
Multi-Scale Protein Structure Modelling with Geometric Graph U-Nets

Chang Liu*

University of Cambridge
c1962@cantab.ac.uk

Vivian Li*

University of Cambridge
vd121@cam.ac.uk

Linus Leong

University of Cambridge
linus.leong.yh@gmail.com

Vladimir Radenkovic

University of Cambridge
vr375@cam.ac.uk

Pietro Liò

University of Cambridge
pl219@cam.ac.uk

Chaitanya K. Joshi

University of Cambridge
ckj24@cam.ac.uk

Abstract

Geometric Graph Neural Networks (GNNs) and Transformers have become state-of-the-art for learning from 3D protein structures. However, their reliance on local message passing prevents them from capturing the hierarchical interactions that govern protein function, such as global domains and long-range allosteric regulation. In this work, we argue that the network architecture itself should mirror this biological hierarchy. We introduce Geometric Graph U-Nets, a new class of models that learn multi-scale representations by recursively coarsening and refining the protein graph. We prove that this hierarchical design is theoretically as or more expressive than standard Geometric GNNs. Empirically, on the task of protein fold classification, Geometric U-Nets substantially outperform invariant and equivariant baselines, demonstrating their ability to learn the global structural patterns that define protein folds. Our work provides a principled blueprint for designing next-generation geometric architectures that can learn the multi-scale structure of biomolecules.

1 Introduction

Proteins are not just bags of atoms; they are exquisitely organized hierarchical structures, where local motifs like alpha-helices and beta-sheets assemble into functional domains, which in turn form the global quaternary structure [1]. This multi-scale organization is critical for protein function, as local interactions (e.g., hydrogen bonds) combine with long-range effects (e.g., allosteric regulation) to determine how a protein behaves in its biological context [2]. Standard geometric GNNs generally rely on local, fixed-radius message passing [3] and thus overlook the hierarchical, compositional organisation of proteins. Existing fixes such as virtual nodes [4] or ad hoc long-range edges [5, 6] are heuristic and fail to provide a principled, continuous multi-scale representation of protein structure.

Here, we introduce the Geometric Graph U-Net, a novel architectural blueprint that explicitly models the multi-scale nature of proteins by integrating geometric pooling layers within a hierarchical U-Net structure [7]. Further, we provide a rigorous theoretical foundation for this approach, extending the Geometric Weisfeiler-Leman (GWL) test [8] to prove that our hierarchical architecture is strictly as or more expressive than standard message-passing GNNs. Finally, we empirically evaluate our model’s performance on protein fold classification, where Geometric U-Nets consistently improve performance over invariant and equivariant baselines.

*These authors contributed equally to this work

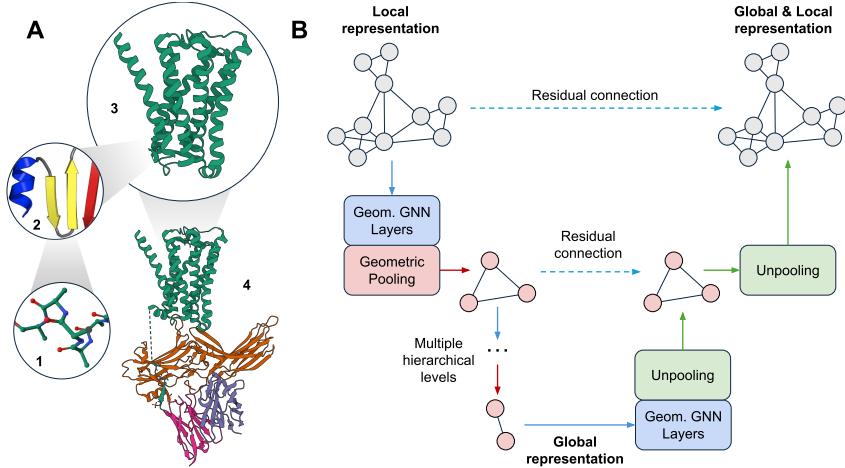


Figure 1: Geometric Graph U-Nets for multi-scale protein representation learning. (A) Protein function is governed by a structural hierarchy. Primary structures (1) organise into local secondary structures (2) that fold into tertiary domains (3), which assemble into quaternary complexes (4) to perform complex functions like allosteric signalling. Standard GNNs struggle to capture these long-range interactions. (B) The Geometric Graph U-Net is a biologically inspired architecture that encodes a protein graph by progressive coarsening, moving from local representations to global ones. The decoder then refines this information, using residual connections to re-inject high-resolution local information at each level. Thus, the final representation integrates both global context and local geometric details.

2 Background

Geometric Graph Neural Networks (GNNs) have become the standard toolkit for representation learning of 3D atomic systems such as biomolecules [9]. Duval et al. [10] offers a comprehensive review of such models and the different symmetries that are encodable in different message-passing GNN layers. Motivated by Joshi et al. [8], we mainly consider two classes of geometric GNN architectures: using invariant [11–14] or equivariant [15–17] geometric GNN layers, with further details in Appendix B.1. Geometric GNNs excel in capturing local relationships between nodes, but they often struggle to model global relations, usually adding more layers to enlarge the receptive field. This enlarged receptive field may cause *over-squashing*, as fixed-size feature vectors cannot encode the added information [18], limiting model expressivity [19]. In this paper, we will solve this problem through hierarchical GNNs.

Hierarchical Geometric GNNs enhance global representation by reducing node distances and enlarging the receptive field. Prior approaches fall into three main categories: (1) Local pooling clusters nodes at each GNN layer to form new nodes for the next layer, with approaches [20–25] ranging from basic clustering to learnable and attention-based adaptive pooling. Our research extends the benefits of pooling via a U-Net, of which the effectiveness has been demonstrated in prior studies [26–28]. (2) Virtual nodes create artificial connections between distant nodes and can embed domain knowledge in edge construction, with methods [29, 30] decomposing protein graphs into virtual supernodes for structural motifs. In contrast, we explore multi-layer hierarchies to model both intrinsic structures and global interactions. (3) Virtual edges enable information flows across distant residues, with some approaches [5] adding random long-range connections, whereas our method builds edges by sampling nearby nodes, reducing randomness and forming long-range links between structurally similar nodes. See Appendix B.2 for detailed citations.

GNN Expressivity formally characterizes how architectural ideas influence the class of functions a GNN can represent [31–33]. Joshi et al. [8] proposes the Geometric Weisfeiler-Leman (GWL) test, a geometric adaptation of the Weisfeiler-Leman test that measures expressivity of Geometric GNNs by their ability to distinguish geometric graphs while preserving 3D symmetries. See Appendix B.3 for further details on GWL and Invariant GWL (IGWL).

3 Geometric Graph U-Nets

Our goal in designing a geometric pooling operation is threefold. The operation must be (1) geometrically meaningful, preserving the 3D structure and respecting physical symmetries; (2) computationally efficient, allowing for the construction of deep, multi-level hierarchies; and (3) architecturally compatible with both invariant and equivariant Geometric GNN layers.

3.1 Geometric Pooling Layers

To define the geometric pooling layers, we follow the Select-Reduce-Connect (SRC) framework introduced in Grattarola et al. [34] as combination of three functions: *selection* (SEL), *reduction* (RED), and *connection* (CON), where $\text{POOL} = (\text{SEL}, \text{RED}, \text{CON})$. Below, we summarise two geometric pooling layers, detailed further in Appendix C.

Point Pooling Layers adapt the pooling operation used for point cloud data in PointNet++ [21] to protein structures. This layer forms supernodes by assigning the immediate 1-hop neighbourhood of each supernode to its cluster. As the reduction function, we use the GIN message-passing layer [32] for scalar features \mathbf{S} , while a summation aggregation of neighbouring nodes’ is applied for vector features $\vec{\mathbf{V}}$, ensuring invariance and equivariance to $\text{SE}(3)$ transformations. Point Pooling creates supernodes that represent local structural motifs by aggregating information from their immediate 1-hop neighborhood, akin to identifying a patch of the protein surface.

$$\mathbf{C}_{j,i} = \begin{cases} 1 & \text{if } i \in \mathcal{N}_j \\ 0 & \text{otherwise} \end{cases}, \quad \mathbf{S} = \text{MLP}(\mathbf{C}^\top \mathbf{S}), \quad \vec{\mathbf{V}} = \mathbf{C}^\top \vec{\mathbf{V}},$$

where \mathbf{C} is the cluster assignment matrix, j indexes each pooled supernode, and i indexes the nodes in the neighborhood \mathcal{N}_j of the j -th supernode, and MLP denotes a fully-connected layer.

Sparse Pooling Layers assign each node to its nearest supernode, satisfying the second condition of Theorem 4.1. The cluster assignment matrix of this layer is sparse, with each row containing a single non-diagonal, non-zero entry (set as 1) indicating supernode affiliation. Scalar and vector features are updated with a consistent reduction function. This layer is termed *Sparse* due to the constant $O(1)$ cardinality of each supernode. Sparse Pooling creates a sparser graph, effectively partitioning the structure into distinct, non-overlapping regions.

$$\mathbf{C}_{j,i} = \begin{cases} 1 & \text{if } i = \arg \min_{k \neq j} \|\mathbf{x}_j - \mathbf{x}_k\|_2 \text{ or } i = j \\ 0 & \text{otherwise} \end{cases}, \quad \mathbf{S} = \mathbf{C}^\top \mathbf{S}, \quad \vec{\mathbf{V}} = \mathbf{C}^\top \vec{\mathbf{V}},$$

where \mathbf{x} denotes the feature vector of each node, and $\|\cdot\|_2$ computes the Euclidean distance.

3.2 U-Nets with Geometric Pooling Layers

We follow the framework in Jamasb et al. [35], using a graph encoder to generate graph embeddings and extending it with a U-Net architecture of pooling and unpooling blocks. For the pooling blocks, we adopt the original encoder’s six message-passing layers, inserting a geometric pooling layer after every two message-passing layers. Each pooling layer clusters nodes into supernodes that connect longer-distance neighbours.

Formally, given an input graph \mathcal{G}^{i-1} with $|\mathcal{V}^{i-1}| = N$ nodes, the i -th pooling block outputs a pooled graph \mathcal{G}^i with $|\mathcal{V}^i| = K \leq N$, where $\mathcal{G}^i = \text{POOL}(\mathcal{G}^{i-1})$ with node features $(\mathbf{S}^i, \vec{\mathbf{V}}^i) = \text{GNN}(\mathcal{G}^i)$. POOL denotes the geometric pooling layer and GNN represents the message passing layer updating the scalar and vector features \mathbf{S}^i and $\vec{\mathbf{V}}^i$ of the pooled graph \mathcal{G}^i .

Each unpooling block expands the previously unpooled features through an unpooling layer and a residual connection. As supernodes still retain geometric information, the new unpooled graph is constructed by first reintroducing removed nodes with learnable scalar and vector features initialised to zeros, and then concatenating cached features through a skip connection from corresponding pooling blocks to this unpooled output.

Formally, given an input graph \mathcal{G}^{j-1} with $|\mathcal{V}^{j-1}| = K$ nodes, the j -th unpooling block outputs an unpooled graph \mathcal{G}^j with $|\mathcal{V}^j| = N \geq K$, where $\mathcal{G}^j = \text{UNPOOL}(\mathcal{G}^{j-1})$. As supernodes retain physical

information, the unpooled graph is reconstructed by reintroducing removed nodes with learnable scalar and vector features initialised to zeros, with a skip connection concatenating cached features from the corresponding pooling blocks to the unpooling output. An overview of this hierarchical U-Net architecture is shown in Figure 1.

4 Expressive Power of Geometric Pooling

We study expressivity through the lens of distinguishing non-isomorphic geometric graphs, adopting the Geometric Weisfeiler–Leman (GWL) test [8]. We then extend the hierarchical expressivity framework introduced for (non-geometric) pooling by Bianchi and Lachi [36] and Lachi et al. [37] to the geometric setting, enabling analysis of when pooling preserves or strengthens GWL (and IGWL) distinguishing capacity. See Appendix D for preliminary definitions.

Let $\mathcal{X}_{\mathcal{G}_i}^{k-(I)GWL} = \{\{(s_j^i, \vec{v}_j^i) : j \in \mathcal{V}\}\}$ be the multi-set of k -GWL-discriminative or IGWL-discriminative scalar and vector node features for a graph \mathcal{G}_i . From Proposition D.1, as long as $\text{RED} \circ \text{SEL}$ uniquely maps unpooled nodes to pooled nodes, POOL will maintain expressivity. We expand this to geometric graphs:

Theorem 4.1. *Let $\mathcal{G}_1 = (\mathcal{A}_1, \mathbf{S}_1, \vec{\mathbf{V}}_1, \vec{\mathcal{X}}_1)$ and $\mathcal{G}_2 = (\mathcal{A}_2, \mathbf{S}_2, \vec{\mathbf{V}}_2, \vec{\mathcal{X}}_2)$ be geometric graphs with $|\mathbf{V}_1| = |\mathbf{V}_2| = N$ such that $\mathcal{G}_1 \neq_{k-(I)GWL} \mathcal{G}_2$. Let $\text{POOL} = (\text{SEL}, \text{RED}, \text{CON})$ be a graph pooling layer placed after a block of L message-passing layers such that $\mathcal{G}_{1P} = \text{POOL}(\mathcal{G}_1^L)$ and $\mathcal{G}_{2P} = \text{POOL}(\mathcal{G}_2^L)$ with $|\mathbf{V}_{1P}| = |\mathbf{V}_{2P}| = K$. Let $\mathbf{S}_{1/2}^L, \vec{\mathbf{V}}_{1/2}^L \in \mathbb{R}^{N \times d}$ and $\mathbf{S}_{1/2}^P, \vec{\mathbf{V}}_{1/2}^P \in \mathbb{R}^{K \times f}$ be the node features before and after POOL respectively.*

Then, \mathcal{G}_{1P} and \mathcal{G}_{2P} will be k -(I)GWL-distinguishable if the following conditions hold:

1. $\sum_i^N s_{1_i}^L \neq \sum_i^N s_{2_i}^L$ or $\sum_i^N \vec{v}_{1_i}^L \neq \sum_i^N \vec{v}_{2_i}^L$
2. *The memberships generated by SEL satisfy $\sum_j^K c_{ij} = \lambda$, with $\lambda > 0$ for each node i , i.e., the cluster assignment matrix \mathbf{C} is a right stochastic matrix up to the global constant λ*
3. *The function RED is of type $\text{RED} : (\mathbf{S}^L, \vec{\mathbf{V}}^L, \mathbf{C}) \mapsto (\mathbf{S}^P = \mathbf{C}^T \mathbf{S}^L, \vec{\mathbf{V}}^P = \mathbf{C}^T \vec{\mathbf{V}}^L)$*

Theorem 4.1 gives sufficient conditions ensuring pooling preserves the GNN’s ability to distinguish graphs: if two graphs are distinguishable pre-pooling, they remain so post-pooling. Thus, hierarchical coarsening does not destroy discriminative information. The requirements are: (1) the pre-pooling message passing yields distinct multisets for distinguishable inputs, (2) every original node is assigned (right-stochastic membership up to constant λ), and (3) RED applies convex combination $(\mathbf{S}^P, \vec{\mathbf{V}}^P) = \mathbf{C}^T(\mathbf{S}^L, \vec{\mathbf{V}}^L)$. These guarantees are independent of CON. See Appendix D for a full proof.

For increasing expressivity (see Theorem D.1), any SEL that separates previously k -(I)GWL-indistinguishable graphs can be paired with an injective RED to form a POOL that strictly improves distinguishing power. Section 5.2 empirically validates this on k -chains.

5 Experimental Results

5.1 Fold Classification

We evaluate on protein fold classification to demonstrate the effectiveness of our proposed Geometric Graph U-Net. SchNet [11] and GVP [15] serve as baselines, tuned with decoders with fewer hidden layers but higher performance. For both models, we construct the input protein graphs using a one-hot encoding of amino acids, along with a positional encoding of the protein backbone and virtual backbone torsions and angles as features. Then, we introduce U-Net variants with pooling and unpooling layers. We employ the SCOP 1.75 dataset [38] for model training on a single NVIDIA RTX 4090 with up to 150 epochs using Adam and learning rates in $[0.001, 0.01]$. Evaluation follows the Fold Prediction task [35], classifying proteins into 1,195 folds. Results are reported on three test subsets (Fold, Family, Superfamily) as micro-averaged accuracy and F1 score over three runs.

Table 1: **Benchmarking Geometric Graph U-Nets on protein fold classification.** The best results are highlighted in **bold**, with improved performance from the baseline indicated in green.

Model	Architecture	Params	Accuracy			F1		
			Fold	Superfamily	Family	Fold	Superfamily	Family
SchNet	Baseline	5.23M	0.331 \pm 0.014	0.442 \pm 0.012	0.928 \pm 0.013	0.087 \pm 0.008	0.157 \pm 0.005	0.688 \pm 0.037
	U-Net Point Pool	5.23M	0.314 \pm 0.007	0.436 \pm 0.014	0.937 \pm 0.007	0.081 \pm 0.001	0.156 \pm 0.014	0.715 \pm 0.033
	U-Net Sparse Pool	5.23M	0.340 \pm 0.010	0.461 \pm 0.002	0.948 \pm 0.003	0.095 \pm 0.007	0.170 \pm 0.004	0.759 \pm 0.018
GVP	Baseline	3.60M	0.336 \pm 0.024	0.512 \pm 0.021	0.961 \pm 0.005	0.097 \pm 0.010	0.215 \pm 0.010	0.793 \pm 0.013
	U-Net Point Pool	3.06M	0.404 \pm 0.002	0.514 \pm 0.007	0.971 \pm 0.003	0.119 \pm 0.002	0.206 \pm 0.001	0.849 \pm 0.010
	U-Net Sparse Pool	3.06M	0.373 \pm 0.015	0.529 \pm 0.009	0.966 \pm 0.004	0.111 \pm 0.005	0.217 \pm 0.004	0.830 \pm 0.019

As shown in Table 1, our Geometric U-Net models provide substantial and consistent performance gains across both invariant (SchNet) and equivariant (GVP) backbones. For instance, the GVP U-Net with Point Pooling (U-Net Point Pool) improves the Superfamily F1 score from 0.793 to 0.849 with few additional parameters, a significant gain on this task. This demonstrates that the hierarchical architecture, rather than the specifics of the message-passing layer, is the primary driver of the improved ability to capture global structural information. For a fairer comparison, the U-Net variants remove one decoder layer to match the baseline parameter numbers. Despite the lighter decoders, they still achieve performance gains, demonstrating the efficacy of our approach.

5.2 Synthetic Experiment on Expressivity: k-chains

Table 2 shows how different pooling layers enhance expressivity on synthetic k-chain graphs from Joshi et al. [8]. We report averages over 10 runs for $k = 4$ and assess pooling effectiveness on SchNet [11], SphereNet [39], TFN [40], GVP-GNN [15], and EGNN [17]. The GWL test requires $(\lfloor \frac{k}{2} \rfloor + 1)$ iterations to differentiate k-chains. Our pooling procedure essentially reduces k by merging the central nodes and adding edges between endpoint second-order neighbours. As expected, with pooling, we see all models distinguish k-chains with fewer layers than the minimal requirement. This is a simple yet effective demonstration of Theorem D.1, showing that pooling can enhance expressivity by enabling the model to distinguish geometric graphs it previously could not.

Table 2: **k-chain geometric graphs.** k -chains are $(\lfloor \frac{k}{2} \rfloor + 1)$ -hop distinguishable and $(\lfloor \frac{k}{2} \rfloor + 1)$ GWL iterations are theoretically sufficient to distinguish them. Anomalous results are marked in red and expected results in green.

(k = 4-chains)		Number of layers				
GNN Layer		$\lfloor \frac{k}{2} \rfloor$	$\lfloor \frac{k}{2} \rfloor + 1$	$\lfloor \frac{k}{2} \rfloor + 2$	$\lfloor \frac{k}{2} \rfloor + 3$	$\lfloor \frac{k}{2} \rfloor + 4$
Equivariant	GWL	50%	100%	100%	100%	100%
	E-GNN	50.0 \pm 0.0	50.0 \pm 0.0	50.0 \pm 0.0	50.0 \pm 0.0	100.0 \pm 0.0
	E-GNN - 3 nodes	100.0 \pm 0.0				
	E-GNN - 4 nodes	100.0 \pm 0.0				
	GVP-GNN	50.0 \pm 0.0	90.0 \pm 20.0	80.0 \pm 24.5	100.0 \pm 0.0	90.0 \pm 20.0
	GVP-GNN - 3 nodes	100.0 \pm 0.0	95.0 \pm 15.0	100.0 \pm 0.0	90.0 \pm 20.0	100.0 \pm 0.0
	GVP-GNN - 4 nodes	100.0 \pm 0.0	95.0 \pm 15.0	100.0 \pm 0.0	100.0 \pm 0.0	100.0 \pm 0.0
	TFN	50.0 \pm 0.0				
	TFN - 3 nodes	100.0 \pm 0.0				
	TFN - 4 nodes	100.0 \pm 0.0				
Invariant	IGWL	50%	50%	50%	50%	50%
	SchNet	50.0 \pm 0.0				
	SchNet - 3 nodes	80.0 \pm 24.5	95.0 \pm 15.0	90.0 \pm 20.0	95.0 \pm 15.0	95.0 \pm 15.0
	SchNet - 4 nodes	85.0 \pm 22.9	95.0 \pm 15.0	95.0 \pm 15.0	100.0 \pm 0.0	95.0 \pm 15.0
	SphereNet	50.0 \pm 0.0				
	SphereNet - 3 nodes	100.0 \pm 0.0				
	SphereNet - 4 nodes	100.0 \pm 0.0				

6 Conclusion

In this work, we demonstrated that the architectural design of Geometric GNNs should reflect the inherent hierarchical nature of the biological structures they model. Our Geometric Graph U-Net, provably at least as expressive as standard Geometric GNNs and empirically more powerful, serves as a flexible blueprint for a new generation of multi-scale models, extendable with different supernode selection or reduction methods, like the one introduced in Zhang et al. [41]. By treating proteins not as flat graphs of atoms, but as nested structural assemblies, we show how multi-scale architectures can yield more powerful representations for predicting function from structure.

References

- [1] Jane S. Richardson. The anatomy and taxonomy of protein structure. In C.B. Anfinsen, John T. Edsall, and Frederic M. Richards, editors, *Advances in Protein Chemistry*, volume 34 of *Advances in Protein Chemistry*, pages 167–339. Academic Press, 1981.
- [2] Hesam N. Motlagh, James O. Wrabl, Jing Li, and Vincent J. Hilser. The ensemble nature of allostery. *Nature*, 508(7496):331–339, Apr 2014. ISSN 1476-4687.
- [3] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212, 2017.
- [4] Florian Sestak, Lisa Schneckenreiter, Johannes Brandstetter, Sepp Hochreiter, Andreas Mayr, and Günter Klambauer. Vn-egnn: E (3)-equivariant graph neural networks with virtual nodes enhance protein binding site identification. *arXiv preprint arXiv:2404.07194*, 2024.
- [5] John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023.
- [6] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishabh Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zieliński, Martin Steinegger, Michałina Pacholska, Tamás Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 2021.
- [7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III* 18, pages 234–241. Springer, 2015.
- [8] Chaitanya K Joshi, Cristian Bodnar, Simon V Mathis, Taco Cohen, and Pietro Lio. On the expressive power of geometric graph neural networks. In *International Conference on Machine Learning*. PMLR, 2023.
- [9] Xuan Zhang, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, Meng Liu, Yuchao Lin, Zhao Xu, Keqiang Yan, et al. Artificial intelligence for science in quantum, atomistic, and continuum systems. *Foundations and Trends® in Machine Learning*, 2025.
- [10] Alexandre Duval, Simon V. Mathis, Chaitanya K. Joshi, Victor Schmidt, Santiago Miret, Fragkiskos D. Malliaros, Taco Cohen, Pietro Liò, Yoshua Bengio, and Michael Bronstein. A hitchhiker’s guide to geometric gnns for 3d atomic systems, 2024.
- [11] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.
- [12] Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining, 2023.
- [13] Johannes Gasteiger, Shankari Giri, Johannes T. Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules, 2022.
- [14] Hehe Fan, Zhangyang Wang, Yi Yang, and Mohan Kankanhalli. Continuous-discrete convolution for geometry-sequence modeling in proteins. In *The Eleventh International Conference on Learning Representations*, 2022.

- [15] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2020.
- [16] Alex Morehead and Jianlin Cheng. Geometry-complete perceptron networks for 3D molecular graphs. *Bioinformatics*, 40(2):btae087, 02 2024. ISSN 1367-4811.
- [17] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.
- [18] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications, 2021.
- [19] Francesco Di Giovanni, Lorenzo Giusti, Federico Barbero, Giulia Luise, Pietro Lio, and Michael M. Bronstein. On over-squashing in message passing neural networks: The impact of width, depth, and topology. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 7865–7885. PMLR, 23–29 Jul 2023.
- [20] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.
- [21] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.
- [22] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies, 2022.
- [23] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling, 2019.
- [24] Ekagra Ranjan, Soumya Sanyal, and Partha Pratim Talukdar. Asap: Adaptive structure aware pooling for learning hierarchical graph representations, 2020.
- [25] Ning Liu, Songlei Jian, Dongsheng Li, Yiming Zhang, Zhiqian Lai, and Hongzuo Xu. Hierarchical adaptive pooling by capturing high-order dependency for graph representation learning, 2021.
- [26] Xiaodan Hu, Mohamed A Nael, Alexander Wong, Mark Lamm, and Paul Fieguth. Runet: A robust unet architecture for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.
- [27] Yuanhong Jiang, Qiaoqiao Ding, Yu Guang Wang, Pietro Lio, and Xiaoqun Zhang. Vision graph u-net: Geometric learning enhanced encoder for medical image segmentation and restoration. *Inverse Problems and Imaging*, 18(3):672–689, 2024.
- [28] Yuhe Shen, Jiang Li, Weifang Zhu, Kai Yu, Meng Wang, Yuanyuan Peng, Yi Zhou, Liling Guan, and Xinjian Chen. Graph attention u-net for retinal layer surface detection and choroid neovascularization segmentation in oct images. *IEEE Transactions on Medical Imaging*, 42(11):3140–3154, 2023.
- [29] Xuan Zang, Xianbing Zhao, and Buzhou Tang. Hierarchical molecular graph self-supervised learning for property prediction. *Communications Chemistry*, 6(1):34, Feb 2023. ISSN 2399-3669.
- [30] Zhonghui Gu, Xiao Luo, Jiaxiao Chen, Minghua Deng, and Luhua Lai. Hierarchical graph transformer with contrastive learning for protein function prediction. *Bioinformatics*, 2023.
- [31] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *International conference on machine learning*, 2017.

- [32] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.
- [33] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *AAAI*, 2019.
- [34] Daniele Grattarola, Daniele Zambon, Filippo Maria Bianchi, and Cesare Alippi. Understanding pooling in graph neural networks. *CoRR*, abs/2110.05292, 2021.
- [35] Arian Rokkum Jamasb, Alex Morehead, Zuobai Zhang, Chaitanya K. Joshi, Kieran Didi, Simon V Mathis, Charles Harris, Jian Tang, Jianlin Cheng, Pietro Lio, and Tom Leon Blundell. Evaluating representation learning on the protein structure universe. In *The Twelfth International Conference on Learning Representations*, 2024.
- [36] Filippo Maria Bianchi and Veronica Lachi. The expressive power of pooling in graph neural networks, 2023.
- [37] Veronica Lachi, Alice Moallemi-Oureh, Andreas Roth, and Pascal Welke. Graph pooling provably improves expressivity. In *NeurIPS 2023 Workshop: New Frontiers in Graph Learning*, 2023.
- [38] Jie Hou, Badri Adhikari, and Jianlin Cheng. Deepsf: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, 2018.
- [39] Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d molecular graphs. In *International Conference on Learning Representations*, 2021.
- [40] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.
- [41] Zhang Zhang, Arsham Ghavasieh, Jiang Zhang, and Manlio De Domenico. Coarse-graining network flow through statistical physics and machine learning. *Nature Communications*, 2025.

A Author Contributions

This work originated as team projects at the University of Cambridge MPhil/Part III course on Geometric Deep Learning (L65). We summarise author contributions as follows:

Chang Liu and **Vivian Li** served as equal contributing first authors, driving the final research and publication efforts after the L65 course. CL was the overall project lead and provided GPU resources for the final experiments. CL reproduced the initial results from VR and CL’s L65 project, and developed an improved version of the U-Net architecture proposed by VR and CL. CL wrote the final Background, Method and Experiment sections, and VL wrote the final Abstract, Introduction, Theory, and Conclusion, all of which were based on the initial L65 project report by CL and VR.

Linus Leong contributed to the final protein structure experiments, including validating baselines and hyperparameter tuning. LL contributed to writing the final Experiments section.

Vladimir Radenkovic primarily contributed to the L65 project with CL, which provided the structural foundation, early results, and written material for the final manuscript. VR co-developed the geometric pooling, Sparse Pooling and Point Pooling layers, and U-Net framework as well as their expressivity theory. VR wrote the proofs. VR implemented the initial models and experimental code used for both the protein structure experiments and the k-chains task. VR designed the k-chains experiment.

Pietro Liò and **Chaitanya Joshi** conceived the project, supervised the research, and contributed to writing the final manuscript.

B Background

B.1 Invariant and Equivariant GNN Layers

We define a geometric graph $\mathcal{G} = (\mathcal{A}, \mathbf{S}, \vec{\mathbf{V}}, \vec{\mathcal{X}})$ as an attributed graph with adjacency matrix \mathbf{A} of size $N \times N$, scalar features $\mathbf{S} \in \mathbb{R}^{N \times f}$ and geometric attributes: node coordinates $\vec{\mathcal{X}} \in \mathbb{R}^{N \times d}$ and optional vector features $\vec{\mathbf{V}} \in \mathbb{R}^{N \times d}$. Feature dimensions f and d represent the scalar feature channel dimensions and the Euclidean space dimensions, respectively. We say two geometric graphs are *geometrically isomorphic* if the underlying attributed graphs, represented by an adjacency matrix with scalar features, are isomorphic and their geometric attributes are equivalent, up to global group actions like rotation and reflection.

Motivated by Joshi et al. [8], we mainly consider two broad classes of geometric GNN architectures: invariant [11–14] and equivariant [15–17] geometric GNN layers.

Invariant GNN layers only propagate invariant scalar features computed using geometric information within local neighbourhoods, following the equation below:

$$\mathbf{s}_i^{(t+1)} := \text{UPD}\left(\mathbf{s}_i^{(t)}, \text{AGG}\left(\{(\mathbf{s}_i^{(t)}, \vec{\mathbf{v}}_i^{(t)}), (\mathbf{s}_j^{(t)}, \vec{\mathbf{v}}_j^{(t)}), \vec{\mathbf{x}}_{ij} \mid j \in \mathcal{N}_i\}\right)\right).$$

Equivariant GNN layers update both scalar and geometric vector features from iteration t to $t + 1$ via learnable aggregate and update functions, AGG and UPD, respectively, following the equation below:

$$\mathbf{s}_i^{(t+1)}, \vec{\mathbf{v}}_i^{(t+1)} := \text{UPD}\left((\mathbf{s}_i^{(t)}, \vec{\mathbf{v}}_i^{(t)}), \text{AGG}\left(\{(\mathbf{s}_i^{(t)}, \vec{\mathbf{v}}_i^{(t)}), (\mathbf{s}_j^{(t)}, \vec{\mathbf{v}}_j^{(t)}), \vec{\mathbf{x}}_{ij} \mid j \in \mathcal{N}_i\}\right)\right),$$

where $\vec{\mathbf{x}}_{ij} = \vec{\mathbf{x}}_i - \vec{\mathbf{x}}_j$ denote relative position vectors.

As illustrated in the equations above, each geometric GNN layer aggregates information from a node’s neighbourhood, meaning information in GNNs spreads in terms of *k-hop receptive fields*, where k is the number of GNN layers.

B.2 Hierarchical Geometric GNN Approaches

Local pooling approaches cluster nodes at each GNN layer to form new nodes for the next layer. PointNet models [20–22] use basic clustering for point clouds, while Ying et al. [23] develops learnable clustering, and Ranjan et al. [24] and Liu et al. [25] use attention for adaptive pooling.

Virtual nodes create artificial connections between distant nodes and can embed domain knowledge in edge construction. Zang et al. [29] decomposes protein graphs into virtual supernodes for structural motifs and the overall graph, while Gu et al. [30] similarly represent motifs with virtual supernodes.

Virtual edges enable information flows across distant residues. Ingraham et al. [5] add random long-range connections, whereas hierarchical methods can build edges by sampling nearby nodes, reducing randomness and forming long-range links between structurally similar nodes.

B.3 GNN Expressivity

The GWL test operates in iterations: in each iteration, graph nodes update their colour by aggregating both scalar and geometric information in its neighbourhood with \mathfrak{G} -orbit injective and \mathfrak{G} -invariant function over Lie groups $\mathfrak{G} = SO(d)$ or $\mathfrak{G} = O(d)$. Auxiliary geometric information variables are additionally updated by aggregating local geometric information around each node with injective and \mathfrak{G} -equivariant aggregation. The test then terminates when the colours of the nodes do not change from the previous iteration, and the two graphs are geometrically non-isomorphic if multi-sets of their node features are different. We establish definitions to lay the framework for our later theoretical work.

Definition B.1. *Graphs \mathcal{G}_1 and \mathcal{G}_2 are k -GWL distinguishable* $\mathcal{G}_1 \neq_{k\text{-GWL}} \mathcal{G}_1$ *if there exists an test iteration $i \leq k$ for which their multisets of node features are different.*

Joshi et al. [8] additionally defines the Invariant GWL (IGWL) test that only updates node colours using k -hop neighbourhood scalar and geometric information with a \mathfrak{G} -orbit injective and invariant function without updating the geometric representation of nodes.

Definition B.2. *Graphs \mathcal{G}_1 and \mathcal{G}_2 are IGWL distinguishable* $\mathcal{G}_1 \neq_{IGWL} \mathcal{G}_1$ *if there exists a test iteration i for which their multisets of node features are different.*

Building on GWL and IGWL, Joshi et al. [8] provides an upper bound on the expressive power of both invariant and equivariant GNNs by proving that equivariant GNNs are at most as powerful as GWL at distinguishing non-isomorphic geometric graphs, while invariant GNNs are limited by the IGWL at the same task. Thus, a theoretical framework for proving the expressivity of new GNN architectures can be established by showing the GWL- or IGWL-distinguishability of graphs under the new model.

Definition B.3. *Graphs \mathcal{G}_1 and \mathcal{G}_2 are currently GWL or IGW distinguishable* $\mathcal{G}_1 \neq_{C\text{-GWL/IGWL}} \mathcal{G}_2$ *if their multisets of node features are different at the current iteration i , according to the respective GWL or IGWL test.*

C Further Methodological Details

C.1 Preliminaries

To define the geometric pooling layers, we follow the Select-Reduce-Connect (SRC) framework introduced in Grattarola et al. [34], where POOL is the combination of three functions: *selection* (SEL), *reduction* (RED), and *connection* (CON). SEL clusters the nodes of the input graph into subsets called *supernodes*, producing cluster assignment matrix \mathbf{C} where c_{ij} is the membership score of node i to supernode j . RED creates the pooled node features by aggregating the features of the nodes assigned to the same supernode:

$$\text{RED}(\mathcal{G}, \mathbf{C}) \mapsto \mathbf{S}^P$$

where $|P|$ denotes node cardinality of the pooled graph \mathcal{G}_P . Finally, CON constructs new edges between supernodes:

$$\text{SEL}(\mathcal{G}, \mathbf{C}) \mapsto \mathbf{A}'$$

We outline the shared components for these functions among all of our proposed pooling layers below:

Selection (SEL): Inspired by PointNet++ [21] and recognizing that each graph node corresponds to physical coordinates, we employ farthest point sampling (FPS) for graph coarsening, where sampled nodes $\mathcal{V}_P = \{v_1, \dots, v_K\}$ define the centres of supernode clusters $\mathbf{C} = \mathbf{C}_1, \dots, \mathbf{C}_K$, while the rest of the nodes are assigned to one or more supernodes based on proximity. By having supernodes adopt

the coordinates of their central nodes, we ensure that the physical interpretability and properties of the coarsened graph are preserved. This method effectively sparsifies the graph by selectively thinning dense protein representations. We empirically set the sampling ratio for FPS in each pooling layer at 0.6.

Reduction (RED): The reduction function (RED) creates the pooled node features by aggregating the features of the vertices assigned to the same supernode. We extend reduction function RED to update node coordinates $\vec{\mathcal{X}} \in \mathbb{R}^{N \times d}$ and vector features $\vec{\mathbf{V}} \in \mathbb{R}^{N \times v}$:

$$\text{RED}(\mathcal{G}, \mathbf{C}) \mapsto (\mathbf{S}_P, \vec{\mathbf{V}}_P, \vec{\mathcal{X}}_P)$$

Connection (CON): The edge generation process in coarsened graphs is guided by the need for pooled graphs to retain regular structure based on the proximity of nodes. To achieve this, we adopt the K-nearest neighbours (K-NN) technique, with a fixed value of $k = 16$ in our experiments, to form edges between supernodes, just as the original graphs in the benchmark datasets are constructed.

This design of SEL and CON allows for flexible, controlled, and adaptive graph size reduction and edge set design. This way, we aim to construct geometric pooling layers specifically designed for graphs structured from point clouds, as often used in biochemistry and material science applications.

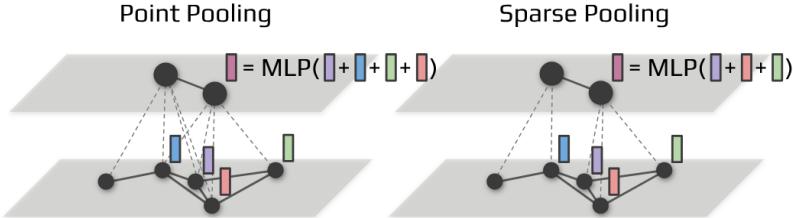


Figure 2: Comparison of point vs. sparse pooling receptive fields.

C.2 Point Pooling Layer

For each pair of nodes $v_i \notin \mathcal{V}_P$ and supernodes v_j^{sn} , where v_i denotes the i-th node, \mathcal{V}_P stands for the sampled node set, and v_j^{sn} is the j-th supernode, we define the cluster assignment matrix \mathbf{C} , as the following:

$$\mathbf{C}_{j,i} = \begin{cases} 1 & \text{if } i \in \mathcal{N}_j \\ 0 & \text{otherwise} \end{cases}$$

As a reduction function for scalar features, we use the message-passing layer from GIN [32] to update the features of each pooled supernode $j \in \mathbf{C}$:

$$\begin{aligned} s_j^{sn} &= \text{MLP}(\mathbf{C}_j^\top \mathbf{s}) \\ &= \text{MLP}(s_j + \sum_{i \in \mathcal{N}_j} s_i), \end{aligned}$$

where \mathbf{C}_j represents row j in the cluster assignment matrix. For the reduction of vector features, we find that a summation aggregation of neighbouring nodes' vector features of supernodes, without introducing non-linear transformations, provides the most effective results. For each pooled supernode $j \in \mathbf{C}$ we define reduction as:

$$\begin{aligned} \vec{\mathbf{v}}_j^{sn} &= \mathbf{C}_j^\top \vec{\mathbf{V}} \\ &= \vec{\mathbf{v}}_j + \sum_{i \in \mathcal{N}_j} \vec{\mathbf{v}}_i, \end{aligned}$$

which ensures that updated scalar and vector features of coarsened point clouds remain invariant and equivariant to both $\text{SO}(3)$ transformations. This pooling layer efficiently extracts representations of subgraphs surrounding each sampled node within the original graphs. In this way, supernodes

encapsulate not only information about individual nodes they physically represent, but also about the broader molecular regions around them. While this layer possesses an injective reduction function for scalar features, the combined selection and reduction function ($\text{SEL} \circ \text{RED}$) is not injective and thus does not fully satisfy the sufficient conditions required for maintaining expressivity. Nevertheless, the simplicity of this layer, along with our later proof with IGWL for its expressivity, makes it a compelling choice for an initial pooling layer.

C.3 Sparse Pooling Layer

For each pair of nodes $v_i \notin \mathcal{V}_P$ and supernodes $v_j^{sn} \in \mathcal{V}_P$ we define cluster assignment matrix \mathbf{C} , as the following:

$$\mathbf{C}_{j,i} = \begin{cases} 1 & \text{if } i = \arg \min_{k \neq j} \|\mathbf{x}_j - \mathbf{x}_k\|_2 \text{ or } i = j \\ 0 & \text{otherwise} \end{cases}$$

For the reduction function, we use the summation of assigned nodes to update the scalar and vector features of supernodes:

$$\begin{aligned} \mathbf{s}_j^{sn} &= \mathbf{C}_j^\top \mathbf{S} \\ \vec{\mathbf{v}}_j^{sn} &= \mathbf{C}_j^\top \vec{\mathbf{V}} \end{aligned}$$

This pooling layer is called *Sparse* because of the constant cardinality $O(1)$ of each supernode [34]. If preceding message passing layers of this pooling layer are expressive enough and satisfy first condition of theorem 4.1, this pooling layer maintains expressivity.

C.4 Analysis of GNN U-Net with Geometric Pooling Layers

These pooling blocks extract subgraph representations around each supernode, which represent specific physical segments of a protein, with local information aggregated in their feature representations. As the granularity of the graph decreases, there is an increased likelihood of connectivity between supernodes representing distant subgraphs. This connectivity enhances the flow of information across remote sections of the graph. By limiting the number of message-passing layers on each coarsened graph, we focus these layers on learning local features, while the broader, more complex interactions are learned in the subsequent encoder blocks following each pooling layer. Each pooling block in the pooling phase generates a graph embedding, which is cached for use in the unpooling phase.

D Theoretical Remarks for Section 4

Definition D.1. An invariant graph pooling layer $\text{POOL} = (\text{SEL}, \text{RED}, \text{CON})$ **maintains expressivity** if it maps any pair of currently IGWL-distinguishable graphs (CIGWL) to a pair of IGWL-distinguishable graphs, i.e., $\mathcal{G}_1 \neq_{\text{CIGWL}} \mathcal{G}_2 \Rightarrow \text{POOL}(\mathcal{G}_1) \neq_{\text{IGWL}} \text{POOL}(\mathcal{G}_2)$

Definition D.2. An equivariant graph pooling layer $\text{POOL} = (\text{SEL}, \text{RED}, \text{CON})$ **maintains expressivity** if there exists a k such that it maps any pair of currently GWL-distinguishable graphs (CGWL) to a pair of k -GWL-distinguishable graphs, i.e., $\mathcal{G}_1 \neq_{\text{CGWL}} \mathcal{G}_2 \Rightarrow \text{POOL}(\mathcal{G}_1) \neq_{k-\text{GWL}} \text{POOL}(\mathcal{G}_2)$ for any k

Definition D.3. An invariant graph pooling layer $\text{POOL} = (\text{SEL}, \text{RED}, \text{CON})$ **increases expressivity** if it maps any pair of IGWL-indistinguishable graphs to a pair of IGWL-distinguishable graphs, i.e., $\mathcal{G}_1 =_{\text{IGWL}} \mathcal{G}_2 \Rightarrow \text{POOL}(\mathcal{G}_1) \neq_{\text{IGWL}} \text{POOL}(\mathcal{G}_2)$

Definition D.4. An equivariant graph pooling layer $\text{POOL} = (\text{SEL}, \text{RED}, \text{CON})$ **increases expressivity** if it maps any pair of k -GWL-indistinguishable graphs to a pair of k -GWL-distinguishable graphs, i.e., $\mathcal{G}_1 =_{k-\text{GWL}} \mathcal{G}_2 \Rightarrow \text{POOL}(\mathcal{G}_1) \neq_{k-\text{GWL}} \text{POOL}(\mathcal{G}_2)$

Proposition D.1. Let $\text{POOL} = (\text{SEL}, \text{RED}, \text{CON})$ such that $\text{RED} \circ \text{SEL} : (\mathcal{X}_{\mathcal{G}}^{k-(I)GWL}) \mapsto \mathcal{X}_{\mathcal{G}^P}^{k-(I)GWL}$ is injective on $\mathcal{X}_{\mathcal{G}}^{k-(I)GWL}$. Then, POOL **maintains expressivity**.

Proof. If $\text{RED} \circ \text{SEL}$ is injective, then it maps two different node feature multisets $\mathcal{X}_{\mathcal{G}_m}^{k-(I)GWL} \neq \mathcal{X}_{\mathcal{G}_n}^{k-(I)GWL}$ to different pooled node feature multisets $\mathcal{X}_{\mathcal{G}_m^P}^{k-(I)GWL} = (\text{RED} \circ \text{SEL})(\mathcal{X}_{\mathcal{G}_m}^{k-(I)GWL}) \neq (\text{RED} \circ \text{SEL})(\mathcal{X}_{\mathcal{G}_n}^{k-(I)GWL}) = \mathcal{X}_{\mathcal{G}_n}^{k-(I)GWL}$.

Thus, expressivity is maintained, as the IGWL/GWL test can distinguish the two pooled graphs independent of the choice of CON. \square

Theorem 4.1

Proof. Let $\mathbf{C}_1 \in \mathbb{R}^{N \times K}$ and $\mathbf{C}_2 \in \mathbb{R}^{M \times K}$ be the matrices representing the cluster assignments generated by $\text{SEL}(\mathcal{G}_1^L)$ and $\text{SEL}(\mathcal{G}_2^L)$, respectively.

Assuming Condition 2 holds, we have that the entries of matrices \mathbf{C}_1 and \mathbf{C}_2 satisfy equations

$$\begin{aligned}\sum_{j=1}^K c_{1_{ij}} &= \lambda, \forall i = 1, \dots, N \\ \sum_{j=1}^K c_{2_{ij}} &= \lambda, \forall i = 1, \dots, M\end{aligned}$$

Assuming condition 3 holds, we have that the j -th row of \mathbf{S}_1^P is $\mathbf{s}_{1_j}^P = \sum_{i=1}^N \mathbf{s}_{1_i}^L \cdot c_{1_{ij}}$. The same holds for the j -th row of \mathbf{S}_2^P , which is $\mathbf{s}_{2_j}^P = \sum_{i=1}^N \mathbf{s}_{2_i}^L \cdot c_{2_{ij}}$.

Suppose that there exists a permutation of rows $\pi : \{1, \dots, K\} \rightarrow \{1, \dots, K\}$ such that $\mathbf{s}_{1_j}^P = \mathbf{s}_{2_{\pi(j)}}^P \forall i = 1, \dots, N$, that is:

$$\sum_{i=1}^N \mathbf{s}_{1_i}^L \cdot c_{1_{ij}} = \sum_{i=1}^N \mathbf{s}_{2_i}^L \cdot c_{2_{i\pi(j)}} \quad \forall j = 1, \dots, K$$

Thus,

$$\begin{aligned}\sum_{j=1}^K \sum_{i=1}^N \mathbf{s}_{1_i}^L \cdot c_{1_{ij}} &= \sum_{j=1}^K \sum_{i=1}^N \mathbf{s}_{2_i}^L \cdot c_{2_{i\pi(j)}} \Leftrightarrow \\ \sum_{i=1}^N \mathbf{s}_{1_i}^L \cdot \sum_{j=1}^K c_{1_{ij}} &= \sum_{i=1}^N \mathbf{s}_{2_i}^L \cdot \sum_{j=1}^K c_{2_{i\pi(j)}} \Leftrightarrow \quad (\text{Condition 2}) \\ \sum_{i=1}^N \mathbf{s}_{1_i}^L \cdot \lambda &= \sum_{i=1}^N \mathbf{s}_{2_i}^L \cdot \lambda \Leftrightarrow \\ \sum_{i=1}^N \mathbf{s}_{1_i}^L &= \sum_{i=1}^N \mathbf{s}_{2_i}^L\end{aligned}$$

This contradicts Condition 1, thus proving that the underlying attributed coarsened graphs will remain non-isomorphic if the theorem's conditions hold.

According to Proposition 7 in Joshi et al. [8], the GWL can distinguish any pair of geometric graphs with non-isomorphic underlying attributed graphs under the assumption that graphs are constructed from point clouds using radial cutoffs. Additionally, geometric graphs with distinct underlying attributed graphs are IGWL-distinguishable.

The same proof holds for vector features that satisfy Condition 1 of the theorem. \square

Lemma D.1. *Let RED be injective function. For any two geometric graphs \mathcal{G}_1 and \mathcal{G}_2 we have*

$$\text{SEL}(\mathcal{G}_1) \neq_{k-(I)GWL} \text{SEL}(\mathcal{G}_2) \implies \text{POOL}(\mathcal{G}_1) \neq_{k-(I)GWL} \text{POOL}(\mathcal{G}_2).$$

Proof. Let \mathcal{G}_1 and \mathcal{G}_2 be two geometric graphs with $\text{SEL}(\mathcal{G}_1) \neq \text{SEL}(\mathcal{G}_2)$. In other words, the obtained supernode sets $\{C_{1_1}, \dots, C_{1_K}\} \neq \{C_{2_1}, \dots, C_{2_K}\}$ are not equal. Assuming RED is injective, it maps different multisets of node features to different supernode representations. Thus, $\text{POOL}(\mathcal{G}_1) \neq_{k-(I)GWL} \text{POOL}(\mathcal{G}_2)$. \square

Theorem D.1. Let SEL be a function such that it distinguishes any pair of GWL - or IGWL -indistinguishable graphs, i.e., $\exists \mathcal{G}_1, \mathcal{G}_2. \mathcal{G}_1 =_{k\text{-}(I)\text{GWL}} \mathcal{G}_2 \Rightarrow \text{SEL}(\mathcal{G}_1) \neq_{k\text{-}(I)\text{GWL}} \text{SEL}(\mathcal{G}_2)$

Then, a geometric pooling operator POOL exists that **increases expressivity**.

Proof. From Lemma D.1, given an injective function RED , then

$$\begin{aligned} \mathcal{G}_1 =_{k\text{-}(I)\text{GWL}} \mathcal{G}_2 &\Rightarrow \\ \text{SEL}(\mathcal{G}_1) \neq_{k\text{-}(I)\text{GWL}} \text{SEL}(\mathcal{G}_2) &\Rightarrow \\ \text{POOL}(\mathcal{G}_1) \neq_{k\text{-}(I)\text{GWL}} \text{POOL}(\mathcal{G}_2) & \end{aligned}$$

Since such an injective RED exists, there exists a function $\text{POOL} = \text{SEL} \circ \text{RED}$ that **increases expressivity**. \square