 224×224). $H\times W$: input image resolution; C: number of feature channels; γ: FFN's hidden dimension ratio; K: number of K-nearest neighbors for medium-range edges; $\mathcal{Y}$: label set for classification. "T" denotes the tiny model; "S" denotes the small model; "B" denotes the base model.

Module	#Patches	EurNet-T	EurNet-S	EurNet-B
Stem	$\frac{H}{4}\times\frac{W}{4}$	4×4 conv, stride=4	4×4 conv, stride=4	4×4 conv, stride=4
Graph Construction	$\frac{H}{4}\times\frac{W}{4}$	[short-range edges, long-range edges]	[short-range edges, long-range edges]	[short-range edges, long-range edges]
Stage 1	$\frac{H}{4}\times\frac{W}{4}$	[GRMP ($C=96$), FFN ($C=96$, $\gamma=4$)] $\times2$	[GRMP ($C=96$), FFN ($C=96$, $\gamma=4$)] $\times2$	[GRMP ($C=128$), FFN ($C=128$, $\gamma=4$)] $\times2$
Downsample	$\frac{H}{8}\times\frac{W}{8}$	PatchMerging	PatchMerging	PatchMerging
Graph Construction	$\frac{H}{8}\times\frac{W}{8}$	[short-range edges, medium-range edges ($K=12$), long-range edges]	[short-range edges, medium-range edges ($K=12$), long-range edges]	[short-range edges, medium-range edges ($K=12$), long-range edges]
Stage 2	$\frac{H}{8}\times\frac{W}{8}$	[GRMP ($C=192$), FFN ($C=192$, $\gamma=4$)] $\times2$	[GRMP ($C=192$), FFN ($C=192$, $\gamma=4$)] $\times2$	[GRMP ($C=256$), FFN ($C=256$, $\gamma=4$)] $\times2$
Downsample	$\frac{H}{16}\times\frac{W}{16}$	PatchMerging	PatchMerging	PatchMerging
Graph Construction	$\frac{H}{16}\times\frac{W}{16}$	[short-range edges, medium-range edges ($K=12$), long-range edges]	[short-range edges, medium-range edges ($K=12$), long-range edges]	[short-range edges, medium-range edges ($K=12$), long-range edges]
Stage 3	$\frac{H}{16}\times\frac{W}{16}$	[GRMP ($C=384$), FFN ($C=384$, $\gamma=4$)] $\times6$	[GRMP ($C=384$), FFN ($C=384$, $\gamma=4$)] $\times18$	[GRMP ($C=512$), FFN ($C=512$, $\gamma=4$)] $\times18$
Downsample	$\frac{H}{32}\times\frac{W}{32}$	PatchMerging	PatchMerging	PatchMerging
Graph Construction	$\frac{H}{32}\times\frac{W}{32}$	[short-range edges, medium-range edges ($K=12$), long-range edges]	[short-range edges, medium-range edges ($K=12$), long-range edges]	[short-range edges, medium-range edges ($K=12$), long-range edges]
Stage 4	$\frac{H}{32}\times\frac{W}{32}$	[GRMP ($C=768$), FFN ($C=768$, $\gamma=4$)] $\times2$	[GRMP ($C=768$), FFN ($C=768$, $\gamma=4$)] $\times2$	[GRMP ($C=1024$), FFN ($C=1024$, $\gamma=4$)] $\times2$
Head	1×1	Pooling & Linear ($\lvert\mathcal{Y}\rvert=1000$)	Pooling & Linear ($\lvert\mathcal{Y}\rvert=1000$)	Pooling & Linear ($\lvert\mathcal{Y}\rvert=1000$)
#Parameters (M)		28.8	50.2	88.7
FLOPs (G)		4.6	8.8	15.6

For image modeling, we basically follow the hierarchical architecture proposed by Swin Transformer (Liu et al., 2021b), as summarized in Tab. 7. The architecture begins with a patch embedding module implemented by non-overlapping 2D convolution. After that, the model is split into 4 modeling stages: (1) the number of patches (*i.e.*, nodes in our graph) is reduced to a quarter across consecutive stages by the "PatchMerging" operation (Liu et al., 2021b); (2) increasing feature channels $[C, 2C, 4C, 8C]$ are used for all stages. We place a graph construction layer before each modeling stage to update the multi-relational graph structure. For the first stage, we only use short- and long-range edges to reduce the computational cost (computing medium-range edges by representation

similarity comparison is expensive in the first stage with many patches), and the relational edges of all three ranges are adopted in the last three stages. Each stage is composed of multiple modeling blocks, where each block contains a GRMP layer (Sec. 3.3) for relational message passing and a feed-forward network (FFN) (Vaswani et al., 2017) for feature transformation. In the end, a global average pooling layer produces the whole-image representation, and a linear head outputs the final prediction. We adjust the number of feature channels and the number of blocks in each stage to derive EurNet-T, EurNet-S and EurNet-B with standard number of parameters and FLOPs. We implement the models based on the PyTorch (Paszke et al., 2017) deep learning library.

C INTRODUCTION TO PROTEIN STRUCTURE



(a) Primary structure (b) Secondary structure (c) Tertiary structure (d) Alpha carbons (Cαs)

Figure 5: The primary structure, secondary structure, tertiary structure and all alpha carbons of the single-chain insulin protein (ID in PDB (Berman et al., 2000): 2LWZ).

Proteins are macromolecules that perform critical biological functions in living organisms. A protein owns multiple levels of structures, as described below:

- **Primary structure** (Fig. 5(a)). At the chemical level, a protein is composed of one or multiple chains of amino acid residues, forming the protein sequence which is the primary protein structure. In the protein sequence $s = (s_1, s_2, \cdots, s_L)$, each element s_l denotes a type of amino acid (there are 20 common amino acids and two rare ones, *i.e.*, Selenocysteine and Pyrrolysine). The primary structure tells the sequential order of amino acids in a protein, but otherwise it does not reveal any information about the 3D folded structure of the protein. This fact limits its usefulness in the analysis/prediction of protein functions, due to the principle that "protein folded structures largely determine their functions" (Harms & Thornton, 2010).

- **Secondary structure** (Fig. 5(b)). The secondary structures of proteins are some repeatedly-occurred local structures like the α-helices shown in Fig. 5(b). These structures are stabilized by hydrogen bonds, and, together with the tight turns and flexible loops in between, they constitute the complete protein folded structure.

- **Tertiary structure** (Fig. 5(c)). The spatial arrangement of different secondary structure components leads to the formation of the tertiary structure (*i.e.*, the folded structure of a protein). The tertiary structure is jointly held by short-range interactions like hydrogen bonding and long-range interactions like hydrophobic interactions. Thanks to the recent advances of highly accurate protein folded structure predictors based on deep learning (Jumper et al., 2021; Baek et al., 2021), we can now efficiently acquire numerous previously unknown protein tertiary structures with reasonable confidence. These advances are expected to promote the understanding of protein functions based on tertiary structures.

In this work, we focus on protein function prediction tasks based on tertiary structures. Specifically, we adopt an informative and light-weight representation format, *i.e.*, **all alpha carbons (Cαs) in the tertiary structure** (Fig. 5(d)), which is widely used in the literature (Gligorijević et al., 2021; Baldassarre et al., 2021; Zhang et al., 2022). A Cα can be seen as the center of its corresponding amino acid, and thus the overall tertiary structure of a protein can be well captured by the collection of all Cαs. At this time, the Cαs are actually a set of separate points in the 3D space, since there is no chemical bond among them. To better describe the interactions within a protein, we seek to construct edges among Cαs and lead to a more informative representation format, *i.e.*, the **Cα graph**.

D MORE EXPERIMENTAL SETUPS

D.1 MORE EXPERIMENTAL SETUPS ON IMAGENET-1K CLASSIFICATION

In the following, we state the detailed model and training configurations of (1) training on ImageNet-1K from scratch and (2) pretraining on ImageNet-22K followed by ImageNet-1K fine-tuning. For training configurations, we mainly follow the standards set up by Swin Transformer (Liu et al., 2021b) for fair comparison.

D.1.1 FROM-SCRATCH TRAINING ON IMAGENET-1K

Model configurations. The whole model architectures of EurNet-T, EurNet-S and EurNet-B are presented in Tab. 7. For medium-range edges, 12 nearest semantic neighbors of each patch are linked to it to capture medium-range interactions. For long-range edges, we compute the representations of per-patch global-context virtual nodes by a stack of depth-wise 2D convolutions with the accumulative receptive field as 7, and these virtual nodes are linked to their corresponding patches.

Training configuration. An AdamW (Loshchilov & Hutter, 2017) optimizer (betas: $[0.9, 0.999]$, weight decay: 0.05) is employed to train each EurNet model for 300 epochs. We set the batch size as 2048, the base learning rate as 0.002 and the gradient clipping norm as 5.0. A cosine learning rate scheduler is adopted to adjust the learning rate from 2.0×10^{-6} to 0.002 in the first 20 warm-up epochs, and the learning rate is decayed to 2.0×10^{-5} in the rest epochs with a cosine rate. The stochastic depth drop rates are set to 0.15, 0.3 and 0.5 respectively for EurNet-T, EurNet-S and EurNet-B. We follow the augmentation functions and mixup strategies used in Swin Transformer. All experiments are conducted on 16 Tesla-V100-32GB GPUs.

D.1.2 PRE-TRAINING ON IMAGENET-22K AND FINE-TUNING ON IMAGENET-1K

Model configurations. The EurNet-B with the standard model architecture as in Tab. 7 is used, except that the last linear classification head outputs 21,841-dimensional logits to perform ImageNet-22K classification.

Training configuration. For ImageNet-22K pre-training, we train EurNet-B with an AdamW optimizer (betas: $[0.9, 0.999]$, weight decay: 0.05) for 90 epochs with the batch size 4096 and the image resolution 224×224. A cosine learning rate scheduler is employed to linearly increase the learning rate from 0 to 4.0×10^{-3} in the first 5 warm-up epochs, and it decays the learning rate to 1.0×10^{-6} in the rest epochs with a cosine rate. The stochastic depth drop rate is set as 0.1. All augmentation functions and mixup strategies follow Swin Transformer. The pre-training is performed on 64 Tesla-V100-32GB GPUs.

For ImageNet-1K fine-tuning, the pre-trained model is fine-tuned for 30 epochs by an AdamW optimizer (betas: $[0.9, 0.999]$, weight decay: 1.0×10^{-8}). The cosine learning rate scheduler adjusts the learning rate from 8.0×10^{-8} to 8.0×10^{-5} in the first 5 warm-up epochs, and the learning rate is decayed to 8.0×10^{-7} in the rest epochs with a cosine rate. The stochastic depth drop rate is set as 0.2. Both Mixup (Zhang et al., 2017) and CutMix (Yun et al., 2019) are muted during fine-tuning, following FocalNet Yang et al. (2022). The fine-tuning is performed on 16 Tesla-V100-32GB GPUs.

D.1.3 THROUGHPUT COMPUTATION

We follow Swin Transformer to measure the inference throughput on a Tesla-V100-32GB GPU with batch size 128. We adopt graph checkpoints to enhance the speed of inferring an image that has been seen. During inference, we add short-range edges to the list of medium-range edges and merge their corresponding relations to further promote the efficiency.

D.2 MORE EXPERIMENTAL SETUPS ON COCO OBJECT DETECTION

Model configurations. We use the EurNet-T, EurNet-S and EurNet-B pre-trained on ImageNet-1K as the backbone of Mask R-CNN (He et al., 2017). In specific, we take the patch representations output from all four modeling stages as the inputs of the Feature Pyramid Network (FPN) Lin et al. (2017). For medium-range edge construction on the high-resolution images of COCO, we select

the semantic neighbors of each patch from a 112×112 dilated window (dilation ratio: 2) to reduce the computational cost. For long-range edge construction, the representations of per-patch global-context virtual nodes are computed by a stack of depth-wise 2D convolutions with the accumulative receptive field as 31, and these virtual nodes are linked to their corresponding patches.

Training configurations. We follow Swin Transformer (Liu et al., 2021b) to adopt a multi-scale training strategy where the shorter side of an image is resized to $[480, 800]$, and the longer side is with length 1,333. An AdamW (Loshchilov & Hutter, 2017) optimizer (betas: [0.9, 0.999], weight decay: 0.05) with initial learning rate 1.0×10^{-4} is employed for model training. In the 1× schedule with 12 total epochs, the learning rate is decayed at the 9th and 11th epoch with the decay rate 0.1. In the 3× schedule with 36 total epochs, the learning rate is decayed at the 27th and 33rd epoch with the decay rate 0.1. The stochastic depth drop rate is set as 0.1, 0.2, 0.3 in 1× schedule and 0.25, 0.5, 0.5 in 3× schedule for EurNet-T/S/B, respectively. All models are trained with batch size 8 on 8 Tesla-V100-32GB GPUs (*i.e.*, one image per GPU). Our implementations are based on the mmdetection (Chen et al., 2019) framework.

D.3 More Experimental Setups on ADE20K Semantic Segmentation

Model configurations. The EurNet-T, EurNet-S and EurNet-B pre-trained on ImageNet-1K serve as the backbone of UperNet (Xiao et al., 2018) to perform semantic segmentation. The patch representations output by all four modeling stages serve as the inputs of the Feature Pyramid Network (FPN) Lin et al. (2017). For medium-range edge construction, each patch is connected with its semantic neighbors from a 144×144 dilated window (dilation ratio: 2). For long-range edge construction, we use a stack of depth-wise 2D convolutions with accumulative receptive field 31 to compute the representations of per-patch global-context virtual nodes, and we connect these virtual nodes with their corresponding patches.

Training configurations. All input images are resized to the resolution 512×512. We adopt an AdamW (Loshchilov & Hutter, 2017) optimizer (betas: [0.9, 0.999], weight decay: 0.01) to train the model for 160K iterations with the base learning rate 6.0×10^{-5}. All models are trained with batch size 16 on 8 Tesla-V100-32GB GPUs (*i.e.*, two images per GPU). Our implementations are based on the mmsegmentation (Contributors, 2020) framework.

D.4 More Experimental Setups on Protein Function Prediction

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.

Dataset details. 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. This dataset contains 15,550 training, 1,729 validation and 1,919 test proteins.
- **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. The GO benchmark dataset contains 29,898 training, 3,322 validation and 3,415 test proteins.

Model configurations. The backbone architecture of EurNet is described in Sec. 4.2.2. Based on this backbone, we append a three-layer MLP with the architecture $\mathrm{Linear}(C_{\mathrm{out}}, C_{\mathrm{out}}) \rightarrow \mathrm{ReLU} \rightarrow \mathrm{Linear}(C_{\mathrm{out}}, C_{\mathrm{out}}) \rightarrow \mathrm{ReLU} \rightarrow \mathrm{Linear}(C_{\mathrm{out}}, N_{\mathrm{task}})$ to predict the binary classification logits of all tasks simultaneously (C_{out}: the dimension of output protein representation; N_{task}: the number of binary classification tasks). We employ the binary cross entropy loss for model optimization.

Training configurations. An AdamW (Loshchilov & Hutter, 2017) optimizer (betas: [0.9, 0.999], weight decay: 0) is utilized to train the model for 200 epochs. 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).

E More Ablation Study

E.1 Effect of GRMP Components

In Tab. 8, we analyze the key components of GRMP by substituting or removing the original component. This part of ablation studies are conducted on ImageNet-1K classification with EurNet-T.

Table 8: Ablation study of the key components of GRMP on ImageNet-1K with EurNet-T.

Setting	#Params. (M)	FLOPs (G)	Throughput (imgs/s)	Top-1 Acc (%)
GRMP	28.8	4.6	530.3	82.3
GRMP (gating $\rightarrow$ addition)	28.8	4.6	530.3	$81.6_{(\downarrow 0.7)}$
GRMP ($\alpha_r(v) \rightarrow \lvert\mathcal{R}\rvert^{-1}$)	28.8	4.6	567.4	$81.9_{(\downarrow 0.4)}$
GRMP (*w/o* W^{in})	26.7	4.3	561.9	$81.7_{(\downarrow 0.6)}$
GRMP (*w/o* W^{out})	26.7	4.3	562.5	$81.5_{(\downarrow 0.8)}$

Effect of gating mechanism. In the first row of the second block, we study the importance of the gating mechanism in GRMP by substituting the Hadamard product in the step ⑤ of Eq. (3) with the addition. After such a change, the top-1 accuracy decays by 0.7%. This performance decay demonstrates that, by using the separable graph convolution scheme in GRMP, the gating operation is more suitable than addition for node representation update (in contrast to the additive node representation update of RGConv in Eq. (1)), which shares similar insights with the modulation mechanism in FocalNet (Yang et al., 2022).

Effect of node-adaptive relation weighting. In the second row of the second block, we replace GRMP's node-adaptive relation weighting operation with simply taking the mean over all relations. This change leads to a 0.4% drop of accuracy. This relation weighting operation helps the GRMP layer to adaptively aggregate the messages of different relations based on each node's status, which benefits the model performance.

Effect of pre-layer and post-layer node-wise channel aggregation. In the third and fourth rows of the second block, we respectively evaluate the model variants without W^{in} and W^{out}. Under these two settings, the model accuracy decays by 0.6% and 0.8%, respectively. Therefore, it is important to perform both pre-layer and post-layer node-wise channel aggregation in the GRMP layer.

E.2 Effect of Multi-range Edges for Protein Structure Modeling

Tab. 9 shows the performance of EurNet on the EC function prediction benchmark by using different ranges of edges. When a single range of edges are employed, the model with short-range edges obtains the highest $\mathrm{F}_{\max}$ score 0.750. This result illustrates the importance of capturing short-range interactions for protein structure modeling, which coincides with the fact that many short-range interactions (*e.g.*, peptide and hydrogen bonds) contribute to the formation of protein structures. By adding long-range edges, the model performance is improved to 0.760, where the extra modeling of long-range interactions (*e.g.*, hydrophobic interactions) contributes to this improvement. By using all three ranges of edges, the full model of EurNet achieves the best $\mathrm{F}_{\max}$ score 0.768, which demonstrates the necessity of capturing short-, medium- and long-range interactions for protein structure modeling.

Table 9: Ablation study of multi-range edges on EC with EurNet.

short	medium	long	$\mathrm{F}_{\max}$
✓			**0.750**
	✓		0.708
		✓	0.647
✓	✓		0.755
✓		✓	**0.760**
	✓	✓	0.720
✓	✓	✓	**0.768**

E.3 EFFECT OF GRMP FOR PROTEIN STRUCTURE MODELING

Table 10: Ablation study of multi-relational modeling layer on EC with EurNet.

Layer	Hidden Dimension	Throughput (proteins/s)	F_{max}
RGConv	422	34.4	0.752
GRMP	512	34.6	**0.768**
RGConv	512	31.2	0.767
GRMP	592	31.5	**0.780**

In Tab. 10, we compare between RGConv and GRMP under the comparable throughput (*i.e.*, the number of proteins that the model can process in one second). All experiments are performed on EC with EurNet. (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 F_{max} score 0.752 is lower than GRMP's 0.768. (2) We then increase RGConv's hidden dimension to 512. At this time, RGConv achieves the F_{max} score 0.767 which is comparable to GRMP's performance under the same dimension, while its throughput is decreased by 3.2. Under the comparable throughput, GRMP can have the hidden dimension of 592, which leads to a higher F_{max} score 0.780. These results demonstrate that GRMP owns a better efficiency-performance trade-off than RGConv on protein structure modeling.