Foundations for Robust yet Simple Sparse Hierarchical Pooling: A New Perspective on Sparse Graph Pooling

Sarith Imaduwage
Independent Researcher
imaduwagesarith932@gmail.com

Abstract

This work investigates Standard Sparse Pooling (SSP) methods within Graph Neural Networks, focusing on their effectiveness in preserving graph-level information while performing local pooling. We analyze the role of Selection and Reduction functions in SSP and introduce a new perspective that addresses the shortcomings of existing methods. We reveal that while SSP is simple, it has limitations in forming hierarchical representations, leading to potential over-representation in certain regions. This study provides foundational insights into achieving robust yet simple sparse pooling without unnecessary complexities.

1 Introduction

Multi-resolution representation processes input data at multiple scales, using local pooling to enhance local invariance [1]. Several studies have explored how local pooling on graphs can improve Graph Neural Networks (GNN), which are the preferred choice for graph-structured data. In graph-level tasks like classification [2] and regression [3, 4], some details may be lost when capturing low-resolution representations, so effective graph pooling must preserve crucial graph-level information while taking advantage of local pooling. *Hierarchical Graph Pooling* achieves this by identifying hierarchical structures, such as clusters, within the feature space of input nodes. Hierarchical graph pooling can be divided into two categories: *Dense* methods, which aim to overcome the *flat* nature of GNNs by aggregating nodes to form hierarchical clusters, for example,DIFFPOOL [5], and then *Sparse* methods, which select only the nodes necessary to represent the existing hierarchical clusters.

According to the SRC framework [6] pooling operators are composed of 3 functions: ***S**elect*, ***R**educe*, and ***C**onnect*. *Select* specifies how to select a subset of nodes, *Reduce* specifies the transformation of selected node features to output node features, and *Connect* specifies relations(edges) among output nodes. The Select function of sparse methods assigns an importance score s_i to each node i, then selects a subset based on the rank determined by scores. We identify a standard setup where the scores are a function of node attributes x_i, i.e., the score is $s_i = f(x_i)$, *Select* nodes with Top-K scores, and *Reduce* selected node i' as $g(s_{i'})x_{i'}$ where g is an arbitrary function. We abuse the notation s_i for $g(s_i)$ as well, implying the assignment: $s_i := g(s_i)$. We define methods that adhere to this setup as ***S**tandard-**S**parse-**P**ooling* (SSP). We reckon SSP is the simplest form of sparse hierarchical (differentiable) pooling. TOPKPOOL [7, 8] & SAGPOOL [9] belong to SSP. TOPKPOOL calculate scores as $f(x_i) = p^\top x_i$ and SAGPOOL calculate as $f(x_i) = p^\top \mathrm{GNN}(x_i)$ where p is a trainable vector and GNN is any GNN.

The existing perspective on SSP is to interpret scores as gates, with the objective of pushing $s_i \simeq 1$ on selected nodes and $s_i \simeq 0$ on discarded nodes. Thus, *Sigmoid* is usually used as g to match with this interpretation. Yet, SSP performs well or sometimes even better when individual s_i deviates from these bounds or when g is the identity. This reveals a fundamental gap in our understanding of SSP's true objective. This gap matters because recent improvements in sparse pooling (e.g.,

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: New Perspectives in Graph Machine Learning.

ASAPPOOL [10], KMIS [11], and EDGEPOOL [12]) have added architectural complexities to SSP without questioning whether the foundational view of SSP was accurate. By overlooking the real cause of SSP's limitations, such methods risk undermining the simplicity of sparse pooling.

In this paper, we introduce a new perspective: aligning SSP's functions with the formal definition of pooling, and from that derive the accurate objective of SSP. The accurate objective we derive reveals that SSP fails to effectively form hierarchical representations due to its tendency to *over-represent* certain regions of the signal space, at the expense of balanced representation across the entire space. We theoretically attribute the cause of over-representation to a phenomenon called *partitioning*. Our contribution is not a new pooling operator, but a theoretical framework that (i) redefines SSP's objective, (ii) identifies partitioning as the cause of over-representation and, consequently, of SSP's failure to build hierarchical representations, and (iii) points to simple strategies, such as avoiding redundant partitions that improve hierarchical representation without unnecessary complexity. As mentioned earlier, recent methods increase architectural complexity to *explicitly* form clusters. See A.4) for details of explicit clustering. In contrast, we argue that once partitioning is addressed, SSP's *implicit* cluster formation is sufficient. Our work thus provides foundational insights into SSP's behavior and guiding principles for designing robust yet simple sparse pooling methods.

2 Background

2.1 Establishing a New Perspective

Informally, the purpose of pooling is to obtain a compact representation by assimilating a set of nearby inputs. Unlike the existing perspective, which treats scores as mere gates, our new perspective views them as factors that drive similar assimilation in SSP. Let $x_i \in \mathbb{R}^d$ and $s_i \in \mathbb{R}^+$ be the attributes and the score of node i, respectively.

Definition 1: Formal Definition of Pooling [1] Pooling is a transformation of a multiset of signals on a domain to another multiset of signals on a coarsened domain. Pooling at scale J is $P_J : \mathcal{X}(\Omega) \to \tilde{\mathcal{X}}(\tilde{\Omega})$ where $\mathcal{X}(\Omega, \mathcal{C}) := \{x_i : \Omega \to \mathcal{C}\}$, same for $\tilde{\mathcal{X}}(\tilde{\Omega}, \tilde{\mathcal{C}})$. $\Omega, \mathcal{C}$ denote points on the domain and channels of the signal, respectively. A function $f : \mathcal{X}(\Omega) \to \mathcal{Y}$ is said locally stable at scale J if approximation $f \approx f_J \circ P_J$ is attainable with the composition of function $f_J : \tilde{\mathcal{X}}(\tilde{\Omega}) \to \mathcal{Y}$, and pooling P_J.

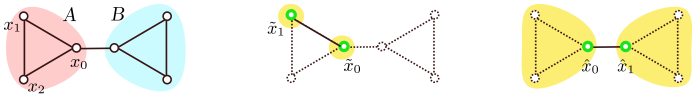
See *Def.* 1 of [2] for the formal definition of a *multiset*. For any pooled representation $\tilde{x}_j \in \tilde{\mathcal{X}}(\tilde{\Omega})$, the scaling factor $J(\tilde{x}_j)$ signifies the size of the region (number of points) on domain Ω from which the pooling map to $\tilde{x}_j$. We call this region *Receptive Field* (RF) of $\tilde{x}_j$ denoted by $\mathrm{RF}(\tilde{x}_j) \subseteq \mathcal{X}(\Omega)$. Note, $J(\tilde{x}_j) = |\mathrm{RF}(\tilde{x}_j)|$. The output $\tilde{x}_j$ is the compact representation obtained by assimilating points in $\mathrm{RF}(\tilde{x}_j)$. When the input domain Ω is a grid or sequence, pooling maps the input to the output at regular intervals, making the RF size constant for all $\tilde{x}_j$.

However, on graphs, fixed-size RFs are not feasible due to structural irregularities, so RFs on graphs must be adaptive and learned. *Our new perspective is that the scores in SSP embody RFs and drive adaptive learning of RFs on the graph*. The following formalism introduces three measures that allow us to analyze characteristics of RFs in the context of SSP.

Definition 2 Suppose any output $\tilde{x}_j = s_i x_i \in \tilde{\mathcal{X}}(\tilde{\Omega})$ from the *SSP* denoted by P such that $P(\mathrm{RF}(\tilde{x}_j)) = \tilde{x}_j$, then we introduce following measures to characterize the *(i) size, (ii) spread, and (iii) information loss while the pooling* of $\mathrm{RF}(\tilde{x}_j)$, respectively:

1. We introduce $e_s \in \mathbb{R}^+$ such that $s_i = e_s\,|\mathrm{RF}(\tilde{x}_j)|$.
2. We introduce $e_d \in \mathbb{R}^+$ such that $\sum_{\forall x_k \in \mathrm{RF}(\tilde{x}_j)} d(x_k, x_i) = e_d$, for a metric $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$.
3. If there exists a function $\Phi : \mathcal{X}(\Omega) \to \mathbb{R}^d$ so that $\Phi(X)$ is unique for each multiset $X \subseteq \mathcal{X}(\Omega)$ then $\mathrm{RF}(\tilde{x}_j) = \Psi(\Phi(\mathrm{RF}(\tilde{x}_j)))$ where Ψ is the retraction function of Φ. We introduce $e_r \in \mathbb{R}^d$ such that $\Phi(\mathrm{RF}(\tilde{x}_j)) = \tilde{x}_j + e_r = s_i x_i + e_r$.

(1) states that the score s_i scales with the size of the corresponding RF, enabling SSP to adaptively learn RFs. Here e_s measures deviation from exact proportionality. (2) defines e_d, which captures the diversity of points inside the RF by measuring their spread around the anchor node x_i using a metric. (3) states that if a suitable Φ exists the vector $\Phi(\mathrm{RF}(\tilde{x}_j))$ can be retracted to recover the original RF. In this case, e_r measures the discrepancy between the pooled output $\tilde{x}_j$ and this lossless



Input (a) Likely output (b) Desired output

Figure 1: (a) & (b) show 2 possible output graphs when 2 nodes are pooled. Yellow blobs around the pooled nodes (green circles) represent the RFs of nodes projected onto the output. In contrast to RFs of (a), RFs of (b) capture both clusters *A*, *B*, preserving the hierarchical structure of the graph. Thus (b) is preferable, whereas SSP yields (a). Assume A is information-dense, so capturing A to the fullest is desirable. Suppose node x_0 is selected and both $d(x_0, x_1)$, $d(x_0, x_2)$ are greater than ϵ in Eq. (1). Then from Eq. (1), $\Theta(x_0, \mathrm{RF}(\tilde{x}_0)) > \Theta(x_0, \mathrm{RF}(\hat{x}_0))$ and $\Theta(x_1, \mathrm{RF}(\tilde{x}_1)) > \Theta(x_0, \mathrm{RF}(\hat{x}_0))$. Thus, $\mathrm{RF}(\tilde{x}_0))$ and $\mathrm{RF}(\tilde{x}_1))$ are partitions of $\mathrm{RF}(\hat{x}_0))$, making $\tilde{x}_0$ more likely to be pooled than $\hat{x}_0$. If SSP pooled $\hat{x}_0$ it would fully capture A and then reach B as in (b). Instead, SSP over-represents A by pooling redundant partitions as shown in (a).

representation. Thus, when $e_r = 0$, no information is lost in mapping RF to $\tilde{x}_j$. Together (1), (2) & (3) suggests how each output $\tilde{x}_j$ implicitly encodes corresponding RF. Refer B.1 for more on *Def.* 2.

We next examine the conditions under which a GNN layer is maximally expressive. These conditions must be considered as we analyze the objective of SSP.

Definition 3 [2] Assume the input node feature space is *countable*. If the condition: The summation of any subset of output features of a GNN layer is injective to that subset, is satisfied, then that GNN layer is *maximally expressive*.

This is a direct derivation from the theoretical framework laid in [2]. The same work introduces the GIN layer, which is modelled to satisfy the mentioned condition to achieve maximal expressivity. Whether GIN or similarly powerful GNN satisfies this condition strictly for *any* subset is an empirical concern. But Lemma 5 in [2] shows theoretically that this is very well possible.

2.2 Objective of SSP

Here, we discuss the underlying objective of an SSP under *ideal conditions* to maintain the expressivity of preceding GNN layers. Two conditions need to be met for the ideal condition: (i) There exists a function Φ as in *Def.* 2 (3) and $e_r = 0$ (i.e, information loss is zero) for all pooled nodes, and (ii) SSP has maximum local stability (same notion in *Def.* 1, but here in its maximal form; see B.1 for detailed discussion).

Proposition 1 Assume the preceding GNN satisfies the condition in *Def.* 3 and is therefore maximally expressive. Under the aforementioned ideal conditions, the objective of SSP is to solve the following search problem:

$$\{(x_i, \mathrm{r}_k)\} = \underset{\mathbf{s}_{i,k} \in \mathbf{S}}{\text{Top-K}}\ \Theta(x_i, r_k) \quad \text{where} \quad \Theta(x, r) = \frac{|r|}{\sum_{\forall x_j \in \mathrm{r}} d(x, x_j) + \epsilon} \tag{1}$$

Proofs are in the Appendix B. $\mathbf{S}$ is a set of tuples with all the possible combinations of $\mathbf{s}_{i,k} = (x_i, \mathrm{r}_k)$ such that x_i is the i^{th} node in $\mathcal{X}(\Omega)$ and r_k is the k^{th} nonempty subset of $\mathcal{X}(\Omega)$. If $|\mathcal{X}(\Omega)| = n$, then $k \leq 2^n - 1$ and $|\mathbf{S}| = n(2^n - 1)$. Eq. (1) means the selection of Top-K tuples of unique nodes and RFs such that the sizes of those RFs are maximized while the deviation between the picked node and nodes within the corresponding RF is minimal. A negligible $\epsilon \approx 0$ is added to avoid division by zero. It's easier to understand SSP with a compact description like *Prop.* 1 more than with the existing decoupled Select-Reduce scheme. Much of what was discussed so far is based on $s_i \in \mathbb{R}^+$. In Appendix A.1, we discuss why *Prop.* 1 applies to non-linear activated scores as well.

3 Analysis of SSP

In this section, we use *Prop.* 1 to examine SSP's drawbacks and potential improvements. To that end, we define *Partitions* as follows, and all the notations are the same as in *Prop.* 1.

Partitions For a given $\mathbf{s}_{i,k} \in \mathbf{S}$, we define all $\mathbf{s}_{i',k'} \in \mathbf{S}$ such that $r_{k'} \subset r_k$ and $\Theta(x_i, r_k) \leq \Theta(x_{i'}, r_{k'})$ as *partitions* of r_k. The logical extreme case is r_k is fully symmetric i.e. all $x_i \in r_k$ are equivalent, where there are $|r_k|$ number of partitions and all partitions $(x_{i'}, r_{k'})$ satisfy $\Theta(x_{i'}, r_{k'}) = 1/\epsilon$. Generally, we can expect a sufficiently large r_k to have two or more partitions.

3.1 Over-Representative Regions

Suppose a pair (x_k, r_k) that satisfies Eq. (1). Note that partitions of r_k are also valid solutions to Eq. (1). According to Eq. (1), it's equally likely to output these partitions instead of (or along with) r_k. We define partitioning as the phenomenon where, instead of the RF r_k alone, the partitions of r_k are also pooled—this redundant pooling within the same region results in over-representation of that region. Ideally, RFs should cover the entire input space; but pooling from the same region multiple times, while others are under-represented, breaks the hierarchical property expected of SSP. When partitioning causes over-representation, the pooled graph fails to robustly encode hierarchical structures like clusters in the input. Regions with higher degrees of symmetry are the most likely for such over-representation. We illustrate this drawback with the following example and with Fig. 1.

Example: Suppose Eq. (1) gives Top-3 pairs of outputs $(x_1, r_1), (x_2, r_2), (x_3, r_3)$ where each r_i corresponds to a distinct region. Assume $x_1 \in r_1$ and let r_1 consist of two disjoint symmetric subregions r_1' and r_1''. By symmetry, each region contains at least one element equivalent to x_1. We denote these as $x_1' \in r_1'$ and $x_1'' \in r_1''$. According to definition r_1' and r_1'' are partitions, such that $\Theta(x_1', r_1') = \Theta(x_1'', r_1'') = \Theta(x_1, r_1)$. In this case, Eq. (1) can instead yield $(x_1', r_1'), (x_1'', r_1''), (x_2, r_2)$ as Top-3 pairs, leaving region r_3 excluded in favour of redundant partitions of r_1.

3.2 Mitigating Partitioning

$$\{(x_i, \mathrm{r}_k)\} = \underset{\mathbf{s}_{i,k} \in \mathbf{S}}{\operatorname{argmax}} \sum^{\mathrm{K}} \Theta(x_i, r_k) - \Delta_K(x_i, r_k) \tag{2}$$

Equation (2) selects a set of K tuples $\{(x_i, r_k)\}$ from the candidate pool by maximizing the total base utility $\Theta(x_i, r_k)$ minus a *set-dependent* redundancy penalty $\Delta_K(x_i, r_k)$. Here, Θ is the per-pair score from Proposition 1 (favouring larger, more compact receptive fields), while $\Delta_K \geq 0$ depends on the current selected K-tuples and increases when r_k overlaps with or closely resembles receptive fields already in selected K-tuples. This induces *diminishing returns*: once one subregion is selected, adding another from the same area yields a less marginal gain, so the optimizer favours complementary regions over redundant partitions.

We do not explicitly model Eq. (2) in this paper. But the key takeaway is that one must incorporate a set-aware operation that promotes diverse region selection. In practice, Δ_K can be implemented *implicitly* via a set-aware operation such as layer-wise summaries/readouts, which saturate under repeated coverage. Saturation then leads to diminishing returns, so the gradient optimizer naturally prefers picking something different.

Example: Under Eq. (2), the set of K pairs is chosen to maximize $\sum \Theta(x_i, r_k) - \Delta_K(x_i, r_k)$, where the set-dependent penalty Δ_K increases when a new selection is redundant with those already in K pairs. In above example, after selecting (x_1, r_1), taking a partition (x_1', r_1') incurs a penalty, so its marginal gain becomes $\Theta(x_1', r_1') - \Delta_K(x_1', r_1')$. Same for the partition (x_1'', r_1''). If the third region satisfies $\Theta(x_3, r_3) > \Theta(x_1', r_1') - \Delta_K(x_1', r_1')$, Eq. (2) prefers the diverse set $(x_1, r_1), (x_2, r_2), (x_3, r_3)$ over the redundant $(x_1', r_1'), (x_1'', r_1''), (x_2, r_2)$. Hence, the penalty term discourages selecting partitions and promotes complementary coverage.

4 Experiments & Results

We design two experiments to empirically validate our theoretical analysis of SSP.

- **Experiment 1** investigates how partitions manifest in practice and provides evidence that it leads to over-representation.
- **Experiment 2** examines whether incorporating summaries of pooled nodes can mitigate partitioning and yield more balanced hierarchical representations.

Recall, in SSP, nodes are ranked by their scores, and the top-ranked ones are selected for reduction. In both experiments, we cluster input nodes by features, record how many from each cluster are selected or left out, and measure the cosine similarity between the weight vector p and the highest-ranked member of each cluster. The cosine similarity indicates SSP's selection bias toward that cluster, since higher alignment with p makes selection more likely. This way, we can analyze how p contributes to over-representation. For clustering, we used HDBSCAN [13].

No.	Inx	Pooled	Left	Cos. Value
0	592	15	42	-0.021142
1	912	0	90	-0.056889
2	568	34	70	-0.032174
3	0	602	28	0.072586
4	883	0	181	-0.087058
5		16	255	N/A

(a) Airplane(mcs $= 20$)

No.	Inx	Pooled	Left	Cos. Value
0	51	58	0	0.037432
1	396	22	0	0.050846
2	35	67	0	0.059104
3	138	408	0	0.054445
4	924	36	960	-0.039293
5		369	0	N/A

(b) Car (mcs $= 20$)

No.	Inx	Pooled	Left	Cos. Value
0	3514	0	10	-0.094549
1	0	2480	2181	0.458036
2		394	683	N/A

(c) DD without G-JK

No.	Inx	Pooled	Left	Cos. Value
0	5680	0	13	-0.063847
1	4199	0	11	0.199028
2	18	1205	1329	0.579876
3		1669	1521	N/A

(d) DD mean G-JK

Figure 2: *mcs* denotes minimum cluster size. The last row quantifies *noise*: nodes belonging to clusters below mcs. The columns are - **No.**: cluster identifier, **Inx**: index of the first pooled node, **Pooled**: number of pooled nodes from the cluster, **Left**: number of nodes left out, **Cos.Value**- Cosine similarity between p and 1st member.

4.1 Experiment 1

Evaluation procedure We first test whether SSP exhibits partitioning and over-representation in practice. We use a Graph Autoencoder (GAE) [14] with SSP as the final encoder layer. Since GAE minimizes reconstruction error, the pooled graph must retain as much information as possible from the original, allowing fair evaluation of SSP's representational limits. See C.1 for full details on the GAE setup. We select Airplane, Car & Person point clouds from ModelNet40 [15] for the experiment.

Results We clearly observe that SSP lacks balanced pooling, exhibiting two distinct partitioning patterns. **Pattern P1:** SSP disproportionately pools many smaller clusters, leaving the largest clusters under-represented. These left-out clusters tend to be located farther from the vector p. **Pattern P2:** SSP over-represents the largest clusters, leaving smaller clusters under-represented. In practice, larger clusters are typically information-dense, so a slight bias toward them is expected. Pattern P1, however, shows the opposite: instead, p is overwhelmed by partitions of smaller clusters, leaving fewer chances to pool from larger ones. P2 is an extreme bias toward large clusters, causing loss of crucial information from smaller ones. In Fig. 2(b), P1 appears as cluster 4 is disproportionately left out (under-represented) and lies farthest from p in cosine similarity. In (a), P2 occurs as cluster 3 is disproportionately pooled (over-represented), leaving out moderately large clusters 1 and 4. See A.2 for further evidence.

4.2 Experiment 2

Evaluation procedure We next test whether incorporating summaries of pooled nodes can reduce partitioning effects and improve hierarchical representations. A readout produces a single vector summary of the input. Since the last layer of a graph classifier is itself a readout, it is straightforward to incorporate readouts from each SSP layer into the final output. In our setup, we have 2 GNN layers, each followed by an SSP. Let $\tilde{\mathcal{X}}^{(i)}$ be the output node features of i^{th} SSP, then the corresponding readout is $\text{aggr}(\tilde{\mathcal{X}}^{(i)}) || \max(\tilde{\mathcal{X}}^{(i)})$ where $aggr$ is element-wise sum or mean, and $\max$ is element-wise max. The final output is obtained by summing all readouts. We dub this network G-JK-Net. The rationale behind G-JK is to indirectly encode Eq. (2) in the model's objective. See C.2 for full details on the graph classifier. We use D&D & PROTEINS [16] datasets for the experiment.

Models	D&D	PROTEINS
NO-POOL	74.2±3.4	74.8±4.6
TOPKPOOL	58.7±0.3	73.1±5.5
TOPKPOOL-MEAN	75.5±2.7	74.7±3.6
TOPKPOOL-SUM	79.1±3.7	76.2±3.8
SAGPOOL	70.6±3.2	71.3±2.7
SAGPOOL-MEAN	75.4±3.3	74.5±3.5
SAGPOOL-SUM	79.1±3.5	75.7±3.4

Table 1: Each model has 2 GNN layers. NO-POOL denotes no pooling. SAGPOOL, SAGPOOL-MEAN, SAGPOOL-SUM denote SAGPOOL without G-JK and G-JK with stated aggregations respectively. Same for TOPKPOOL

Results We evaluate this setup for its effect on partitioning and accuracy, with accuracy serving as another indicator for the quality of hierarchical representations. Cluster analysis of the first SSP layer (Fig. 2(c), 2(d)) shows that adding summaries (G-JK-Net using mean or sum aggregation) avoids the over-representation patterns identified in Experiment 1. Without G-JK, more nodes from

the largest cluster are pooled, leaving less room to capture other clusters or noise. In contrast, with G-JK, SSP pools more evenly from the largest cluster—even when it is closest to p—indicating p is not overwhelmed by partitioning and thereby allowing space to include other regions. See A.3 for more evidence from cluster analysis. This more balanced pooling leads to the significant accuracy improvements reported in Table 1 compared to vanilla SSP.

5 Conclusion and Future Directions

Our analysis shows that the key to robust hierarchical representations in sparse pooling is avoiding over-representation caused by redundant partitions. Incorporating summaries of pooled nodes, as in G-JK-Net, demonstrates a simple yet effective way to reduce such redundancy and achieve more balanced pooling.

Rather than *explicitly* forming clusters through assignment matrices, as in ASAPPOOL [10] and KMIS [11], we argue that SSP can naturally recover hierarchical structure accurately once redundancies are identified and removed. From this perspective, the role of the *Select* function is not merely to score nodes, but to actively eliminate redundant nodes. Notably, a recent independent work applies redundancy handling at the *Select* function and attains state-of-the-art results on standard graph benchmarks, further validating this perspective [17]. We therefore encourage future research to invest in *redundancy handling* as a guiding principle for sparse pooling design. See A.4 for the rationale in detail. The right redundancy-aware mechanism, combined with SSP's inherent simplicity, will yield pooling methods that are both more effective and more efficient than existing approaches.

References

[1] Michael M Bronstein, Joan Bruna, and Cohen. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[2] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

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

[4] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28, 2015.

[5] Zhitao Ying, Jiaxuan You, Christopher Morris, Ren, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31, 2018.

[6] Daniele Grattarola, Daniele Zambon, Filippo Maria Bianchi, and Cesare Alippi. Understanding pooling in graph neural networks. *IEEE transactions on neural networks and learning systems*, 2022.

[7] Hongyang Gao and Shuiwang Ji. Graph u-nets. In *international conference on machine learning*, pages 2083–2092. PMLR, 2019.

[8] Cătălina Cangea, Veličković, Thomas Kipf, and Pietro Liò. Towards sparse hierarchical graph classifiers. *arXiv preprint arXiv:1811.01287*, 2018.

[9] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. In *International conference on machine learning*, pages 3734–3743. PMLR, 2019.

[10] Ekagra Ranjan, Soumya Sanyal, and Partha Talukdar. Asap: Adaptive structure aware pooling for learning hierarchical graph representations. In *Proceedings of the AAAI conference on artificial intelligence*, pages 5470–5477, 2020.

[11] Davide Bacciu, Alessio Conte, and Francesco Landolfi. Generalizing downsampling from regular data to graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6718–6727, 2023.

[12] Frederik Diehl. Edge contraction pooling for graph neural networks. *arXiv preprint arXiv:1905.10990*, 2019.

[13] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining*, pages 160–172, 2013.

[14] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[15] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[16] Paul D. Dobson and Andrew J. Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of Molecular Biology*, 330(4):771–783, 2003.

[17] Sarith Imaduwage. Skippool: Improved sparse hierarchical graph pooling with differentiable exploration. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(17):17546–17554, Apr. 2025. doi: 10.1609/aaai.v39i17.33929. URL `https://ojs.aaai.org/index.php/AAAI/article/view/33929`.

[18] Karsten M. Borgwardt, Cheng Soon Ong, Stefan Schönauer, S. V. N. Vishwanathan, Alex J. Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21:47–56, 2005. URL `https://doi.org/10.1093/bioinformatics/bti1007`.

[19] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(77):2539–2561, 2011. URL `http://jmlr.org/papers/v12/shervashidze11a.html`.

[20] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020. URL `www.graphlearning.io`.

No.	Inx	Pooled	Left	Cos. Value
7	2963	0	38	-0.077716
8	2965	0	40	-0.078803
19	1646	1	41	-0.042371
9	3097	0	66	-0.079881
13	1428	2	69	-0.024132
15	24	86	3	0.018040
18	1606	2	147	-0.047230
25	317	315	0	-0.018426
22	91	734	0	-0.009497
26		459	1013	N/A

(a) Person (mcs = 10)

No.	Inx	Pooled	Left	Cos. Value
4	976	8	19	-0.025765
2	2543	0	30	-0.030166
11	172	44	0	0.001459
3	2645	0	133	-0.051508
9	1940	0	142	-0.054570
6	1186	46	107	-0.038707
1	41	146	108	0.089808
8	914	277	14	-0.023322
10	189	607	119	0.000553
12		454	980	N/A

(b) Person (mcs = 15)

No.	Inx	Pooled	Left	Cos. Value
0	51	58	0	0.037432
1	396	22	0	0.050846
2	35	67	0	0.059104
3	138	408	0	0.054445
4	924	36	960	-0.039293
5		369	0	N/A

(c) Car (mcs = 20)

No.	Inx	Pooled	Left	Cos. Value
0	1	10	0	0.036772
1	842	37	960	0.056337
2	386	380	0	0.038970
3	69	145	0	0.050565
4	41	135	0	0.114321
5		253	0	N/A

(d) Car (mcs = 10)

No.	Inx	Pooled	Left	Cos. Value
0	592	15	42	-0.021142
1	912	0	90	-0.056889
2	568	34	70	-0.032174
3	0	602	28	0.072586
4	883	0	181	-0.087058
5		16	255	N/A

(e) Airplane (mcs = 20)

No.	Inx	Pooled	Left	Cos. Value
44	140	20	0	0.066526
7	264	20	0	0.082965
28	4	38	0	0.071139
58	539	34	15	0.058029
51	698	0	67	0.041896
82		125	157	N/A

(f) Airplane (mcs = 5)

Figure 3: Cluster analysis on *Person*, *Car*, *Airplane* graphs.

A More on the Perspective Shift

A.1 Viability of Relaxing Non-linear Activation on Scores

Graph	g	MSE	Range
Person	Tanh	3.13	$[-0.81, 0.99]$
	Sig.	2.62	$[0.27, 0.46]$
	Id.	2.83	$[-1.28, 4.11]$
Airplane	Tanh	5.53	$[-0.91, 0.99]$
	Sig.	5.16	$[0.41, 0.99]$
	Id	5.37	$[-9.30, 14.91]$

Table 2: Recon. loss comparison. See C.1 for the setup.

g	DD	PROTEINS
Tanh	75.4 ± 3.3 (+)	74.5 ± 3.5 (−)
Sig.	76.0 ± 3.5	74.1 ± 4.7
Id.	75.7 ± 3.4 (+)	74.4 ± 4.0 (−)

Table 3: Analysing impact of sign. Evaluated over 10 folds under the same seed (777).

Recall scores can be followed by an activation g such that $s_i := g(s_i)$. We repeated the reconstruction experiment in *Sec.* 4 with different activations and Table 2 summarizes the MSE scores and the range of scores taken. Sig. & Id. are Identity & Sigmoid respectively. When we repeated the classification experiment in *Sec.* 4 with different activation, we observed that for PROTEINS with both Tanh and Id. activation, all scores are negative, and for DD, all scores are positive. Table 3 summarizes these results, with the sign shown next to the accuracy.

According to the last column in Table 2, irrespective of the activation, scores don't converge to 1 on selected nodes as per the old perspective. Moreover, MSE values suggest having negative scores or scores falling out of the desired bound (0 or 1) does not impact the performance. These results suggest that the old perspective is incomplete and the scores may encompass more complexities. Our new perspective on SSP is that scores relate to RFs. The size of RF of any output $\tilde{x}_j = s_i x_i$ satisfy $|\mathrm{RF}(\tilde{x}_j)| \geq 1$. Thus, assuming any output's score as $s_i \in \mathbb{R}^+$ allows us to build mathematical formalism around the new perspective. Even though we derive an explanation of SSP: *Prop.* 1 which is score-independent, we need to verify if assuming scores as $s_i \in \mathbb{R}^+$ is a strong assumption that would make *Prop.* 1 inapplicable to rest such as non-linear activated scores.

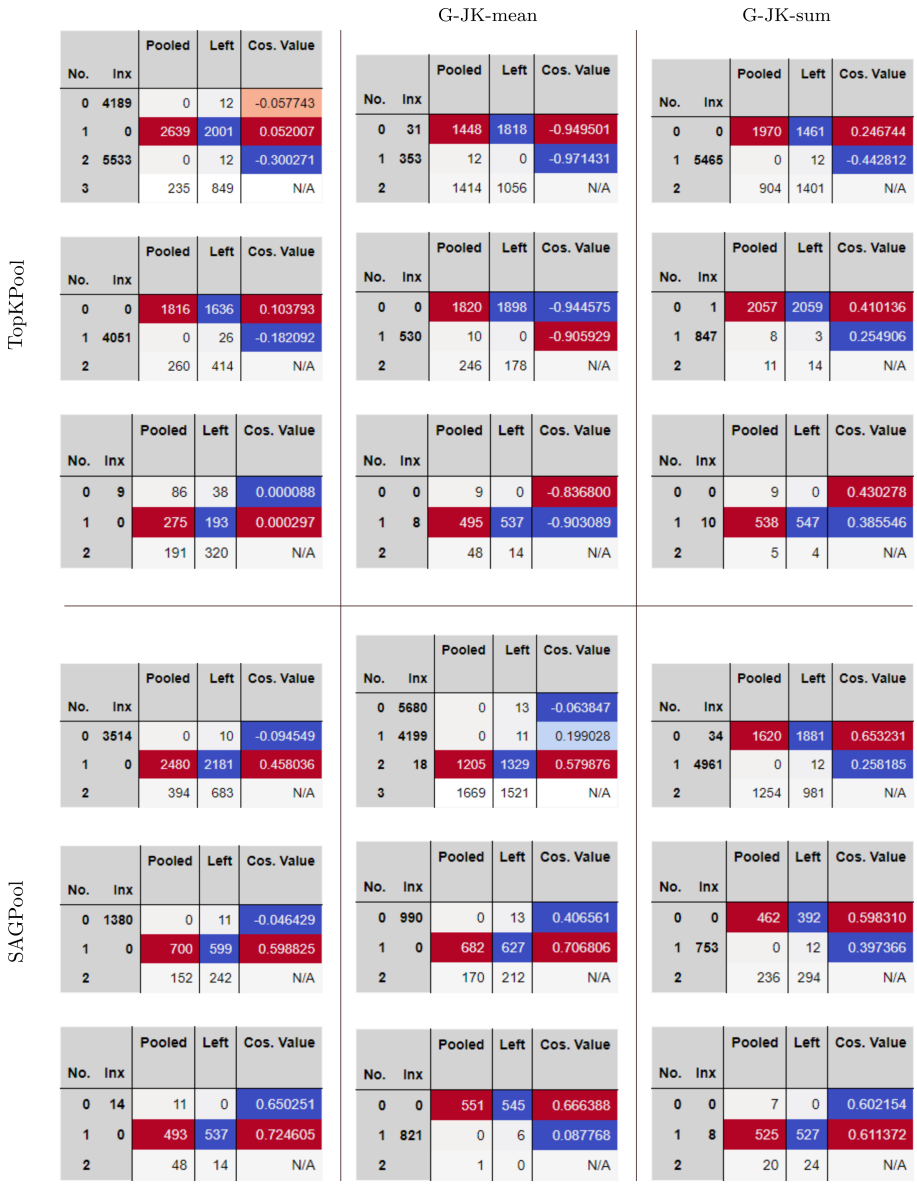


Figure 4: First column denotes runs without G-JK. Each evaluation in a row is produced by the same split of data.

Table 2 indicates that identity-activated scores are as effective as non-linear activated scores, so scores being ≥ 1 or ≤ -1 don't have a significant impact. Furthermore, the last row of Table 3 indicates that identity-activated scores remain effective even if the signs of scores are all positive or negative. Altogether suggests that positive identity-activated scores, where $g = \text{Id.}$ & $\text{sign} = +$, are sufficient to reason about the combined hypothesis class $\{\text{Tanh, Sig., Id.}\} \times \{+, -, \pm\}$ of scores' activation and sign. Thus we justify assuming all $s_i \in \mathbb{R}^+$ doesn't invalidate *Prop.* 1.

A.2 Empirical Evidence for Partitioning

Recall patterns P1, P2 from *Sec.* 4. When we say a cluster is pooled or left out, we don't necessarily mean entirely but disproportionately. For example, when we say the largest clusters are pooled, we mean about >90% is pooled. We see P1 in Fig. 3 (c),(d), and (f). For example in (d), the largest cluster 1 is disproportionately left out and is furthest from p. We see P2 in (a),(b), and (e). For example in (a), clusters 22 & 25 are completely pooled leaving out modestly large clusters $9, 13, 18$.

A.3 Cluster Analysis with G-JK

Fig. 4 provides more cluster analysis related to Experiment 2. We show 3 out of the 10 total runs that best represent the common patterns. In many G-JK-mean and G-JK-sum instances, we observe that SSP pools fewer nodes from the largest cluster, allowing chance to pool from other clusters or noise. This is happening while p being located close to these pooled large clusters confirms the effectiveness of enforcing summation. This is a clear improvement compared to when no G-JK is used. Even if the above pattern is not present, in G-JK-Sum and G-JK-mean there is less gap between the pooled and left-out amounts than when no G-JK is used. For example, see rows 5,6 for G-JK-mean and rows 1,5 for G-JK-sum. Moreover, we observe that G-JK-Sum and G-JK-mean form clusters more effectively than when no G-JK is used. In some no G-JK instances, all the nodes belong to noises.

Remark: Ideal conditions aren't strictly needed to utilize *Prop.* 1 for insights. Note that the evidence given so far is observed under a general setup. Therefore insights coming from *Prop.* 1 are not exclusive to ideal conditions.

A.4 Opinion: Think Redundancy Handling Instead of Clustering

A.4.1 Explicit Clustering

Explicitly forming clusters means that pooling learns an assignment matrix $S \in \mathbb{R}^{n \times k}$, where each entry $S_{i,j}$ denotes the assignment score of node i to supernode j. Each supernode therefore contains multiple member nodes. By contrast, SSP does not construct such an assignment matrix, which is what makes it simple. If we reinterpret SSP in this framework, the selected nodes themselves can be viewed as supernodes, each containing only a single member. In other words, only the diagonal entries $S_{i',i'}$ corresponding to selected nodes have nonzero values.

A.4.2 Redundancy Handling

Our analysis shows that many of these supernodes are redundant. If the *Select* function of SSP were enhanced to identify and eliminate these redundancies, then, combined with SSP's inherent adaptability, we would obtain a set of supernodes that capture clusters more effectively. We therefore encourage future research to focus on **redundancy handling**, as the right mechanism, together with SSP's simplicity, may ultimately outperform explicit clustering approaches.

B Proofs

B.1 Definitions

Definition 2: The relation between s_i and $|\mathrm{RF}(\tilde{x}_j)|$ is defined in *Def.* 2 (1). e_s is the measurement of deviation between s_i and $|\mathrm{RF}(\tilde{x}_j)|$. Informally, a pooled node is an assimilation of points of its RF; thus, we additionally need a characterization of RFs using a notion of metric, as in *Def.* 2 (2). The metric d in *Def.* 2 (2) measures the deviation between a node feature and the input feature x_i, and e_d is the summation of deviations over all the features in $\mathrm{RF}(\tilde{x}_j)$. Thus, e_d captures how different assimilated features in the RF are from the input feature.

Local Stability: We are concerned about signal deformation. Suppose we are approximating a function $f : \mathcal{X} \to \mathcal{Y}$ with a model f' such that $f'(x) \approx f(x) = y$. Signal deformation stability is the property that if a deformation δ is added to the input x (such that the semantic meaning of x does not change), then $f'(x + \delta) \approx y$ should still hold.

Maximal Local Stability: Among the various options for modeling f', the one that yields the greatest stability is considered to achieve maximally stable approximation. This concept can also be applied locally, referred to as maximal local stability. If the parameters of f' are stable, f' is more likely to produce consistent outputs when small, non-semantic deformations are applied to the input. This consistency is essential for maximal deformation stability.

B.2 Proofs

Lemma 1 Assume the preceding GNN satisfies the condition in *Def.* 3. In the limits, $\lim e_r \rightarrow 0 \wedge \lim e_s \rightarrow 1 \Rightarrow \lim e_d \rightarrow 0$.[1]

Proof. With our assumption, we have a well defined Φ for *Def.* 2 (3) such that $\Phi(\mathrm{RF}(\tilde{x}_j)) = \sum_{\forall x_k \in \mathrm{RF}(\tilde{x}_j)} x_k$. Consider SSP pooling $\mathrm{RF}(\tilde{x}_j)$: $P(\mathrm{RF}(\tilde{x}_j)) = s_i x_i$. According to *Def.* 2 (3), $e_r = \sum_{\forall x_k \in \mathrm{RF}(\tilde{x}_j)} x_k - s_i x_i$. If $e_r = 0$ then the following relation holds:

$$\sum_{\forall x_k \in \mathrm{RF}} x_k = s_i x_i$$

This is satisfied when both $x_k \equiv x_i$ for all $x_k \in \mathrm{RF}(\tilde{x}_j)$ *and* $s_i = |\mathrm{RF}(\tilde{x}_j)|$ i.e. $e_s = 1$. L.H.S summation is unique to the multiset i.e. $\mathrm{RF}(\tilde{x}_j)$, therefore there can not be another multiset that satisfies the relation. This means if $e_r = 0$ and $e_s = 1$, all RF elements must be equivalent to x_i in R.H.S. Thus $e_d = 0$. □

Claim 2 Under mild conditions, *SSP* learns to capture $\mathrm{RF}(\tilde{x}_j)$ by $s_i x_i$ such that $x_i \in \mathrm{RF}(\tilde{x}_j)$.

SSP includes x_i from which the respective receptive field is approximated. We say $\mathrm{RF}(\tilde{x}_j)$ contains x_i and its nearby points. This aligns with the objective of pooling to assimilate nearby points.

Lemma 2 Assume $\mathcal{X} \subset \mathbb{R}$. If $P(\mathrm{RF}(\tilde{x}_j)) = s_i x_i$ then $s_i \leq |\mathrm{RF}(\tilde{x}_j)|$.

Proof. Suppose S is the assimilation desired to be modeled by $s_i x_i$. $S = \sum_{\forall x_k \in \mathrm{RF}(\tilde{x}_j)} \alpha_k x_k \in \mathbb{R}$ is a linear combination of $\mathrm{RF}(\tilde{x}_j)$ where $0 \leq \alpha_k \leq 1$ signifies the contribution of each $x_k \in \mathrm{RF}(\tilde{x}_j)$. According to *Def. 1*, this assimilation S is the input to f_j. Since scores are parameterized and adaptive, the following relations hold for all $x_k \in \mathrm{RF}(\tilde{x}_j)$ and their scores s_k:

$$s_k = \frac{S}{x_k} \tag{3}$$

$$\frac{\mathrm{d}s_k}{\mathrm{d}S} = \frac{1}{x_k} \tag{4}$$

From Eq. (3), for each $x_k \in \mathrm{RF}(\tilde{x}_j)$ has a score s_k that can be learned to model a given S. Therefore all $x_k \in \mathrm{RF}(\tilde{x}_j)$ are plausible candidates for x_i to model any S. If a small distortion is added to the input, then S will also be distorted. While learning, weights of s_i shouldn't abruptly change against small distortions in input so the weights are more resilient and yield stable $\mathcal{X} \rightarrow \mathcal{Y}$. Eq. (4) shows how each s_k changes with the assimilation S, roughly quantifying how s_k changes with the input as well. Even though all $x_k \in \mathrm{RF}(\tilde{x}_j)$ are plausible for x_i, to satisfy maximal local stability (ideal condition), Eq. (4) must be minimal. Hence (ideally) $x_i = \max(\mathrm{RF}(\tilde{x}_j))$. For s_i, x_i following holds:

$$S = s_i x_i \leq |\mathrm{RF}(\tilde{x}_j)| \max(\mathrm{RF}(\tilde{x}_j)) \tag{5}$$

Equality holds iff all $x_k \in \mathrm{RF}(\tilde{x}_j)$ are equivalent and S is summation of $\mathrm{RF}(\tilde{x}_j)$. Under ideal conditions:

$$s_i \max(\mathrm{RF}(\tilde{x}_j)) \leq |\mathrm{RF}(\tilde{x}_j)| \max(\mathrm{RF}(\tilde{x}_j)) \tag{6}$$

$$s_i \leq |\mathrm{RF}(\tilde{x}_j)| \tag{7}$$

Hence we prove for the restricted case $\mathcal{X} \subset \mathbb{R}$, s_i is upper bounded by the corresponding receptive field size. □

Theorem 1 Assume the preceding GNN satisfies the condition in *Def.* 3. Under the ideal conditions, *SSP* satisfies the following;

i. $\lim e_r \rightarrow 0 \Rightarrow \lim e_s \rightarrow 1$ for $\mathcal{X} \subset \mathbb{R}$
ii. $\exists \epsilon \geq 0$ such that $\lim e_r \rightarrow 0 \Rightarrow \lim e_s \rightarrow 1 + \epsilon$ for $\mathcal{X} \subset \mathbb{R}^n, \forall n \in \mathbb{N}^+$

[1] $\wedge$ & $\Rightarrow$ denotes logical AND and implication respectively.

Proof. In the light of changing from one receptive field to another likely receptive field, we see how e_r and e_s interplay. Consider SSP pooling $\mathrm{RF}(\tilde{x}_j)$: $P(\mathrm{RF}(\tilde{x}_j)) = s_i x_i$. With our assumption, we have a well defined Φ for *Def.* 2 (3) such that $\Phi(\mathrm{RF}(\tilde{x}_j)) = \sum_{\forall x_k \in \mathrm{RF}(\tilde{x}_j)} x_k$. According to *Def.* 2 (3) the following relation holds:

$$\Phi(\mathrm{RF}(\tilde{x}_j)) = s_i x_i + e_r \tag{8}$$

There is a nonempty set $P_i \subseteq \mathcal{X}$, such that $x_k = x_i \ \forall x_k \in P_i$ and $\mathcal{X}/P_i \cap x_i = \emptyset$. Suppose there is at least one $x_i \in P_i$ which is not inside $\mathrm{RF}(\tilde{x}_j)$.

According to *claim 2*, $\mathrm{RF}(\tilde{x}'_j) = \mathrm{RF}(\tilde{x}_j) \cup \{x_i\}$ is a likely RF. Then, as per *Def. 2 (3)*:

$$\begin{aligned} \Phi(\mathrm{RF}(\tilde{x}'_j)) &= s_i x_i + e_r + x_i \\ \tilde{s}_i x_i + \tilde{e}_r &= (s_i + 1)x_i + e_r \end{aligned} \tag{9}$$

Notice that node i has not changed, only the RF has changed. With *Def.* 2 (1) and rearrangement we derive;

$$\begin{aligned} (1 + s_i - \tilde{s}_i)x_i &= \tilde{e}_r - e_r \\ (1 + e_s \left|\mathrm{RF}(\tilde{x}_j)\right| - \tilde{e}_s \left|\mathrm{RF}(\tilde{x}'_j)\right|)x_i &= \tilde{e}_r - e_r \end{aligned} \tag{10}$$

Since $\mathrm{RF}(\tilde{x}'_j) = \mathrm{RF}(\tilde{x}_j) \cup \{x_i\}$, we derive:

$$\begin{aligned} (1 + e_s \left|\mathrm{RF}(\tilde{x}_j)\right| - \tilde{e}_s \left\{\left|\mathrm{RF}(\tilde{x}_j)\right| + 1\right\})x_i &= \tilde{e}_r - e_r \\ (1 + (e_s - \tilde{e}_s) \left|\mathrm{RF}(\tilde{x}_j)\right| - \tilde{e}_s)x_i &= \tilde{e}_r - e_r \end{aligned} \tag{11}$$

It's clear if $e_r \to 0$, $\tilde{e}_r \to 0$ then $(\tilde{e}_r - e_r) \to 0$. Similarly for a given $R = |\mathrm{RF}(\tilde{x}_j)| > 0$ and $x_i \neq 0$, $L_R(e_s, \tilde{e}_s) = \{1 + (e_s - \tilde{e}_s)R - \tilde{e}_s\} \to 0$ as $e_r \to 0$, $\tilde{e}_r \to 0$. Hence asymptotics α_1, α_2 of $e_s, \tilde{e}_s$ respectively satisfy the following.

$$\lim_{e_s, \tilde{e}_s \to \alpha_1, \alpha_2} \{1 + (e_s - \tilde{e}_s)R - \tilde{e}_s\} = 0 \tag{12}$$

$\forall e_s, \tilde{e}_s \in \mathbb{R}$, $L_R(e_s, \tilde{e}_s)$ is continuous, hence by substitution we get;

$$1 + (\alpha_1 - \alpha_2)R - \alpha_2 = 0 \tag{13}$$

$$\alpha_1 = \frac{R+1}{R}\alpha_2 - \frac{1}{R} \tag{14}$$

Note, in SSP, s_i is a function $s : \mathcal{X} \to \mathbb{R}$ of x_i or more specifically on the node i. s_i which is used to approximate $\mathrm{RF}(\tilde{x}_j)$, remains unchanged as long as the node i or x_i remains unchanged. Which means $s_i = \tilde{s}_i$. Hence with *Def.* 2 (1);

$$s_i = \tilde{s}_i \Rightarrow R e_s = (1 + R)\tilde{e}_s \Rightarrow \frac{R}{R+1} e_s = \tilde{e}_s \Rightarrow e_s \geq \tilde{e}_s \tag{15}$$

Equality holds *iff* $s_i = \tilde{s}_i = 0$ yielding $e_s = \tilde{e}_s = 0$.

Even though (20) implies there are infinitely many α_1, α_2 pairs, (21) implies e_s, $\tilde{e}_s$ can only reach to a α_1, α_2 pair such that a $\alpha_1 \geq \alpha_2$. From (20) we see:

$$\begin{aligned} \frac{R+1}{R}\alpha_2 - \frac{1}{R} &\geqslant \alpha_2 \\ \frac{1}{R}\alpha_2 &\geqslant \frac{1}{R} \\ \alpha_2 &\geqslant 1 \quad (\because R > 0) \end{aligned} \tag{16}$$

For $\mathcal{X} \subset \mathbb{R}$ from Lemma 2, we can say:

$$\begin{aligned} \tilde{s}_m \leq \left|\mathrm{RF}(\tilde{x}'_j)\right| \Rightarrow \tilde{e}_s \left|\mathrm{RF}(\tilde{x}'_j)\right| &\leq \left|\mathrm{RF}(\tilde{x}'_j)\right| \\ \Rightarrow \tilde{e}_s \leq 1 \Rightarrow \alpha_2 &\leq 1 \end{aligned} \tag{17}$$

Combining results (22) and (23), we say $1 \leq \alpha_2 \leq 1$, which implies $\alpha_2 = 1$. Hence we prove the first part of the theorem that in the case of $\mathcal{X} \subset \mathbb{R}$, as $e_r \to 0$, then asymptotically $e_s \to 1$. For the general $\mathcal{X} \subset \mathbb{R}^n$, from (22) we can say, as $e_r \to 0$, then asymptotically $e_s \to 1 + \epsilon$ for $\epsilon \geq 0$. □

Claim 3 For $\mathcal{X} \subset \mathbb{R}^n$ where $n > 1$, in limit $e_r \to 0$, e_s converges to $1 + o(1)$.

Consider SSP pooling $\mathrm{RF}(\tilde{x}_j)$: $P(\mathrm{RF}(\tilde{x}_j)) = s_i x_i$. Recall Lemma 2 proof. For $\mathcal{X} \subset \mathbb{R}^n$, we can't guarantee above x_i will be $\max(\mathrm{RF}(\tilde{x}_j))$ so that Eq. (4) is minimal, due to multidimensionality. However, x_i must still agree to $x_i[k] \approx \max(\mathrm{RF}(\tilde{x}_j))[k]$ for all k, for a higher local stability.[2] For $\mathcal{X} \subset \mathbb{R}^n$, similar to Eq. (6) we say the following for x_i: For all k,

[2] k denote the index of vector elements.$\min$ and $\max$ over a set of vectors denote element wise min/max.

$s_i(x_i[k]) \leq |\mathrm{RF}(\tilde{x}_j)|\,(\max(\mathrm{RF}(\tilde{x}_j))[k])$. This implies $s_i \leq |\mathrm{RF}(\tilde{x}_j)| + \tilde{\epsilon}$, for some error $\tilde{\epsilon} \geq 0$ such that it holds the following monotonic relation:

$$\tilde{\epsilon} \propto \max_{\forall k}\left(\frac{\max(\mathrm{RF}(\tilde{x}_j))[k]}{x_i[k]}\right)$$

Moreover for $\mathcal{X} \subset \mathbb{R}^n$, $s_i \leq |\mathrm{RF}(\tilde{x}_j)|$ with $\tilde{\epsilon} = 0$ *iff* $x_i = \max(\mathrm{RF}(\tilde{x}_j))$. Knowing that pooling assimilates x_i and features similar to x_i inside the RF, we can restate that $x_i[k]$ is arbitrarily closer to $\max(\mathrm{RF}(\tilde{x}_j))[k]$ for all k. Thus $\tilde{\epsilon}$ becomes arbitrarily small, which means ϵ in Theorem 1 (ii) becomes arbitrarily small.

Corollary 1 Under the ideal conditions i.e when $e_r \to 0$ and local stability is maximal, $s_i \to |\mathrm{RF}(\tilde{x}_j)|$ and $e_d \to 0$ for a pooled node $\tilde{x}_j = s_i x_i$ from SSP.

Proof. From Theorem 1 and Claim 3 altogether: As $e_r \to 0$, $s_i \to |\mathrm{RF}(\tilde{x}_j)|$ that is $e_s \to 1$. With Lemma 1 this consequently means $e_d \to 0$. Thus proves the above for SSP. □

Proposition 1

Proof. According to Corollary 1, since $e_s \to 1$, Top-K score selection in SSP is equivalent to finding the largest k RFs in input node space and at the same time since $e_d \to 0$, in addition to being the largest these RFs must have low variance. Thus SSP can be viewed as a search problem where each element $\mathbf{s}_{i,k} = (x_i, \mathrm{r}_k)$ in the search space $\mathbf{S}$ is weighted by the importance given by:

$$\Theta(x_i, r_k) = \frac{|r_k|}{\sum_{\forall x_j \in \mathrm{r_k}} d(x_i, x_j) + \epsilon}$$

Why Top-K? A score s_i is modeled solely on local information i.e. node itself or local neighborhood. Hence we can't expect that SSP would favor an RF r_k over an RF r'_k located locally to x_i where $|r_k| >> |r'_k|$ even if the corresponding $\Theta(\cdot,\cdot)$ are the same. To model much larger r_k requires more information further to x_i. This weakens the possibility of a *summation* in place of Top-K in Eq. (1). While we rely completely on Top-K for simplicity, a combination of Top-K and summation, with a bias toward the former, is also possible. □

C Architectures & Training Details

C.1 Setup for Experiment 1

Architecture: The encoder of GAE contains a single pooling layer from which the final encoded representation of the input graph is obtained. The architecture of the encoder is as follows;

$$\begin{aligned} X, A &\leftarrow \mathcal{G} \\ X_{\text{in}} &\leftarrow \mathrm{MLP}_{\text{in}}(X) \\ X_{\text{in}} &\leftarrow \mathrm{GCN}_{\text{in}}\,(A, X_{\text{in}}\,) \\ X_{\text{pool}} &\leftarrow \mathrm{POOL}\,(A, X_{\text{in}}) \end{aligned}$$

Suppose the pooled graph is $\mathcal{G}' = (X_{pool}, A_{pool})$, $X \in \mathbb{R}^{N\times d}$ & $X_{out} \in \mathbb{R}^{P\times d}$ where P is the number of pooled nodes. The decoder contains an Unpool layer that upscales the row dimensionality of X_{out} i.e. P to the original node count N. Following is the architecture of the decoder.

$$\begin{aligned} X_{\text{up}} &\leftarrow \mathrm{UNPOOL}\,(X_{\text{pool}}) \\ X_{\text{out}} &\leftarrow \mathrm{GCN}_{\text{out}}\,(A, X_{\text{up}}) \\ X_{\text{out}} &\leftarrow \mathrm{MLP}_{\text{out}}\,(X_{\text{out}}) \end{aligned}$$

Motive is if features in X_{pool} contain hierarchical-level information such that each node feature in X_{pool} aggregates features of nodes under its hierarchy, then by inverting this aggregation we should be able to recover the original features. This aggregation can be expressed as $X_{pool} = S^\top X_{in}$. Then inverse of the aggregation as $X_{up} = U X_{pool}$, where $U = S^{\mp}$ is the pseudo inverse of matrix S. The original adjacency matrix A is used instead of A_{pool} as input by $\mathrm{GCN}_{\text{out}}$ to limit the experiment on the expressivity of node features alone.

Dataset	G_{avg}	C_{avg}	V_{avg}	E_{avg}
D&D	1178	2	284.32	715.66
PROTEINS	1113	2	39.06	72.82

Table 4: Statistics of datasets used for graph classification. G_{avg} ,C_{avg} ,V_{avg} ,E_{avg} denote number of graphs, number of classes, average number of vertices and average number of edges respectively.

Graph	Nodes	Edges	Avg. degree
Airplane	1333	2611	3.91
Car	1920	2372	2.47
Person	3305	9055	5.47

Table 5: Statistics of graphs used for GAE experiment

Training: We train the GAE to minimize MSE between the input and output features with a learning rate of 0.0005, early stopping on the training loss with a 1000 epochs patience and a tolerance of 10^{-6}. Each experiment was conducted 3 times and obtained the average MSE.

C.2 Setup for Experiment 2

Architecture: The architecture contains two blocks of the following. Note that the input to the 2^{nd} block is not X_{readout}, but outputs X_{pool}, A_{pool} of the previous POOL. When no G-JK is used we remove the X_{readout} from the block.

$$
\begin{aligned}
X &\leftarrow \mathrm{GCN}(A, X) \\
A_{\text{pool}}, X_{\text{pool}} &\leftarrow \mathrm{POOL}(A, X) \\
X_{\text{readout}} &\leftarrow \mathrm{AGGR}(X_{\text{pool}}) \,||\, \mathrm{MAX}(X_{\text{pool}})
\end{aligned}
$$

All GCN layers have the same output feature dimension. POOL is either *SAGPool* or *TopKPool*. AGGR is the column-wise *mean* & *sum* of $X_{\text{pool}} \in \mathbb{R}^{P \times d}$ for *G-JK-mean* & *G-JK-sum* respectively. MAX is column-wise max. $||$ denotes concatenation. Tanh is used for gating. Readouts of the two blocks are summed and passed to a final MLP block of the following form: $\text{NN}_h \rightarrow \text{Dropout}_{p=0.5} \rightarrow \text{NN}_{h/2} \rightarrow \text{NN}_c$. c denotes the number of classes. When there are no readouts i.e. no G-JK case, the outputs of the final pooling layer are summed column-wise as a Global Pooling layer (GPL). This is followed by the same MLP block.

Training: All the models are evaluated using 10-fold *stratified* splitting over a single repetition. 10% of training data is used as validation data. Hyperparameters space is- Batch Size: $\{32, 64, 128\}$, Hidden dimension: $\{32, 64, 128\}$, Learning rate: $\{1e^{-2}, 1e^{-3}\}$, L2 regularization: $\{5e^{-4}\}$. Best models are selected using cross-validation on validation data. The number of total epochs is 100k. The model training stops (early) if the validation loss doesn't improve for 50 epochs, and the model is restored to the epoch with the least validation loss on which test data is evaluated.

C.3 Description of Datasets

PROTEINS [16, 18] and D&D [16, 19] datasets are retrieved from the *TUDataset* collection [20]. See Table 4 for dataset statistics. From ModelNet40 [15], sample 151 for *Airplane*, sample 75 for *Car*, sample 83 for *Person* are chosen. See Table 5 for graph statistics.