WEISFEILER AND LEMAN GO INFINITE: SPECTRAL AND COMBINATORIAL PRE-COLORINGS

Or Feldman[1], **Amit Boyarski**[1], **Shai Feldman**[1], **Dani Kogan**[1], **Avi Mendelson**[1], **Chaim Baskin**[1]
[1] Technion – Israel Institute of Technology

ABSTRACT

Two popular alternatives for graph isomorphism testing that offer a good trade-off between expressive power and computational efficiency are combinatorial (i.e., obtained via the Weisfeiler-Leman (WL) test) and spectral invariants. While the exact power of the latter is still an open question, the former is regularly criticized for its limited power, when a standard configuration of uniform pre-coloring is used. This drawback hinders the applicability of Message Passing Graph Neural Networks (MPGNNs), whose expressive power is upper bounded by the WL test. Relaxing the assumption of uniform pre-coloring, we show that one can increase the expressive power of the WL test ad infinitum. Following that, we propose an efficient pre-coloring based on spectral features that provably increases the expressive power of the vanilla WL test. The code to reproduce our experiments is available at `https://github.com/TPFI22/Spectral-and-Combinatorial`.

1 INTRODUCTION

Despite their success, Message Passing Graph Neural Networks (MPGNNs) are bounded in their expressive power (i.e., two different graphs may be encoded to the same descriptor by the same MPGNN). In fact, it is known that any two graphs that pass the WL test (described in detail in section 2) will be encoded by the same descriptor Xu et al. [2018]. For example, MPGNNs cannot distinguish between the Decalin and Bicyclopentyl molecules graphs (Figure 2) although their graphs are non-isomorphic Sato [2020]. Attempts have been made to improve the expressive power of MPGNNs by suggesting new and arguably complicated GNN architectures that are not bounded by the Weisfeiler-Leman (WL) test, e.g., by using high order networks, generalizing graphs to simplicial complexes, etc.

We propose a new and general approach to improve the expressivity of MPGNNs. This approach is based on the traditional and relatively simple MPGNN architectures and does not require them to be changed at all. To that end, we suggest pre-coloring the nodes of a graph with an informative equivariant coloring, i.e., equivariant node features that are precomputed before the MPGNNs' learning process. We present a rigorous proof that this method can be used to improve the expressiveness of the WL test an infinite number of times. In addition, we present an instance of an equivariant coloring based on the spectral decomposition of the graph Laplacian that is also efficient to compute, explainable, and generates constant size features with respect to the graph size. Figure 1 shows an example of the coloring of the Decalin and Bicyclopentyl molecules graphs with our suggested spectral pre-coloring, and the relatively simple degree pre-coloring. The example shows that the pair of graphs can be distinguished easily when using the spectral coloring compared to the degree coloring.

Contributions.

- We prove that the expressive power of WL can be improved ad infinitum by a sequence of equivariant pre-colorings and that each of the latter can be computed in polynomial time. Thus, the upper bound of the existing MPGNNs can be improved accordingly.
- We suggest expressive and informative pre-coloring based on the spectral decomposition of the graph Laplacian, and explicitly prove that it improves the expressivity of the vanilla WL.

2 Preliminaries

Graph coloring. Graph coloring is a mapping from a vertex and its graph to a label (color), from a known set of labels. We say that coloring C *refines* coloring D if for any two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, and for any two vertices $v_1 \in V_1$, $v_2 \in V_2$ s.t. $C(v_1) = C(v_2)$, $D(v_1) = D(v_2)$. Ideally, we would like to find the following coloring: for each $v_1 \in V_1$,$v_2 \in V_2$, $C(v_1) = C(v_2) \iff$ there exists isomorphism $\sigma : \mathcal{G}_1 \to \mathcal{G}_2$ s.t., $\sigma(v_1) = v_2$.

K-WL test. The WL test of isomorphism is an algorithm for testing a necessary but insufficient condition for graph isomorphism. Two graphs that do not pass the test are necessarily non-isomorphic. First, the algorithm assigns to each node the same color using the constant coloring $C^0_{WL}(v) =$ CONST. Then the algorithm continues with iterations. At each iteration i, each node receives its neighbors' colors and together with its own color, it generates a new color for the next iteration, i.e., $C^i_{WL}(v) = (C^{i-1}_{WL}(v), \{\{C^{i-1}_{WL}(x) | x \in \mathcal{N}(v)\}\})$, where '$\{\{\}\}$' denotes a multi-set, and $\mathcal{N}(v)$ denotes the set of neighbors of v. This process continues until convergence whereupon the colors are collected into a histogram. If the two graphs have different histograms, they failed the test and are called distinguishable. If after the convergence, the two histograms are the same, the graphs did not fail. Having thus passed the test, they are called indistinguishable. It was proved in Bevilacqua et al. [2021] that C^{i+1}_{WL} always *refines* C^i_{WL}. The WL test can be extended to K-tuple coloring instead of vertex (1-tuple) coloring. This extension is called the K-WL test. It was proved in Cai et al. [1992] that any pair of graphs that are indistinguishable by k+1-WL are also indistinguishable by K-WL. Moreover, for any K$\geq$2, there exists a pair of graphs s.t. they are distinguishable by k+1-WL but indistinguishable by k-WL, i.e., k+1-WL is *strictly more expressive* than k-WL, for K$\geq$2. The diagonal k-WL coloring on the graph vertices is defined to be $\Delta(k - WL)(v) = C_{k-WL}(v, ..., v)$ where C_{k-WL} is the coloring after k-WL converges. It was proven in Rattan & Seppelt [2021] that Δ(k+1-WL) *refines* Δ(k-WL).

Heat kernel. The heat kernel matrix describes the process of heat diffusion on the graph through time. The heat kernel at time t for graph $\mathcal{G} = (V, E)$ is a $|V| \times |V|$ matrix where the element at the index (u, v) is defined to be $H_t(u, v) = \Sigma^{|V|}_{i=1} e^{-\lambda_i t}\phi_i(u)\phi_i(v)$ where λ_i is the i-th eigenvalue of the graph Laplacian and ϕ_i is its corresponding eigenvector. $H_t(u, v)$ is the amount of heat transferred from node u to node v until time t. When the observed point in time t tends to zero, the kernel is affected mostly by the local structures of the graphs. When the observed time point is relatively large, the global structure of the graphs becomes the dominant structure.

3 Expressive power of 1-WL with pre-colorings

In section 2 we noted that the expressive power of 1-WL is limited. In particular, it is strictly limited by the expressive power of 3-WL. In this section we present a method to improve the expressive power of 1-WL using pre-coloring, i.e., coloring the graph before the iteration phase of 1-WL. If we pre-color 1-WL with coloring C, we mark the new algorithm as 1-CWL.

Theorem 1. *Let R_1, R_2 be two colorings s.t. R_2 refines R_1 and R_2 is permutation equivariant. Accordingly, 1-R_2WL is at least as expressive as 1-R_1WL.*

For R_1 and R_2 that satisfy Theorem 1, it is enough to find a single pair of graphs that are indistinguishable by 1-R_1WL but distinguishable by 1-R_2WL in order to prove strictness in expressive power.

Theorem 2. *Let $\mathcal{G}_1$,$\mathcal{G}_2$ be any two graphs. Their Δ(k-WL) histograms are equal $\iff$ their C_{k-WL} histograms are equal.*

Theorem 3. *For any $K \geq 2$, 1-Δ(k+1-WL)WL is strictly more expressive than 1 -Δ(k-WL)WL.*

The meaning of this theorem is that the expressive power of MPGNNs, which is provingly bounded by the expressive power of 1-WL, can be improved ad infinitium in the WL hierarchy using the right permutation equivariant pre-coloring as a pre-process before the MPGNN learning phase. According to Theorem 3, the coloring can be obtained via the computation of Δ(k-WL).

In section 4 we give another example of such pre-coloring based on spectral features.

Not every permutation equivariant coloring C makes 1-CWL strictly more expressive than 1-WL.

Example 3.1. *If $D(u) = |\mathcal{N}(u)|$, i.e., the degree coloring, then 1-DWL is equal to 1-WL in terms of expressive power.*

4 Spectral pre-coloring

Spectral WL. We propose an expressive pre-coloring based on the graph spectrum, which can be used to color the nodes instead of the constant coloring of the 1-WL algorithm. We will call this variant the *spectral WL algorithm.* To calculate the pre-coloring, we first compute m heat kernel matrices for evenly spaced points in time on the logarithmic scale. Then for each node u, we give the following color: $(H_{t_1}(u,u), ..., H_{t_m}(u,u))$. Finally, we choose a constant amount of quantiles r from the row of u (ignoring the element on the diagonal) and append them in ascending order, e.g., $((q^{t_1}_{1_u}...q^{t_1}_{r_u}), ...(q^{t_m}_{1_u}...q^{t_m}_{r_u}))$, to the existing color of the node, to create the final coloring. In the example of the spectral coloring in Figure 1, nodes that have the same color have the same spectral features with $m = 1$, $t = 1$ and no quantiles. This simple setting is sufficient in order to compute the ideal equivariant coloring of the graphs.

Theorem 4. *Spectral WL is strictly more expressive than 1-WL.*

Spectral features for GNNs. This pre-coloring can be used to create initial node features for MPGNNs as a pre-process before the learning phase. Instead of applying the coloring we can append it to the existing node features of any graph. As hinted by Theorem 4, in section 5 we will see that it is enough to add a relatively small feature vector, e.g., with 10 entries, to achieve great expressivity even for real world graphs with hundreds and thousands of nodes. One can, however, *refine* the pre-processing by adding more quantiles and time samples. The features that we added to each node have the desirable property of being explainable, and they have the following meaning: For node u, the feature at entry $i \leq m$ is the amount of heat left at u at time t_i from the beginning of a diffusion process where all the nodes had 0 heat and u had exactly 1. The features at entries $i > m$ represent the distribution of the heat diffusion through time on the other nodes.

5 Experimental study on synthetic benchmarks

To demonstrate the improvement in expressivity that the spectral features add, we built two benchmarks, each of which is based on a single pair of graphs. The first pair of graphs is the Decalin and Bicyclopentyl molecule graphs that have the same 1-WL histogram Sato [2020], but their spectrum is different. The second pair of graphs are distinguishable by 1-WL but cospectral with respect to the Laplacian. Figures of the graphs can be found in the appendix. For each benchmark, we created 1000 examples by adding or removing a single edge at random from the original graphs and reordering their node indices randomly. For each benchmark, we split all the instances into training and test sets with ratio at a 9:1. The goal of a classifier for the benchmark is: Given a graph from the test set, identify the original graph from which it was perturbed. We trained GIN Xu et al. [2018], GCN Kipf & Welling [2016], GraphSAGE Hamilton et al. [2017] and GAT Veličković et al. [2017] and their appropriate Spectral Pre-processed (SP) classifiers with the same settings of five message passing layers, a hidden dimension of 64, a learning rate of 0.01 and spectral features from 10 points in time using only the maximum quantile, for 100 epochs. We repeated the experiment 100 times and report in subsection C.1 the average accuracy and standard deviation of each classifier.

As expected, for the 1-WL indistinguishable pair of graphs, the MGNNs struggle to identify the source of each graph, because 1-WL cannot differentiate between the sources. The spectral features help them to overcome this issue easily. GIN, which has the most expressive aggregation operation among all the MPGNNs, achieves great accuracy on the cospectral graphs; the other MPGNNs, however, do not. These results make sense, since cospectral graphs have common structural properties. In Figure 4 and Figure 5 we can see the spectral coloring of the cospectral graphs introduced by the spectral pre-prossessing – nodes with the same color have the same spectral features. In Figure 4 the pre-processing does not use any quantiles and in Figure 5 the pre-processing uses only the maximum quantile. We can see that not only do both colorings *strictly refine* the constant coloring, but that the coloring that uses the maximum quantile *strictly refines* the one that does not.

6 Evaluation on real benchmarks

We evaluate our pre-processing method on two graph learning tasks: graph classification and node classification. For each task we used four types of GNNs (GIN, GCN, GraphSAGE and GAT) from the Pytorch Geometric framework Fey & Lenssen [2019] to compare the standard use of the network to our SP method.

6.1 Graph classification

We used nine graph classification benchmarks for this task: five social network datasets (COLLAB, IMDB-BINARY, IMDB-MULTI, REDDITBINARY and REDDIT-MULTI5K), three molecule datasets (MUTAG, PTC, NCI1) and a dataset from the field of bioinformatics (PROTEINS) Yanardag & Vishwanathan [2015]. The task of the benchmarks here is to achieve the highest average validation accuracy with 10-fold cross-validation. We used GNNs with five layers where in each layer's MLP a single hidden layer was used. We used concatenation to create the final graph descriptor and a linear layer to create the final output. We fine-tuned the dropout of the linear layer to be one of $\{0,0.5\}$. For the bioinformatics and molecule datasets, we fine-tuned the hidden dimension of all the MLPs to be one of $\{16,32\}$, while for the social network benchmarks we consistently used a hidden dimension of size 64. The number of epochs that achieved the best cross-validation accuracy, averaged over the 10 folds, was selected. We examined 700 epochs for each configuration. For the SP-MPGNNs we chose the best out of the following two: 1. Sampling 10 points in time and not using quantiles at all; 2. Sampling 5 points in time and using the maximum quantile. We reported in subsection C.2 the average validation accuracy and standard deviation over 10 folds.

In general, the SP-MPGNNs performed better than the MPGNNs, especially on the social network benchmarks that contains no initial features for the nodes.

6.2 Node classification

We used four node classification benchmarks for this task: three citation network datasets (Cora, CiteSeer and PubMed) Yang et al. [2016] and a biochemistry dataset (PPI) Zitnik & Leskovec [2017]. The task of the benchmarks here is to achieve the highest average test accuracy upon 100 random initializations of the GNNs. For the citation networks, only the number of message passing layers, the hidden dimension of the MLPs and the number of training epochs, were fine-tuned, using the validation set. The number of the layers was one of $\{2, 3, 4\}$, the hidden dimension was one of $\{128, 256, 384, 512\}$ and each model was trained for at most 200 epochs. Specifically for PPI, there were two layers, the hidden dimension was 512 and the models were trained for 800 epochs. The spectral pre-process was calibrated exactly as in the graph classification evaluation. We repeated each training-testing session 100 times and report in subsection C.3 the average accuracy and standard deviation of the test set.

Even though each node in the benchmark contains a feature vector with hundreds of entries, appending to it a relatively small number of spectral features usually improved the accuracy of the MPGNNs. This can be explained by the fact that the spectral features also contain global information about the graph and the node's position according to it. This information cannot be learned using a small amount of message passing iterations.

7 Discussion

In this work we demonstrated how one can strictly improve the expressive power of the WL test an infinite number of times in the WL hierarchy using the diagonal coloring of the k-WL algorithm, and simultaneously improve the upper bound for MPGNNs, without any change in their architecture. We also proposed spectral pre-processing for MPGNNs that is based on the diagonal and quantiles of the heat kernel matrix. From the results of the graph classification and node classification benchmarks, we conclude that our method of pre-processing improves the performance of MPGNNs on real world graph-structured data.

References

Beatrice Bevilacqua, Fabrizio Frasca, Derek Lim, Balasubramaniam Srinivasan, Chen Cai, Gopinath Balamurugan, Michael M Bronstein, and Haggai Maron. Equivariant subgraph aggregation networks. *arXiv preprint arXiv:2110.02910*, 2021.

Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

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

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4602–4609, 2019.

Gaurav Rattan and Tim Seppelt. Weisfeiler–leman, graph spectra, and random walks. *arXiv preprint arXiv:2103.02972*, 2021.

Ryoma Sato. A survey on the expressive power of graph neural networks. *arXiv preprint arXiv:2003.04078*, 2020.

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

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Pinar Yanardag and S. V. N. Vishwanathan. Deep graph kernels. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pp. 40–48. PMLR, 2016.

Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):i190–i198, 2017.

A Proofs

In the following proofs we assume the definition of the k-WL as defined in Morris et al. [2019]. C^0_{k-WL} is defined to be equal between any two tuples of vertices from $\mathcal{G}_1$ and $\mathcal{G}_2$, if and only if the two subgraphs of $\mathcal{G}_1$ and $\mathcal{G}_2$ comprising all the vertcies in each tuple are isomorphic. We first define the multiset for iteration i at index j to be $c^{i,j}_{k-WL}(v_1,...v_k) = \{\{C^{i-1}_{k-WL}(v_1,...v_{j-1},w,v_{j+1},...v_k)|w \in V\}\}$. Finally, we define the k-WL coloring at iteration i on tuple s to be $C^i_{k-WL}(s) = (c^{i,1}_{k-WL}(s),...c^{i,K}_{k-WL}(s))$.

A.1 Theorem 1 proof

Proof. 1. Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two isomorphic graphs where $\sigma : V_1 \to V_2$ is the isomorphism. We will prove by induction that after n message passing iterations of 1-WL initilized with permutation equivariant coloring, R_2, the coloring of every pair $v \in V_1$ and $u \in V_2$ s.t. $\sigma(v) = u$ is the same.

Base (n=0): R_2 is permutation equivarinat and hence by its definition $R_2(v) = R_2(u)$ for each $v \in V_1$ and $u \in V_2$ s.t. $\sigma(v) = u$.

Step: From the induction assumption we know that every two nodes $v \in V_1$ and $u = \sigma(v) \in V_2$ have the same color after n message passing iterations of 1-WL. For each such v and u we will look at the coloring after the $n+1$ iteration of 1-WL. These are equal to $(C^n_{1-R_2WL}(v), \{\{C^n_{1-R_2WL}(x)|x \in \mathcal{N}(v)\}\})$ and $(C^n_{1-R_2WL}(u), \{\{C^n_{1-R_2WL}(x)|x \in \mathcal{N}(u)\}\})$, respectively. $C^n_{1-R_2WL}(u)$ and $C^n_{1-R_2WL}(v)$ are equal from the induction assumption. σ is an isomorphism and hence $x \in \mathcal{N}(v) \iff \sigma(x) \in N(u)$ and, therefore, $\{\{C^n_{1-R_2WL}(x)|x \in \mathcal{N}(v)\}\}$ and $\{\{C^n_{1-R_2WL}(x)|x \in \mathcal{N}(u)\}\}$ are equal. Since σ is a bijection, we get that the coloring histogram of $\mathcal{G}_1$ and $\mathcal{G}_2$ is the same for each n.

2. Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be any two graphs and let R_1, R_2 be two initial colorings for 1-WL s.t. R_2 *refines* R_1. We will prove by induction that for each $v \in V_1$ and $u \in V_2$, s.t. $C^n_{1-R_2WL}(v) = C^n_{1-R_2WL}(u)$, u, v also satisfy $C^n_{1-R_1WL}(v) = C^n_{1-R_1WL}(u)$ for any number n of 1-WL message passing iterations. Therefore, if $C^n_{1-R_1WL}(v) \neq C^n_{1-R_1WL}(u)$ then $C^n_{1-R_2WL}(v) \neq C^n_{1-R_2WL}(u)$.

Base (n=0): For any $v \in V_1$ and $u \in V_2$, if $C^0_{1-R_2WL}(v) = C^0_{1-R_2WL}(u)$ then $C^0_{1-R_1WL}(v) = C^0_{1-R_1WL}(u)$ since R_2 *refines* R_1.

Step: Let $v \in V_1$ and $u \in V_2$ be any two vertices s.t. $C^{n+1}_{1-R_2WL}(v) = C^{n+1}_{1-R_2WL}(u)$. Their coloring in the $n+1$ iteration is equal to $(C^n_{1-R_2WL}(v), \{\{C^n_{1-R_2WL}(x)|x \in \mathcal{N}(v)\}\})$ and $(C^n_{1-R_2WL}(u), \{\{C^n_{1-R_2WL}(x)|x \in \mathcal{N}(u)\}\})$, respectively. From the induction assumption we find that $C^n_{1-R_1WL}(v) = C^n_{1-R_1WL}(u)$. In addition, we know that the two multisets in the second part of the tuples are equal, this means that there exists an injective mapping $\mu : \mathcal{N}(u) \to \mathcal{N}(v)$ s.t. $C^n_{1-R_2WL}(x) = C^n_{1-R_2WL}(\mu(x))$ and hence by the induction assumption $\{\{C^n_{1-R_1WL}(x)|x \in \mathcal{N}(v)\}\} = \{\{C^n_{1-R_1WL}(x)|x \in \mathcal{N}(u)\}\}$ and therefore $C^{n+1}_{1-R_1WL}(v) = C^{n+1}_{1-R_1WL}(u)$.

If $\mathcal{G}_1$ and $\mathcal{G}_2$ are 1-R_1WL distinguishable they have different 1-R_1WL histograms after some iteration n. Hence, there does not exist an injective mapping $\mu : V_1 \to V_2$ s.t. $C_{1-R_1WL}(x) = C_{1-R_1WL}(\mu(x))$ for any $x \in V_1$. From the claim proved by induction there does not exist an injective mapping $\mu : V_1 \to V_2$ s.t. $C_{1-R_2WL}(x) = C_{1-R_2WL}(\mu(x))$ for any $x \in V_1$. Therefore $\mathcal{G}_1$ and $\mathcal{G}_2$ have different $1 - R_2$WL histograms and are distinguishable by 1-R_2WL.

□

A.2 Theorem 2 proof

Proof. Given $\mathcal{G}_1$ and $\mathcal{G}_2$ s.t. $\{\{\Delta(\text{3-WL})(v)|v \in V_1\}\} = \{\{\Delta(\text{3-WL})(v)|v \in V_2\}\}$ we will prove that $\{\{C_{3-WL}(x,y,z)|x,y,z \in V_1\}\} = \{\{C_{3-WL}(x,y,z)|x,y,z \in V_2\}\}$.

From the equality of the diagonal colorings histogram we know that there is an injective mapping $\mu : V_1 \rightarrow V_2$ s.t. for any $v \in V_1$, Δ(3-WL)$(v) = \Delta$(3-WL)$(\mu(v))$. From structure of C_{3-WL} we know that $\{\{C^{n-1}_{3-WL}(v, v, z)|z \in V_1\}\} = \{\{C^{n-1}_{3-WL}(\mu(v), \mu(v), z)|z \in V_2\}\}$ for any $v \in V_1$. Hence, there is an injective mapping $\mu_2 : V_1 \times V_1 \rightarrow V_2 \times V_2$ s.t. for any $u, v \in V_1$, $C^{n-1}_{3-WL}(v, v, u) = C^{n-1}_{3-WL}(\mu_2(v, u)_1, \mu_2(v, u))$, and again from the structure of C_{3-WL}, the following exists $\{\{C^{n-2}_{3-WL}(v, y, u)|y \in V_1\}\} = \{\{C^{n-2}_{3-WL}(\mu_2(v, u)_1, y, \mu_2(v, u)_2)|y \in V_2\}\}$ for any $u, v \in V_1$. Hence $\{\{C_{3-WL}(x, y, z)|x, y, z \in V_1\}\} = \{\{C_{3-WL}(x, y, z)|x, y, z \in V_2\}\}$, since the 3-WL algorithm converges and we assume it converges after n-2 iterations. The proof can be generalized easily to any K.

Given $\mathcal{G}_1$ and $\mathcal{G}_2$ s.t. $\{\{C_{k-WL}(v_1, ...v_k)|v_1, ...v_k \in V_1\}\} = \{\{C_{k-WL}(v_1, ...v_k)|v_1, ...v_k \in V_2\}\}$, we will prove that $\{\{\Delta$(k-WL)$(v)|v \in V_1\}\} = \{\{\Delta$(k-WL)$(v)|v \in V_2\}\}$. From the initialization of k-WL we know that the color of each tuple of the form $(v, .., v)$ is equal only to other tuples of this form since they are the only ones that represents a graph with a single vertex. Hence, for any $v \in V_1$ and $u_1, u_2, ...u_k \in V_2$ if $C_{k-WL}(v, ..., v) = C_{k-WL}(u_1, ..., u_k)$; then necessarily $u_1 = u_2 = ... = u_k$. Since any $v \in V_1$ is injectively mapped to $u \in V_2$ with the same diagonal coloring, we get that $\{\{\Delta$(k-WL)$(v)|v \in V_1\}\} = \{\{\Delta$(k-WL)$(v)|v \in V_2\}\}$. □

A.3 THEOREM 3 PROOF

From Theorem 1 it immediately is derived that 1-Δ(k+1-WL)WL is as expressive at least as 1-Δ(k-WL)WL. To show that this inequality is strict, we will find a pair of graphs for each $K \geq 2$ s.t. they are indistinguishable by 1-Δ(k-WL)WL but distinguishable by 1-Δ(k+1-WL)WL. For any $K \geq 2$ we know there exists $\mathcal{G}_1$ and $\mathcal{G}_2$ s.t. they are distinguishable by k+1-WL and indistinguishable by k-WL. From Theorem 2 we know that this pair of graphs is also distinguishable by the 1-Δ(k+1-WL)WL algorithm. We also know from Theorem 2 that the Δ(k-WL) histograms of the graphs are equal. We will prove that the 1-Δ(k-WL)WL histograms of the graphs are also equal by showing that the message passing iterations of 1-WL does not change the nodes colors except for the marking/representation of the colors, i.e., the message passing iterations of the 1-WL does not add any new information to the coloring. After a single iteration of 1-Δ(k-WL)WL, the new coloring of any vertex v is $(\Delta(k - WL)(v), \{\{\Delta(k - WL)(u)|u \in \mathcal{N}(v)\}\})$, i.e., the new information added to the coloring is the coloring histogram of the neighbors. We will show that this information can be derived from $\Delta(k - WL)(v)$ for any v. From the initialization of k-WL we can find any color of a tuple $(v, v, ...u)$ such that $u \in \mathcal{N}(v)$ since their representing graphs are isomorphic and different from the representing graphs for $(v, v, ...x)$ where $x \notin \mathcal{N}(v)$. In this way we can find any color of a tuple $(v, u, ...u)$ s.t. $u \in \mathcal{N}(v)$. Again from the initialization of k-WL we can find the color of any $(u, u, ...u)$ s.t. $u \in \mathcal{N}(v)$.

Since the coloring of 1-Δ(k-WL)WL does not change in any iteration and because the coloring histograms are equal from the beginning, 1-Δ(k-WL)WL cannot distinguish between the pair of graphs.

A.4 EXAMPLE 1 PROOF

Proof. We will prove that $C^1_{1-WL} \equiv D$, i.e., the coloring generated after a single iteration of 1-WL initialized with constant coloring equals D. For any vertex v, it is colored with the following coloring: $(C^0_{1-WL}(v), \{\{C^0_{1-WL}(x)|x \in \mathcal{N}(v)\}\}) = (CONST, \{\{CONST, CONST, ...CONST\}\})$ where the multiset size is equal to the size of $\mathcal{N}(v)$. Hence $C^1_{1-WL} \equiv D$.

□

A.5 THEOREM 4 PROOF

Proof. From Theorem 1 it is immediately derived that *Spectral WL* is as expressive at least as 1-WL since the spectral pre-coloring is permutation equivariant and any coloring *refines* the constant coloring. We will show that there exist two graphs that are indistinguishable by 1-WL but distinguishable by *Spectral WL* and hence *Spectral WL* is strictly more expressive than 1-WL.

Coloring Histogram						
Color Graph	0.1914	0.1929	0.2891	0.291	0.3078	0.3098
$\mathcal{G}_1$	2	0	4	0	4	0
$\mathcal{G}_2$	0	2	0	4	0	4

Table 1: Coloring histograms after initialization of *Spectral WL*

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be the graphs representing the Decalin and Bicyclopentyl molecules (Figure 2). It was previously shown that $\mathcal{G}_1$ and $\mathcal{G}_2$ are not isomorphic but cannot be distinguished by the 1-WL test Sato [2020] . Their *Spectral WL* histograms using $m = 1$ with $t = 1$ and $r = 0$ after the initialization phase are shown in Table 1. Since these histograms are different, *Spectral WL* will determine that these graphs are not isomorphic.

□

B Graphs Figures

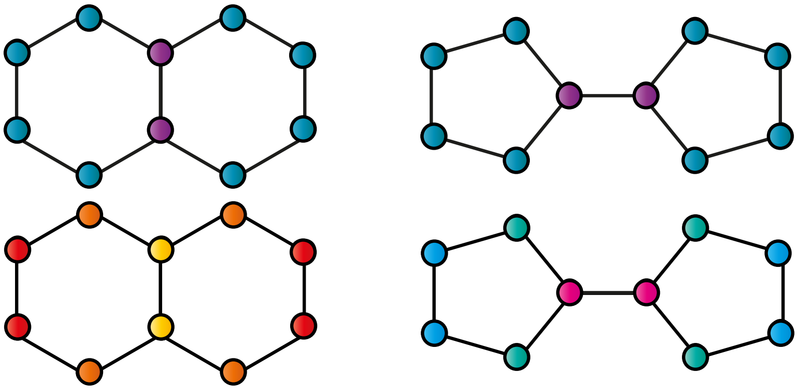

Figure 1: The pair of graphs as colored by the degree coloring (upper) and the spectral coloring (lower).

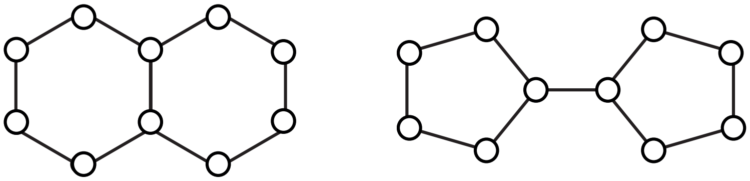

Figure 2: First pair of the original graphs. 1-WL indistinguishable but not cospectral.

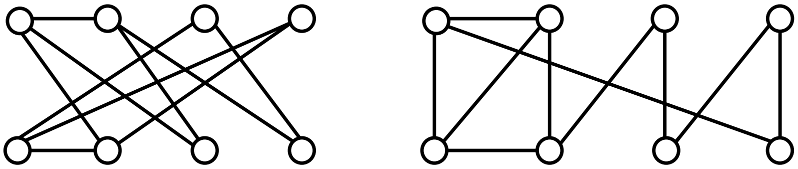

Figure 3: Second pair of the original graphs. 1-WL distinguishable but cospectral with respect to the Laplacian.

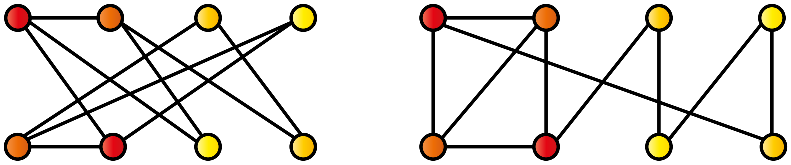

Figure 4: Cospectral graph coloring based only on the diagonal of the heat kernel.

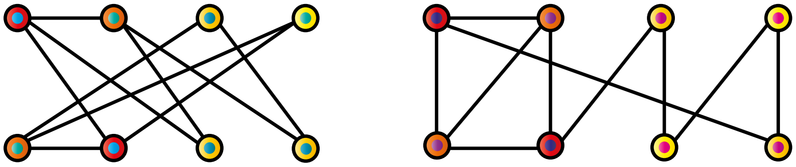

Figure 5: Cospectral graph coloring based on the diagonal of the heat kernel and the maximum quantile.

C Results

C.1 Experimental study results

Table 2: Experimental study results

GNN / Test set	**1-WL indistinguishable**	**Cospectral**
GIN	64±4	93±2
SP-GIN	**99±5**	93±4
GCN	51±4	73±16
SP-GCN	**98±6**	**92±5**
GAT	50±0	49±0
SP-GAT	**97±11**	**77±16**
GraphSAGE	49±0	49±0
SP-GraphSAGE	**95±12**	**91±6**

C.2 Graph classification results

Table 3: Graph classification results – Molecules and bioinformatics

Method	MUTAG	PTC	PROTEINS	NCI1
GIN	88±7	66±8	75±3	82±1
SP-GIN	91±6	66±7	76±3	82±1
GCN	83±6	67±6	75±3	82±1
SP-GCN	**91±6**	68±8	75±3	81±1
GAT	80±9	66±9	75±3	81±1
SP-GAT	**90±5**	68±6	75±4	81±1
GraphSAGE	83±8	65±7	73±4	82±1
SP-GraphSAGE	**91±7**	65±6	73±4	82±1

Table 4: Graph classification results – Social networks

Method	COLLAB	IMDB-B	IMDB-M	REDDIT-B	REDDIT-M
GIN	70±1	73±3	50±3	78±2	54±1
SP-GIN	**77±1**	73±4	51±4	**86±2**	**57±2**
GCN	76±1	65±3	41±3	90±1	55±1
SP-GCN	77±2	**74±4**	**50±4**	91±1	56±1
GAT	42±10	52±3	36±2	71±4	32±5
SP-GAT	**74±2**	**73±4**	**50±4**	**91±2**	**56±1**
GraphSAGE	40±9	52±3	36±2	73±3	35±2
SP-GraphSAGE	**77±2**	**73±3**	**50±4**	**91±1**	**57±2**

C.3 Node classification results

Table 5: Node classification results

Method	CiteSeer	Cora	PubMed	PPI
GIN	71.9±0.6	81.8±0.5	79.6±0.5	91.1±0.2
SP-GIN	71.3±0.6	81.9±1.8	78.8±0.7	91.4±0.2
GCN	63.5±4.4	78.1±2.6	80.4±0.5	88.8±0.1
SP-GCN	**72.1±0.8**	**82.3±1.4**	80.8±0.4	**89.2±0.1**
GAT	64.1±4.5	81.6±1.0	79.9±1.3	79.6±0.2
SP-GAT	**72.3±1.5**	79.2±1.9	80.4±0.7	**80.7±0.3**
GraphSAGE	72.8±0.6	82.9±0.9	80.2±0.6	95.8±0.1
SP-GraphSAGE	72.9±0.6	81.9±2.3	80.8±0.5	**96.0±0.1**