Graph Pooling Provably Improves Expressivity

Veronica Lachi*
University of Siena
veronica.lachi@student.unisi.it

Alice Moallemy-Oureh*
University of Kassel
amoallemy@uni-kassel.de

Andreas Roth*
TU Dortmund University
andreas.roth@tu-dortmund.de

Pascal Welke*
TU Wien
pascal.welke@tuwien.ac.at

Abstract

In the domain of graph neural networks (GNNs), pooling operators are fundamental to reduce the size of the graph by simplifying graph structures and vertex features. Recent advances have shown that well-designed pooling operators, coupled with message-passing layers, can endow hierarchical GNNs with an expressive power regarding the graph isomorphism test that is equal to the Weisfeiler-Leman test. However, the ability of hierarchical GNNs to increase expressive power by utilizing graph coarsening was not yet explored. This results in uncertainties about the benefits of pooling operators and a lack of sufficient properties to guide their design. In this work, we identify conditions for pooling operators to generate WL-*distinguishable* coarsened graphs from originally WL-*indistinguishable* but non-isomorphic graphs. Our conditions are versatile and can be tailored to specific tasks and data characteristics, offering a promising avenue for further research.

1 Introduction

With an ever-increasing amount of graph data available in many applications and the growing success of neural networks, Graph Neural Networks (GNNs) (Scarselli et al, 2008) have become an active field of research. Real-world problems modeled as graphs can grow to exceedingly large sizes. *Pooling operators* address this challenge and generate coarser versions of given graphs by reducing the number of nodes or edges (Bianchi et al, 2020b; Ying et al, 2018). The pooling operator is not only valuable for reducing the graph's size but also for enabling GNNs to gradually learn more global information, thus facilitating the construction of truly deep GNNs.

However, efficiently and intelligently reducing the size of a graph is not a straightforward task, and assessing the quality of a pooling operator presents its own set of challenges. Several metrics exist for evaluating the quantity and type of information lost during graph reduction (Grattarola et al, 2022; Bianchi et al, 2020a). One intriguing perspective to consider is the problem of Weisfeiler-Leman (WL) equivalence (Leman and Weisfeiler, 1968). The WL test is an iterative algorithm that checks whether two graphs are isomorphic. It is widely employed to investigate the expressive capabilities of graph neural networks. Specifically, GNNs, when formulated with the appropriate message-passing mechanisms, exhibit expressive power that is, at most, equivalent to that of the WL test (Maron et al, 2019; Morris et al, 2019; Xu et al, 2019). In recent years, significant efforts have been made to enhance the expressive power of GNNs. Several alternative GNN architectures have been proposed, such as kGNN (Morris et al, 2019), which draws inspiration from the extension of the WL algorithm to k-tuples of nodes, or ESAN (Bevilacqua et al, 2022), which encodes multisets of subgraphs instead of multisets of node features. Such expressive GNNs, however, usually result in a combinatorial explosion of the input data size. In this work, we explore the potential of increasing the expressive

*All authors contributed equally

NeurIPS 2023 New Frontiers in Graph Learning Workshop (NeurIPS GLFrontiers 2023).

power of GNNs while *reducing* the complexity and amount of computations while maintaining the standard message-passing scheme.

Bianchi and Lachi (2023) were the first to delve into the relationship between pooling and the expressiveness of hierarchical GNNs which are composed of message-passing layers and pooling operators. In particular, they demonstrated the existence of three conditions on the formulation of message-passing layers and pooling layers that are sufficient to guarantee that the overall hierarchical GNN is as expressive as the WL graph isomorphism test.

Up to now, however, no one has explored the possibility of designing a pooling operator capable of *increasing* the expressiveness of GNNs while maintaining a standard message-passing mechanism. We show that some pooling operators can produce coarsened graphs that are distinguishable by the WL test when applied to two non-isomorphic graphs that are indistinguishable by WL. We furthermore define sufficient conditions for the pooling operator to increase expressiveness. Our conditions are general in nature, opening the door to the design of various types of pooling operators that satisfy these criteria.

2 Preliminaries

In this section, we set the notation and define the elementary objects used in this paper. Most of our notation is taken from Bianchi and Lachi (2023). For better readability, we use $\{\cdot\}$ to denote multisets, i.e., unordered collections that allow repeated entries. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with $N = |\mathcal{V}|$ nodes $\mathcal{V}$ and edges $\mathcal{E}$. For a given node v, the set of neighboring nodes is denoted by $\mathcal{N}_v$. Every node is equipped with d-dimensional node features $\mathbf{x} \in \mathbb{R}^d$.

We consider the task of graph classification, i.e., our goal is to find a classifier $\phi\colon \mathcal{G} \mapsto [0,1]^{|\mathcal{C}|}$ that maps each graph to probabilities for each class $c \in \mathcal{C}$. The expressivity of a classifier ϕ regarding their ability to identify graph isomorphisms is crucial to distinguish structurally similar, but unidentical graphs. One such test is the Weisfeiler-Leman test (Leman and Weisfeiler, 1968):

Definition 1 (Weisfeiler-Leman test)**.** *The Weisfeiler-Leman test is an iterative node feature (color) refinement algorithm to test whether two graphs are isomorphic. Let Σ be a set of values representing the colors. At iteration 0, let*

$$c_v^{(0)} = \mathit{HASH}_0(\ell_v)$$

where HASH_0 is a function that bijectively codes every possible feature with a color in Σ. For any iteration $t > 0$, let

$$c_v^{(t)} = \mathit{HASH}((c_v^{(t-1)}, \{c_n^{(t-1)} \;:\; n \in \mathcal{N}_v\}))$$

where HASH injectively maps the above pair to a unique value in Σ, which has not been used in the previous iterations. The algorithm terminates if the number of colors between two iterations does not change, i.e. when there exists a bijection between $\{c_n^{(t-1)} \;:\; n \in \mathcal{V}\}$ and $\{c_n^{(t)} \;:\; n \in \mathcal{V}\}$.

Definition 2 (WL distinguishable)**.** *$\mathcal{G}_1$ and $\mathcal{G}_2$ are **WL distinguishable** ($\mathcal{G}_1 \neq_{WL} \mathcal{G}_2$) if there exists an iteration t for which $\{c_n^{(t)} \;:\; n \in \mathcal{V}_1\} \neq \{c_n^{(t)} \;:\; n \in \mathcal{V}_2\}$.*

Definition 3 (currently WL distinguishable)**.** *Two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are **currently WL distinguishable** ($\mathcal{G}_1 \neq_{CWL} \mathcal{G}_2$) if their color multisets are currently different.*

A Graph Neural Network (GNN) is a common way to instantiate ϕ, as it provides a flexible framework and contains parameters that can be adapted to a given task using gradient descent. GNNs typically follow a message-passing scheme, in which the representation of each node gets updated by combining its previous representation with its neighboring nodes. To allow for the same discriminative capabilities as the WL test, the updated node representations may be transformed using a Multilayer Perceptron (MLP) that can represent injective functions (Xu et al, 2019). Several iterations (or layers) of message-passing are performed until the multisets of updated vertex features (or colors) of two graphs $\mathcal{G}_1, \mathcal{G}_2$ differ if they are WL distinguishable. The Graph Isomorphism Network is one commonly used instantiation that achieves this expressivity in theory (Xu et al, 2019).

In this work, we analyze the capabilities of pooling operators regarding their changes to the identification of non-isomorphic graphs. A pooling operator `POOL` is a function that maps a graph to a potentially smaller graph. We require that $\texttt{POOL}(\mathcal{G}_1)$ is isomorphic to $\texttt{POOL}(\mathcal{G}_2)$ whenever $\mathcal{G}_1$ and $\mathcal{G}_2$ are isomorphic. According to Grattarola et al (2022) most pooling operators $\texttt{POOL} : \mathcal{G} \to \mathcal{G}_P := (\mathcal{V}_P, \mathcal{E}_P)$

can be written as a triplet $(\texttt{SEL}, \texttt{RED}, \texttt{CON})$ of Select-Reduce-Connect functions. The select function $\texttt{SEL}\colon \mathcal{G} \mapsto \mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$ clusters the input graph nodes into so-called supernodes $\mathcal{S}_j = \{s_i^j\}_{i=1}^N$ where s_i^j indicates the contribution of node i on supernode j. The reduce function RED aggregates the features of the nodes assigned to the same supernode. For the resulting graph, the connect function CON generates the edges and edge features, if applicable, by connecting the supernodes.

While various pooling operators have been proposed (Ying et al, 2018; Luzhnica et al, 2019; Bianchi et al, 2020a,b; Fey et al, 2020; Sanders et al, 2023; Tsitsulin et al, 2023), their effects on the expressivity of GNNs regarding the WL test are largely unexplored. To the best of our knowledge, conditions for increasing expressivity have not yet been considered by the community. Only recently, a study by Bianchi and Lachi (2023) introduced the formal concepts needed for a pooling operator to maintain the expressivity of a GNN, which we introduce next.

3 Pooling Maintains Expressivity

A recent work by Bianchi and Lachi (2023) formalized the ability of GNNs utilizing pooling operations to maintain the expressivity given by the original GNN. We start by providing our formal definition for pooling operators which maintain expressivity.

Definition 4 (Maintaining Expressivity). *A pooling operator $\texttt{POOL} = (\texttt{SEL}, \texttt{RED}, \texttt{CON})$ is **maintaining expressivity** if it maps any pair of currently WL-distinguishable graphs to a pair of WL-distinguishable graphs, i.e., if $\mathcal{G}_1 \neq_{CWL} \mathcal{G}_2 \Rightarrow \texttt{POOL}(\mathcal{G}_1) \neq_{WL} \texttt{POOL}(\mathcal{G}_2)$.*

(Bianchi and Lachi, 2023) identified three properties on SEL and RED which are sufficient for a pooling operator to maintain expressivity. We generalize these properties, as we find the key to maintaining expressivity to be the injectivity of the combination of the SEL and RED functions on the multisets of node features. Let $\mathcal{X}_{\mathcal{G}_i}^{\mathrm{WL}} = \{\mathbf{x}_{\mathcal{G}_i}^{\mathrm{WL}}(j) \ : \ j \in \mathcal{V}\}$ be a multiset of WL discriminative features for each graph $\mathcal{G}_i$. We now show that this discriminative power is maintained when using any injective pooling operator.

Proposition 1. *Let $\texttt{POOL} = (\texttt{SEL}, \texttt{RED}, \texttt{CON})$ such that*

$$\texttt{RED} \circ \texttt{SEL} : (\mathcal{X}_{\mathcal{G}_i}^{WL}, \mathcal{G}_i) \mapsto \mathcal{X}_{\texttt{POOL}(\mathcal{G}_i)}^{WL}$$

is injective on $\mathcal{X}_{\mathcal{G}_i}^{WL}$. Then, POOL maintains expressivity.

Proof. If $\texttt{RED} \circ \texttt{SEL}$ is injective, then different node feature multisets $\mathcal{X}_{\mathcal{G}_m}^{\mathrm{WL}} \neq \mathcal{X}_{\mathcal{G}_n}^{\mathrm{WL}}$ are mapped to different pooled node feature multisets $\mathcal{X}_{\texttt{POOL}(\mathcal{G}_m)}^{\mathrm{WL}} = \texttt{RED} \circ \texttt{SEL}(\mathcal{X}_{\mathcal{G}_m}^{\mathrm{WL}}) \neq \texttt{RED} \circ \texttt{SEL}(\mathcal{X}_{\mathcal{G}_n}^{\mathrm{WL}}) = \mathcal{X}_{\texttt{POOL}(\mathcal{G}_n)}^{\mathrm{WL}}$. As a result, expressivity is maintained: The Weisfeiler-Leman test can distinguish the two pooled graphs independent of the choice of CON. In fact, $\texttt{POOL}(\mathcal{G}_1) \neq_{\mathrm{CWL}} \texttt{POOL}(\mathcal{G}_2)$. □

We note that this proof is not specifically formulated for WL, and the expressivity is maintained even when the original features are more discriminative. Bianchi and Lachi (2023) have shown that RED can be chosen as a weighted sum of node features in a cluster if the cluster assignments of each node sum to a fixed constant. In the setting of Proposition 1 we can use the trick proposed by Xu et al (2019) to achive expressiveness: We choose a sufficiently powerful MLP $\texttt{RED}_\theta$ that is able to learn the injective function $(\mathcal{X}, \mathcal{S}) \rightarrow \mathcal{X}_P$. We now show that many choices for SEL do not only result in expressive pooling operators but actually result in pooling operators that *increase expressivity*.

4 Pooling Increases Expressivity

In this section, we present sufficient conditions under which GNNs do not just maintain the expressivity but provably increase the expressivity. We start by providing our general definition:

Definition 5 (Increasing Expressivity). *A pooling operator $\texttt{POOL} = (\texttt{SEL}, \texttt{RED}, \texttt{CON})$ is **increasing expressivity** if it is expressive and if there is a pair of graphs that are WL indistinguishable which become WL-distinguishable after pooling, i.e., if there exist $\mathcal{G}_1 =_{WL} \mathcal{G}_2$ with $\texttt{POOL}(\mathcal{G}_1) \neq_{WL} \texttt{POOL}(\mathcal{G}_2)$.*

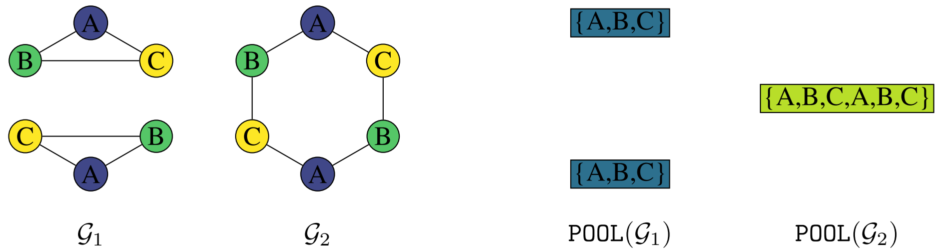

Figure 1: Two WL-indistinguishable graphs $\mathcal{G}_1, \mathcal{G}_2$ which can be distinguished after pooling as in Theorem 1. Clustering cycles maps $\mathcal{G}_1$ to a two supernode graph, while it maps $\mathcal{G}_2$ to a single supernode graph.

In what follows, we assume that the composition of the select and reduce operator is injective, as stated in Prop. 1. We highlight a key property which shows that obtaining different node clusterings for two graphs allows us to construct a pooling method that distinguishes those graphs.

Lemma 1. *Let* `RED` *be an injective function. For any two graphs* $\mathcal{G}_1, \mathcal{G}_2$ *we have*

$$\texttt{SEL}(\mathcal{G}_1) \neq \texttt{SEL}(\mathcal{G}_2) \implies \texttt{POOL}(\mathcal{G}_1) \neq_{WL} \texttt{POOL}(\mathcal{G}_2).$$

Proof. Let $\mathcal{G}_1, \mathcal{G}_2$ be two graphs with $\texttt{SEL}(\mathcal{G}_1) \neq \texttt{SEL}(\mathcal{G}_2)$, i.e., the obtained supernode sets $\{S_1^1, \ldots, S_k^1\} \neq \{S_1^2, \ldots, S_l^2\}$ are not equal. Choosing `RED` to be injective, i.e., different multisets of node representations are mapped to different supernode representations, implies $\texttt{POOL}(\mathcal{G}_1) \neq \texttt{POOL}(\mathcal{G}_2)$. □

This shows us that we can distinguish two pooled graphs by the WL test if we can obtain different cluster assignments for any two graphs. Combining this insight with Prop. 1, we also know that all cases distinguished by WL on the original graph remain distinguishable after pooling the graph. The injectivity of `RED` can be satisfied by utilizing the sum as aggregation and an MLP for feature transformation, as proposed for the GIN (Xu et al, 2019). Thus, selecting a suitable `SEL` function that (a) maps isomorphic graphs to identical cluster assignments and (b) can assign two WL-indistinguishable graphs to different cluster assignments is sufficient to achieve a strictly increased expressivity of a model utilizing pooling. We can achieve this by using an operation that is incomparable to WL or strictly more expressive than WL. We formalize this in the following statement:

Theorem 1. *Let* `SEL` *distinguish some graphs that WL does not distinguish, i.e., let* `SEL` *be incomparable to or more expressive than WL. Then,* `POOL` *can be constructed to increase expressivity.*

Proof. By definition, there exists a pair of graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ that is not distinguishable by WL but can be distinguished by `SEL`. For these two graphs, there exists a cluster assignment such that $\{S_1^1, \ldots, S_k^1\} \neq \{S_1^2, \ldots, S_l^2\}$ are different. Then, the resulting coarsened graphs $\texttt{POOL}(\mathcal{G}_1) \neq_{\text{WL}} \texttt{POOL}(\mathcal{G}_2)$ are different by injectivity as shown in Lemma 1. By our injectivity assumption (Prop. 1), `POOL` maps any two other graphs currently distinguished by WL to WL distinguishable graphs. □

Given this insight, we now have the theoretical confirmation that graph pooling can increase the expressivity of GNNs by utilizing a powerful node selection operator. For example, consider the `SEL` operator that clusters nodes together that lie on the same cycle. It maps the node set of $\mathcal{G}$ to the node sets of the biconnected components of $\mathcal{G}$. It is incomparable to WL, as it can not distinguish nonisomorphic trees of the same size. Fig. 1 shows that the resulting pooling operator can distinguish two triangles from a cycle of length six. We point out that several existing methods benefit from this theoretical foundation, including CliquePool (Luzhnica et al, 2019), CurvPool (Sanders et al, 2023), and many others (Fey et al, 2020).

In the following, we show that we can also increase the expressivity of GNNs even when both the cluster assignments and the corresponding WL discriminative features are equal. Pooling methods can achieve this by exploiting the graph topology and utilizing a suitable `CON` function. This is critical, as this does not require a sophisticated `SEL` function but allows computationally light methods to also increase expressivity. This observation is formalized in the following remark:

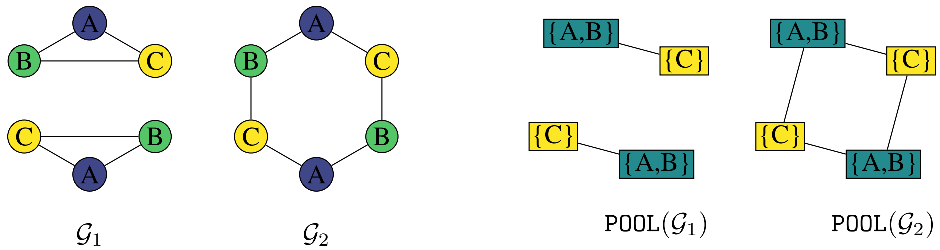

Figure 2: Two WL-indistinguishable graphs $\mathcal{G}_1, \mathcal{G}_2$ which can be distinguished after pooling as in Remark 1. Contracting edges A–B and adding a single superedge between supernodes iff any original nodes were connected results in a disconnected graph for $\mathcal{G}_1$ and a connected graph for $\mathcal{G}_2$.

Remark 1. *Let* $\mathtt{CON}$ *be the function that constructs an edge between two supernodes if any pair of their nodes were connected in the original graph. Then, it can be shown that there are graphs* $\mathcal{G}_1$ *and* $\mathcal{G}_2$ *with*

$$\mathtt{SEL}(\mathcal{G}_1) = \mathtt{SEL}(\mathcal{G}_2) \text{ and } \mathcal{X}^{WL}_{\mathcal{G}_1} = \mathcal{X}^{WL}_{\mathcal{G}_2} \text{ such that } \mathtt{POOL}(\mathcal{G}_1) \neq_{WL} \mathtt{POOL}(\mathcal{G}_2)\,,$$

and $\mathtt{POOL}$ *increases expressivity.*

As an example consider two triangles with all nodes in the triangle having distinct colors, but both triangles are colored equally as $\mathcal{G}_1$*. As* $\mathcal{G}_2$*, consider a hexagon for which the nodes have the same three colors and the neighboring colors are the same for* $\mathcal{G}_1$ *and* $\mathcal{G}_2$*. WL cannot distinguish these graphs, i.e.,* $\mathcal{X}^{WL}_{\mathcal{G}_1} = \mathcal{X}^{WL}_{\mathcal{G}_2}$*. For* $\mathtt{SEL}$*, we choose any pair of node colors and contract all edges corresponding to that pair, i.e.,* $\mathtt{SEL}(\mathcal{G}_1) = \mathtt{SEL}(\mathcal{G}_2)$*. For the triangle, this results in two graphs of two nodes, each with one edge. For the hexagon, this results in a graph that is a cycle of four nodes. WL distinguishes these graphs, i.e.,* $\mathtt{POOL}(\mathcal{G}_1) \neq_{WL} \mathtt{POOL}(\mathcal{G}_2)$*. This scenario is visualized in Fig. 2.*

We note the importance of the CON function for increasing expressivity in this scenario. Transferring *all* edges between nodes in different supernodes to their respective supernode can result in a multigraph. We would not increase expressivity by choosing CON to retain the number of individual edges between nodes from these supernodes, i.e., we consider multigraphs. WL would not distinguish these multigraphs, and expressivity would not be increased, as their resulting WL unfolding trees for two graphs would still be equal. On the other hand, pruning multiple edges to a single edge allows us to distinguish between graphs that WL could not distinguish. We attribute this to multiple edges between two supernodes corresponding to a cycle formed by their individual nodes. When deleting duplicate edges, we then detect that cycle. This allows us to distinguish this graph from other structurally similar graphs, which do not have that same cycle. We visualize such a scenario in Fig. 2. Consequently, pooling methods considering the graph topology when clustering nodes have an advantage in expressivity against pooling methods that only rely on the node features, e.g., DiffPool (Ying et al, 2018), DMoN (Tsitsulin et al, 2023).

We next show that it is not only possible for pooling methods respecting the graph topology to achieve increased expressivity, but this is always achieved when repeated often enough. The intuition is that disconnected components remain disconnected, which WL cannot distinguish in all cases.

Proposition 2. *Any* $\mathtt{POOL}$ *method retaining disconnected components when repeatedly applied until all edges are contracted is increasing expressivity.*

Proof. The ability to maintain expressivity for all graphs follows from Prop. 1. In addition, two triangles are mapped to two nodes, while the hexagon is mapped to a single node. □

Our findings in this section show that pooling methods can increase the expressivity of GNNs. We outlined two directions: First, utilizing a cluster assignment, which can distinguish non-isomorphic graphs that WL cannot, allows the GNN to distinguish additional cases. This holds even when the cluster assignment fails to distinguish some graphs WL could distinguish, as resulting equal graphs are combined with discriminative node representations. Second, even less powerful methods for cluster assignment that cannot distinguish non-isomorphic graphs can increase the expressivity. This result stems from considering the graph topology and the choice of the CON function, as duplicate edges between supernodes correspond to detected cycles between these nodes. Going forward, we aim to further study the maximal achievable expressivity with pooling methods.

5 Conclusion and Future Work

In this work, we established several ways in which pooling can increase the expressivity of GNNs. This can be done by choosing a powerful pool assignment method or by respecting the graph topology and pruning duplicate edges. Notably, this approach marks the first attempt in the literature to enhance expressivity while concurrently reducing the amount of information. Our results provides a valuable theoretical basis for many existing methods and may serve as a guideline for designing simple, more expressive GNNs that utilize pooling.

There are various open questions we want to investigate further. For one, our goal is to precisely quantify the achievable gains in expressivity beyond a general improvement. Second, we want to exploit our insights to adjust existing pooling methods and design novel approaches to achieve better predictive performance, which we plan to evaluate extensively on synthetic and real datasets.

Acknowledgments

AMs work is funded by the Ministry of Education and Research of Germany, under the funding code 01IS20047A, according to the 'Policy for the Funding of Female Junior Researchers in Artificial Intelligence'. ARs work is funded by the Federal Ministry of Education and Research of Germany under grant no. 01IS22094E WEST-AI. PWs work is financed by the Vienna Science and Technology Fund (WWTF) project ICT22-059 StruDL.

References

Beatrice Bevilacqua, Fabrizio Frasca, Derek Lim, Balasubramaniam Srinivasan, Chen Cai, Gopinath Balamurugan, Michael M Bronstein, Haggai Maron (2022) Equivariant subgraph aggregation networks. In: International Conference on Learning Representations, URL https://openreview.net/forum?id=dFbKQaRk15w

Filippo Maria Bianchi, Veronica Lachi (2023) The expressive power of pooling in graph neural networks. In: Advances in Neural Information Processing Systems, DOI 10.48550/arXiv.2304.01575

Filippo Maria Bianchi, Daniele Grattarola, Cesare Alippi (2020a) Spectral clustering with graph neural networks for graph pooling. In: International Conference on Machine Learning

Filippo Maria Bianchi, Daniele Grattarola, Lorenzo Livi, Cesare Alippi (2020b) Hierarchical representation learning in graph neural networks with node decimation pooling. IEEE Transactions on Neural Networks and Learning Systems 33(5):2195–2207

Matthias Fey, Jan-Gin Yuen, Frank Weichert (2020) Hierarchical inter-message passing for learning on molecular graphs. In: Graph Representation Learning and Beyond Workshop, DOI 10.48550/arXiv.2006.12179

Daniele Grattarola, Daniele Zambon, Filippo Maria Bianchi, Cesare Alippi (2022) Understanding pooling in graph neural networks. IEEE Transactions on Neural Networks and Learning Systems

AA Leman, B Weisfeiler (1968) A reduction of a graph to a canonical form and an algebra arising during this reduction. Nauchno-Technicheskaya Informatsiya 2(9):12–16

Enxhell Luzhnica, Ben Day, Pietro Lio (2019) Clique pooling for graph classification. In: Representation Learning on Graphs and Manifolds Workshop, DOI 10.48550/arXiv.1904.00374

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, Yaron Lipman (2019) Provably powerful graph networks. In: Advances in Neural Information Processing Systems

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, Martin Grohe (2019) Weisfeiler and leman go neural: Higher-order graph neural networks. In: AAAI Conference on Artificial Intelligence, DOI 10.1609/aaai.v33i01.33014602

Cedric Sanders, Andreas Roth, Thomas Liebig (2023) Curvature-based pooling within graph neural networks. In: Mining and Learning with Graphs Workshop, DOI 10.48550/arXiv.2308.16516

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, Gabriele Monfardini (2008) The graph neural network model. IEEE Transactions on Neural Networks 20(1):61–80

Anton Tsitsulin, John Palowitch, Bryan Perozzi, Emmanuel Müller (2023) Graph clustering with graph neural networks. Journal of Machine Learning Research 24(127):1–21

Keyulu Xu, Weihua Hu, Jure Leskovec, Stefanie Jegelka (2019) How powerful are graph neural networks? In: International Conference on Learning Representations, URL https://openreview.net/forum?id=ryGs6iA5Km

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, Jure Leskovec (2018) Hierarchical graph representation learning with differentiable pooling. In: Advances in Neural Information Processing Systems