Proceedings Track

Theoretical Insights into Line Graph Transformation on Graph Learning

Editors: List of editors' names

Abstract

Line graph transformation has been widely studied in graph theory, where each node in a line graph corresponds to an edge in the original graph. This has inspired a series of graph neural networks (GNNs) applied on transformed line graphs, which has proven effective in various graph representation learning tasks. However, there is limited theoretical study on how line graph transformation affects expressivity of GNN models. In this study, we focus on two types of graphs known to be challenging to the Weisfeiler-Leman (WL) tests: Cai-Fürer-Immerman (CFI) graphs and strongly regular graphs, and show that applying line graph transformation helps exclude these challenging graph properties, thus potentially assist WL tests in distinguishing these graphs. We empirically validate our findings by conducting a series of experiments that compare the accuracy and efficiency of graph isomorphism tests and GNNs on both line-transformed and original graphs across these graph structure types.

Keywords: Graph isomorphism testing, Strongly regular graphs, CFI graphs, Expressivity, Graph neural networks.

1. Introduction

Graph Neural Networks (GNNs) are prominent models widely studied in the domain of graph representation learning, with various applications from social network analysis (Fan et al., 2019) and recommendation systems (Baskin et al., 2016) to drug discovery (Muzio et al., 2020). In particular, advancements in graph theory have played a significant role in the evolution of GNNs. The integration of graph measures, such as graph spectrum (Baldesi et al., 2018), centrality (Maurya et al., 2021), and modularity (Tsitsulin et al., 2023), has played a crucial role in shaping the design, analysis, and theoretical foundations of GNN research.

Line graph transformation is of particular interest due to its successful applications on GNNs, proving effective in several tasks such as link predictions (Cai et al., 2022; Liu et al., 2021), community detection (Chen et al., 2019a), and material discovery (Choudhary and DeCost, 2021; Ruff et al., 2024). As illustrated in Figure 1, the *line graph transformation* is a graph rewriting algorithm that converts a graph's edges into nodes. Transformed nodes are then connected if their corresponding edges in the original graph share a common node. This transformation effectively captures the relationships between edges in a graph, and is particularly useful for tasks where edge-centric information is critical.

While the alignment of the Weisfeiler-Leman (WL) test to Message Passing Neural Networks (MPNNs) (Morris et al., 2019; Xu et al., 2019) and higher-order GNNs (Maron et al., 2019a) on input graph has provided valuable insights into measuring the expressive power of these architectures, there has been limited exploration on the expressive power of these models applying on line graphs. Specifically, we are interested in answering the following questions:

© 2024 .

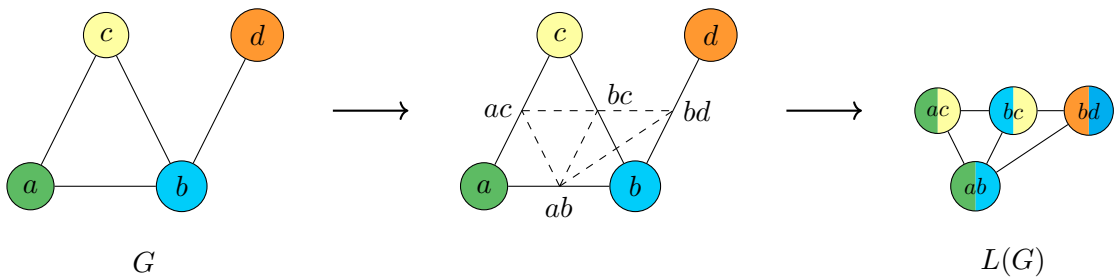


Figure 1: An example of converting a graph G to its line graph $L(G)$.

1. *What are the connections between learning permutation-invariant functions on the line graph and the original graph?*

2. *What classes of graphs can line graph transformation help distinguish?*

In this study, we aim to understand the effects of line graph transformations on the data from a graph theory perspective. We first observe that conducting a graph isomorphism test between a pair of connected graphs is equivalent to conducting it between their respective line graphs with the exception of the pair of graphs isomorphic to $K_{1,3}$ and C_3. Note that this holds for repeated applications of line graph transformation. These observations inspire us to demonstrate the equivalence of the graph isomorphism test and uniform function approximation on graphs after an arbitrary number of transformations $L^{(n)}(G)$ and the original graph G.

Furthermore, we shift our focus to Cai-Fürer-Immerman (CFI) graphs and strongly regular graphs, which are notably difficult for the WL test. We demonstrate that with several applications of line graph transformation, the resulting line graph will no longer exhibit the same property, potentially simplifying the process of differentiation through standard isomorphism tests or GNNs. We then perform a series of experiments to evaluate how line graph transformations affect the performance of standard isomorphism tests and GNNs in distinguishing difficult graph classes, focusing on the effectiveness of applying lower-order WL tests on the transformed line graphs for isomorphism checks.

Our contribution can be summarized as below:

1. We show that the universal approximation of permutation-invariant functions on connected graphs after arbitrary numbers of line graph transformation is equivalent to their original graph, assuming the root graph not isomorphic to $K_{1,3}$.

2. We study two challenging graph types for isomorphism tests: CFI graphs and strongly regular graphs, and show that they are excluded in line graphs with few exceptions.

3. We empirically validate our theory by conducting a series of graph isomorphism tests and GNNs and present line graph transformation's effect on distinguishing challenging graph pairs.

All proof details are presented in Appendix E.

Proceedings Track

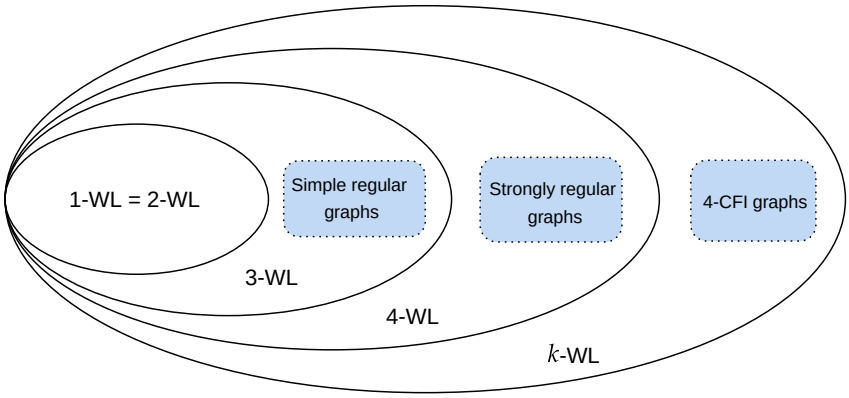


Figure 2: Relationships between WL tests and challenging graph types.

2. Related works

GNNs and WL test The comparison of the expressive power of GNNs with those of the WL tests is a well-studied topic in the literature. Xu et al. (2019) and Morris et al. (2019) demonstrated that Message-Passing Neural Networks (MPNNs) are capable of matching the expressiveness of the 1-WL test. Furthermore, Morris et al. (2019) extended this analysis by developing k-dimensional GNNs (k-GNNs) to align with the expressiveness of the k-WL tests, while inheriting their limitations on computational complexity. This led to theoretical analysis of several other existing classes of higher-order GNNs to study their expressive power. For instance, Frasca et al. (2022) has drawn connections from subgraph GNNs to Invariant Graph Networks (IGNs) (Maron et al., 2019b), showing that the expressive power of subgraph GNNs is inherently limited by 3-WL. Maron et al. (2019a) also derived architectures that closely match the expressive power of 3-WL.

Challenging graphs Another line of works (Babai and Kucera, 1979; Bouritsas et al., 2023; Cai et al., 1992) focuses on understanding what classes of graphs are inherently challenging to the hierarchy of WL tests, shown in Figure 2. One of the well-known classes is the Cai-Fürer-Immerman (CFI) graphs (Cai et al., 1992), designed specifically to be indistinguishable from the standard WL algorithm. CFI graphs are built by modifying the edges of a base graph with additional parity information encoded into gadgets, such that there exists a CFI graph that k-WL indistinguishable from any fixed k.

Another class of counter-examples to the WL test comes from the class of regular graphs, often characterized by their highly symmetrical structure and uniform node degrees, thus resulting in the same color refinement by 1-WL algorithms. In particular, strongly regular graphs include properties such as consistent numbers of common neighbors for adjacent and non-adjacent node pairs and are known to be 3-WL indistinguishable (Arvind et al., 2020), thus presenting challenges even with higher-order GNNs.

Line Graph Neural Networks While there does not yet exist a comprehensive theoretical justification for the use of line graphs with GNNs, there have been some studies showing promising experimental results with line graph neural networks. Cai et al. (2022)

show that line graphs effectively convert link prediction problems into node classification problems, which avoids the information loss caused by necessary graph pooling and leads to competitive performance in various link prediction benchmarks. In materials science, line graphs have been used for explicit modeling of chemical bond angles, which has proven useful in atomic property predictions (Choudhary and DeCost, 2021). Moreover, Chen et al. (2019a) demonstrated that line graph neural networks can construct the non-backtracking operator, enhancing the community detection capability in sparse graphs.

3. Line graphs and their properties

Notations We write $G = (V, E)$ to represent a finite, undirected, simple graph where V is the set of nodes and E is the set of edges. We denote $V(G)$ and $E(G)$ to specify the graph to which the set of nodes and the set of edges belong. Given a node $u \in V(G)$, $d_G(u)$ denotes the degree of u on graph G. We write $G \cong H$ if a graph G is isomorphic to H and $G \ncong H$ otherwise. We also adopt the common notations for special graphs, e.g., P_n, C_n, and K_n represent a path, a cycle, and a complete graph with n nodes, respectively. We also write $K_{1,n}$ to represent star graphs with one center node and n leaves.

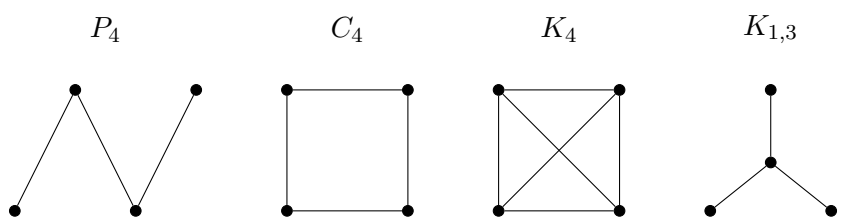


Figure 3: Examples of graph structures.

Line graph A line graph $\mathrm{L}(G)$ is derived from another graph G termed the *root graph*. Formally, given a root graph $G = (V, E)$, where V is the set of nodes and E is the set of edges, the *line graph*, denoted as $\mathrm{L}(G)$, is constructed such that each node in $\mathrm{L}(G)$ corresponds to an edge in E, and two nodes in $\mathrm{L}(G)$ are adjacent if their corresponding edges in G are adjacent with the same node in V. For notation convenience, we use $\mathrm{L}^{(n)}(G)$ to represent the resulting graph after n times line graph transformation (e.g. $\mathrm{L}^{(2)}(G) \coloneqq \mathrm{L}(\mathrm{L}(G))$). The definition of line graphs induces some corollary properties. For instance, let G be a simple undirected graph, it follows naturally that $|E(\mathrm{L}(G))| = |V(G)|$. In addition, Lemma 1 introduces a useful formula when analyzing the degree of new graphs.

Lemma 1 *Let $u, v \in V(G)$ such that they are adjacent by an edge $e \in E(G)$. The edge e's corresponding node representation $w_e \in V(L(G))$ follows $d_{L(G)}(w_e) = d_G(u) + d_G(v) - 2$.*

Some noticeable line graph transformations include $\mathrm{L}(P_n) = P_{n-1}$ (paths to shorter paths), $\mathrm{L}(C_n) = C_n$ (cycles to cycles), $\mathrm{L}(K_{1,n}) = K_n$ (stars to complete graphs). In particular, a *claw graph*, also known as the complete bipartite graph $K_{1,3}$, has a line graph of C_3. van Rooij and Wilf (1965) showed that for a connected graph G, the repeated line graph transformations $\mathrm{L}^{(n)}(G)$ has unbounded size as $n \to \infty$ if and only if G is not a path,

Proceedings Track

cycle or $K_{1,3}$. Additionally, we say a graph G is *triangle-containing* if G has a subgraph C_3, and is *triangle-free* if it does not.

Whitney's Isomorphism Theorem Formulated by Whitney (1932), a clear relation between original graphs and their line graphs is established. Whitney's theorem demonstrates a one-to-one mapping between the line graph and its root graph except for exceptions of C_3 and $K_{1,3}$, providing the theoretical foundation for learning with line graphs.

Theorem 2 (Whitney's Isomorphism Theorem (Whitney, 1932)) *Let G and H be finite, connected graphs. Then G is isomorphic to H if and only if their line graphs are isomorphic, with the exception of the case where G and H is a pair of C_3 and $K_{1,3}$, in which case their line graphs are both isomorphic to C_3.*

Beineke's Forbidden Induced Subgraphs In addition to the "almost injective" relationship between the root graph and its line graph transformation, there is an additional constraint on the constructed line graph. Beineke (1970) introduced the characteristic of line graphs that a graph G is a line graph of another graph if and only if G does not contain any of Beineke's forbidden subgraphs as an induced subgraph (See Appendix C for more details). As a direct consequence, we have the following key corollary, which will be instrumental in the next section.

Corollary 3 *Let G be a simple and undirected graph, then $L(G)$ does not contain $K_{1,3}$ as an induced subgraph.*

4. Theoretical Framework

We start by presenting the theoretical framework overview of this study, summarized as follow:

1. We demonstrate the equivalence between isomorphism testing on line graphs after any number of transformations and the approximation of permutation-invariant functions on the connected root graph with the exception of the $K_{1,3}$ graph. (Theorem 4, Corollary 5)

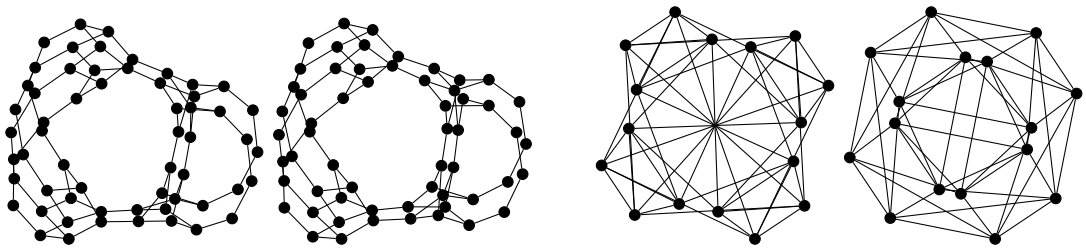
2. We show that higher orders of CFI construction, as proposed by Cai et al. (1992), are excluded from the set of line graphs. (Theorem 6)

3. We prove that with the exception of C_n when $3 \leq n \leq 5$, connected strongly regular graphs are transformed into non-strongly regular graphs by at most two line graph transformations. (Lemma 14, Theorem 7, Theorem 8)

4. We extend the theorem on equivalence and strongly regular graphs to disconnected graphs. (Corollary 9, Corollary 10)

4.1. Equivalence

In this section, we first focus on connected graphs that are not isomorphic to the claw graph $K_{1,3}$. Theorem 4 shows that approximating permutation-invariant functions on root graph G is *equivalent* to that on line graph $L(G)$ after a single transformation.



(a) CFI (b) Strongly Regular

Figure 4: Example of a pair of CFI graphs and a pair of strongly regular graphs.

Theorem 4 *Let $\mathcal{G}$ be a set of connected non-claw graphs and $\mathcal{C}$ be a collection of functions, such that $\forall G_1, G_2 \in \mathcal{G}$ such that $G_1 \ncong G_2$, $\exists h \in \mathcal{C}$, $h(L(G_1)) \neq h(L(G_2))$. Then, $\mathcal{C}$ can universally approximate any permutation-invariant function $f : \mathcal{G} \to \mathbb{R}$.*

The proof idea is to leverage Whitney's isomorphism theorem (Theorem 2) and establish a correspondence between a root graph G and the line graph $\mathrm{L}(G)$ (given that $G \ncong K_{1,3}$), and apply Theorem 11 (Appendix B) to link the isomorphism testing to universal function approximation. Excluding claw graphs, any pair of non-isomorphic root graphs would have non-isomorphic line graphs.

Furthermore, we can extend Theorem 4 to Corollary 5 by the claw-free property of line graphs (Corollary 3). As $\mathrm{L}(G)$ could not be $K_{1,3}$, it satisfies the assumption for Theorem 4 to extend the equivalence of universal function approximation between $\mathrm{L}^{(2)}(G)$ and $\mathrm{L}(G)$. Thus, by induction, the theorem can be extended to n-step line graph transformations for an arbitrary number of n.

Corollary 5 *Let $\mathcal{G}$ be a set of connected non-claw graphs and $\mathcal{C}$ be a collection of functions. If $\forall G_1, G_2 \in \mathcal{G}$, $G_1 \ncong G_2$, and $\forall n \in \mathbb{N}$, $\exists h \in \mathcal{C}$ such that $h(L^{(n)}(G_1)) \neq h(L^{(n)}(G_2))$ Then, $\mathcal{C}$ can universally approximate any permutation-invariant function $f : \mathcal{G} \to \mathbb{R}$.*

4.2. Implication on challenging graphs

The constraints on line graphs as described by Beineke (1970) offer a silver lining: despite the potential increase in graph size, the transformed graph may be structurally simpler. In this section, we introduce two challenging graphs in WL tests, namely CFI graphs and strongly regular graphs shown in Figure 4. We demonstrate how line graph transformation excludes the existence of such counter-examples for the isomorphism testing.

4.2.1. Cai-Fürer-Immerman Graphs

The Cai-Fürer-Immerman (CFI) graphs (Cai et al., 1992) are famously constructed to demonstrate the difficulty of the graph isomorphism problem, especially regarding WL tests, shown in Figure 4, (a). In fact, Cai et al. (1992) has shown that there exists a CFI construction that is indistinguishable under the k-WL test for any fixed k. The CFI graph

Proceedings Track

involves the construction of $X_k = (V_k, E_k)$ with $V_k = A_k \cup B_k \cup M_k$ where

$$A_k = \{a_i \mid 1 \leq i \leq k\}, B_k = \{b_i \mid 1 \leq i \leq k\}$$
$$M_k = \{m_S \mid S \subseteq \{1, \ldots, k\}, |S| \text{ is even}\}$$
$$E_k = \{(m_S, a_i) \mid i \in S\} \cup \{(m_S, b_i) \mid i \notin S\}.$$

However, we observe that applying line graph transformation on CFI graphs can provably remove this construction, since the CFI graphs X_k with $k \geq 3$ has to include the claw graph $K_{1,3}$ as an induced subgraph, thus ruling them out from being a line graph due to Corollary 3. Figure 5 presents an example of $K_{1,3}$ as a subgraph of X_3.

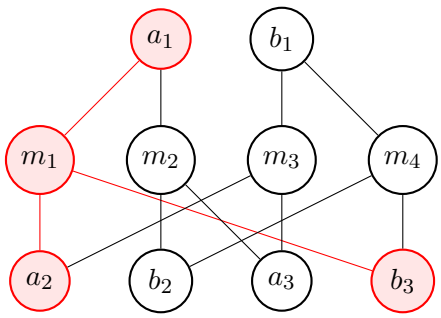


Figure 5: An CFI graph substructure X_3 and its induced subgraph $K_{1,3}$ in red.

Theorem 6 *Line graphs do not include CFI graphs constructed with X_k for $k \geq 3$.*

Theorem 6 demonstrates that after line graph transformation, complex CFI graphs would be non-existent. This prevents graph isomorphism tests and GNNs from dealing with these difficult instances, potentially assisting these models to distinguish challenging graph pairs and thus allowing GNNs to better learn on these graphs with the help of Theorem 4.

4.2.2. Strongly Regular Graphs

Another well-known example of the limitation of the WL test is its challenge to differentiate between strongly regular graphs (See Appendix D for details). To distinguish non-isomorphic strongly regular graphs, graph isomorphism tests or GNNs that are 4-WL expressive are required (Bouritsas et al., 2023). We present an example strongly regular graph pair that is indistinguishable by 3-WL in Figure 7, (b).

In this section, we demonstrate that the application of line graphs disrupts the strong regularity of these graphs, with the exception of three specific cases. We start with the following lemma to showcase the simple cases where the root graph contains C_3 as a subgraph (i.e. triangle-containing) and is not complete.

Theorem 7 *Let G be a connected, strongly regular graph which is not the complete graph K_n for $n \geq 2$, and G contains C_3 as a subgraph. Then $L(G)$ is not strongly regular.*

Furthermore, we show that with a finite number of exceptions, the application of most second-order line graph transformations effectively disrupts the strong regularity of graphs. The resulting graphs therefore may no longer require 4-WL test or equivalent GNN architectures for graph isomorphism testing.

Theorem 8 *Let G be a connected graph that is not the cycle graphs C_3, C_4, or C_5. Then, applying at most two line graph transformations to G yields a graph that is not strongly regular.*

4.2.3. Regular graphs

It is worth mentioning that the line graph of a regular graph remains regular (Ramane et al., 2005). Thus, there exists a pair of graphs, G_1 and G_2, such that are at least 2-WL indistinguishable between $\mathrm{L}^{(n)}(G_1)$ and $\mathrm{L}^{(n)}(G_2)$ for any $n \geq 0$.

4.3. Extension on disconnected graphs

Notice that all of the aforementioned theorem assume that the root graph are connected. In fact, we can show that given a disconnected graph where each of its connected components not isomorphic to $K_{1,3}$ or short paths, Corollary 5 still holds.

Corollary 9 *Let $\mathcal{G}$ be a set of graphs such that $\forall G \in \mathcal{G}$, for a fixed n, G does not have a component isomorphic to $K_{1,3}$ or P_k, $k \leq n$ and Let $\mathcal{C}$ be a collection of functions. If $\forall G_1, G_2 \in \mathcal{G}$ such that $G_1 \ncong G_2$, $\exists h \in \mathcal{C}$ such that $h(L^{(n)}(G_1)) \neq h(L^{(n)}(G_2))$. Then, $\mathcal{C}$ can universally approximate any permutation-invariant function $f : \mathcal{G} \to \mathbb{R}$.*

Similarly, assuming a disconnected graph where each of its connected components is not isomorphic to C_3, Corollary 8 holds for strongly regular graphs.

Corollary 10 *Let G be a graph that is not the cycle graphs C_3, C_4, C_5, or a disjoint group of C_3. Then, applying at most two line graph transformations to G yields a graph that is not strongly regular.*

5. Empirical Evidence

We validate our theoretical findings through graph classification experiments, designed to answer the following question: *What are the effects of line graph transformation on regular graphs and CFI graphs?*

5.1. Experimental Setup

Dataset We utilize the BREC dataset (Wang and Zhang, 2024) to examine the effects of line graph transformation on both CFI graphs and regular graphs. The dataset consists of 100 pairs of CFI graphs and 120 pairs of regular graphs. Among the regular graphs, 50 pairs are simple regular graphs, and 70 pairs are strongly regular graphs. These graph pairs are pre-processed into two distinct groups: with and without line graph transformation.

Experimental design In our experimental setup, we conduct graph isomorphism tests on 100 pairs of CFI graphs and 140 pairs of regular graphs, both before and after applying the line graph transformation, using the 3-WL test. Additionally, we apply the 4-WL on the root graph pairs and compare its expressive power on root graphs with that of 3-WL on their corresponding line graphs. To further demostreate, we also evaluate a GNN model, Provably Powerful Graph Networks (PPGN) (Maron et al., 2019a), whose expressive power is bounded by 3-WL (See Appendix F for more details).

Table 1: The accuracies of distinguishing graph pairs in the BREC regular and CFI graphs.

		Simple Regular (50)		Strongly Regular (70)		CFI (100)	
Graph	Model	Number	Accuracy	Number	Accuracy	Number	Accuracy
G	3-WL	50	100%	0	0%	60	60%
	4-WL	50	100%	70	100%	80	80%
	PPGN	50	100%	0	0%	22	22%
$L(G)$	3-WL	50	100%	70	100%	60	60%
	PPGN	50	100%	70	100%	15	15%

5.2. Results

Simple and strongly regular graphs We present the experiment results in Table 1. First, we note that the 3-WL or more expressive algorithms successfully distinguish all pairs of simple regular graphs in the dataset. Regarding strongly regular graphs which are distinguishable by 4-WL but not 3-WL, we observe that 3-WL can successfully differentiate all pairs of strongly regular graphs after applying the line graph transformation, which was previously indistinguishable. This is consistent with our theoretical analysis, suggesting that the strong regularity of graphs may be disrupted after line graph transformation.

CFI graphs It is noteworthy that the number of pairs of CFI graphs successfully distinguished by the 3-WL algorithm does not increase after a single line graph transformation. Although our theoretical analysis indicates transformed graph does not belong to CFI graph, the resulting line graph still poses significant challenges for the WL algorithm.

WL and PPGN We provide additional experimental insights into the comparison between WL algorithms and a graph neural network instance PPGN. In the regular graph experiments, PPGN aligns its performance with 3-WL before and after the line graph transformation. However, when applied to CFI graphs, PPGN performs slightly worse on the line graphs compared to the root graphs. As noted in Wang and Zhang (2024), PPGN struggles to match the performance of 3-WL on CFI graphs due to their large radii. The line graph transformation further increases the size of the CFI graphs, magnifying this issue and leading to decreased performance in PPGN experiments on the line graphs.

6. Conclusions

In this study, we provide a theoretical analysis of applying GNNs on line graphs. With mild assumptions, we show that line graphs are equivalent to the original graphs for isomorphism testing and permutation-invariant function approximation. In particular, we focus on two challenging graph classes for the isomorphism test, namely CFI graphs and strongly regular graphs, and show that both classes can be excluded or reduced by the application of line graph transformation. Empirically, strongly regular graphs after line graphs are 3-WL distingushable, whereas the line graphs of CFI graphs remains challenging to the WL tests.

References

V. Arvind, Frank Fuhlbrück, Johannes Köbler, and Oleg Verbitsky. On weisfeiler-leman invariance: Subgraph counts and related graph properties. *Journal of Computer and System Sciences*, 2020.

Laszlo Babai and Ludik Kucera. Canonical labelling of graphs in linear average time. In *20th Annual Symposium on Foundations of Computer Science*, 1979.

Luca Baldesi, Carter T. Butts, and Athina Markopoulou. Spectral graph forge: Graph generation targeting modularity. In *IEEE Conference on Computer Communications*, 2018.

I. Baskin, D. Winkler, and I. Tetko. A renaissance of neural networks in drug discovery. *Expert Opinion on Drug Discovery*, 2016.

Lowell W Beineke. Characterizations of derived graphs. *Journal of Combinatorial theory*, 1970.

Norman Biggs. *Algebraic graph theory*. Cambridge university press, 1993.

Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M. Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 1992.

Lei Cai, Jundong Li, Jie Wang, and Shuiwang Ji. Line graph neural networks for link prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Peter J Cameron. 6-transitive graphs. *Journal of Combinatorial Theory*, 1980.

Zhengdao Chen, Lisha Li, and Joan Bruna. Supervised community detection with line graph neural networks. In *7th International Conference on Learning Representations*, 2019a.

Zhengdao Chen, Soledad Villar, Lei Chen, and Joan Bruna. On the equivalence between graph isomorphism testing and function approximation with gnns. In *Advances in Neural Information Processing Systems*, 2019b.

Kamal Choudhary and Brian DeCost. Atomistic line graph neural network for improved materials property predictions. *npj Computational Materials*, 2021.

B. L. Douglas. The weisfeiler-lehman method and graph isomorphism testing, 2011.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, 2019.

Jiarui Feng, Lecheng Kong, Hao Liu, Dacheng Tao, Fuhai Li, Muhan Zhang, and Yixin Chen. Extending the design space of graph neural networks by rethinking folklore weisfeiler-lehman. In *Advances in Neural Information Processing Systems*, 2023.

Fabrizio Frasca, Beatrice Bevilacqua, Michael M. Bronstein, and Haggai Maron. Understanding and extending subgraph GNNs by rethinking their symmetries. In *Advances in Neural Information Processing Systems*, 2022.

F. Harary. *Graph Theory.* Addison-Wesley Publishing Company, 1969.

Shuwen Liu, Bernardo Grau, Ian Horrocks, and Egor Kostylev. Indigo: Gnn-based inductive knowledge graph completion using pair-wise encoding. In *Advances in Neural Information Processing Systems*, 2021.

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

Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. In *International Conference of Learning Representations*, 2019b.

Sunil Kumar Maurya, Xin Liu, and Tsuyoshi Murata. Graph neural networks for fast node ranking approximation. *ACM Transactions on Knowledge Discovery from Data*, 2021.

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*, 2019.

Giulia Muzio, Leslie O'Bray, and K. Borgwardt. Biological network analysis with deep learning. *Briefings in Bioinformatics*, 2020.

H.S. Ramane, H.B. Walikar, S.B. Rao, B.D. Acharya, P.R. Hampiholi, S.R. Jog, and I. Gutman. Spectra and energies of iterated line graphs of regular graphs. *Applied Mathematics Letters*, 2005.

Robin Ruff, Patrick Reiser, Jan Stühmer, and Pascal Friederich. Connectivity optimized nested line graph networks for crystal structures. *Digital Discovery*, 2024.

Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. Graph clustering with graph neural networks. *Journal of Machine Learning Research*, 2023.

Arnoud Caspar Maria van Rooij and Herbert S Wilf. The interchange graph of a finite graph. *Acta Mathematica Academiae Scientiarum Hungarica*, 1965.

Yanbo Wang and Muhan Zhang. An empirical study of realized GNN expressiveness. In *Forty-first International Conference on Machine Learning*, 2024.

Hassler Whitney. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, 1932.

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

Appendix A. Weisfeiler-Leman test

The *Weisfeiler-Leman (WL) test* is a graph isomorphism test widely applied in the field of graph theory and graph-based learning. In particular, the *k-dimensional Weisfeiler-Leman algorithm (k-WL)* operates by iteratively assigning colors to k-tuples of nodes in a graph. Formally, let $G = (V, E)$ be a graph, where V is the set of nodes and E the set of edges. For a fixed dimension k, the k-WL algorithm starts by assigning an initial colouring $\chi_k^{(0)}$ to every k-tuple $\mathbf{u} = (u_1, \ldots, u_k)$ of nodes. The coloring is defined based on the isomorphism type of the subgraph induced by the nodes in the tuple $\mathbf{u}$. At each iteration r, the algorithm refines the coloring as follows:

$$\chi_k^{(r+1)}(\mathbf{u}) = \left(\chi_k^{(r)}(\mathbf{u}), \left\{\chi_k^{(r)}(\mathbf{u}[w/i]) \mid w \in V, i \in [k]\right\}\right),$$

where $\mathbf{u}[w/i]$ denotes the tuple obtained by replacing the i-th node in $\mathbf{u}$ with node w. The refinement process terminates when further iterations no longer produce a different coloring.

Appendix B. Universal approximation over permutation-invariant functions

The link between the universal approximation capabilities of GNNs with respect to graphs and the testing of graph isomorphism was demonstrated by Chen et al. (2019b). In their work, it was shown that universal approximation of permutation-invariant functions on graphs are equivalent to graph isomorphism testing. This equivalence between universal approximation and graph isomorphism testing stated as Theorem 11 serves as the foundation for evaluating the expressiveness of different GNN architectures based on the expressivity of k-WL tests.

Theorem 11 (Chen et al. (2019b)) *A function class is capable of universally approximating permutation-invariant functions on graphs with finite node attributes if and only if it can discriminate non-isomorphic graphs.*

Appendix C. Beineke's forbidden induced subgraphs

Beineke (1970) introduced the following 9 graphs, shown in Figure 6 to be the forbidden induced subgraphs that characterize line graphs. The theorem states that a graph H is a line graph of a root graph G root if and only if H does not contain any of the nine Beineke's forbidden graphs as an induced subgraph.

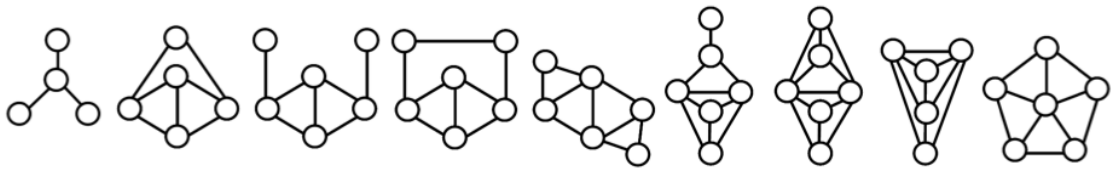


Figure 6: Nine Beineke's forbidden induced subgraphs.

Proceedings Track

Appendix D. Details on regular graphs

D.1. Regular graphs

A regular graph is a graph where each node has the same number of adjacent nodes, called the degree of the graph. In other words, a graph is k-regular if every node has exactly k neighbors. Examples of regular graphs include cycles and complete graphs.

D.2. Strongly regular graphs

A regular graph of v nodes is defined as strongly regular if there are positive integers k, λ, and μ satisfying

1. every node is connected to k other nodes,
2. each pair of connected nodes shares λ mutual neighbours, and
3. every pair of nodes that are not directly connected shares μ mutual neighbours.

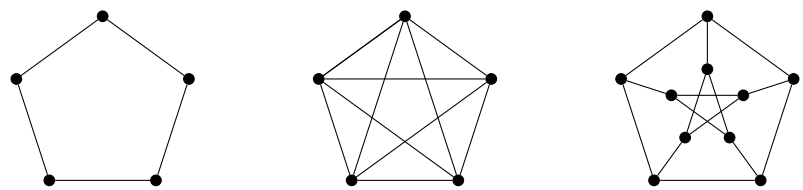


Figure 7: Examples of strongly regular graphs, displayed from left to right: the cycle graph C_5 ($\mathsf{srg}(5,2,0,1)$), the complete graph K_5 ($\mathsf{srg}(5,4,3,0)$), the Petersen graph ($\mathsf{srg}(10,3,0,1)$)

Such a graph can also be denoted as $\mathsf{srg}(v,k,\lambda,\mu)$ and the four parameters must obey the following relation (Biggs, 1993):

$$(v-k-1)\mu = k(k-\lambda-1). \tag{1}$$

D.3. Regular graphs and WL test

The WL distinguishability on regular graphs has been studied. This sections presents the relationship between the regular/strongly regular graphs and WL tests.

Theorem 12 *Regular graphs with the same of number nodes and same degree are not 1-WL distinguishable.*

Proof Initially, in the 1-WL algorithm, all nodes are assigned the same color. In each iteration, nodes update their color based on the same multiset of neighboring colors, as all nodes receive identical information. Consequently, the coloring remains uniform across all nodes in every iteration, preventing the algorithm from distinguishing between them. ■

Theorem 13 *(Bouritsas et al. (2023)) Strongly regular graphs with the same four parameters are not 3-WL distinguishable.*

D.4. Generalization of Regular Graphs

Cameron (1980) introduces the extension of regular graphs, namely *k-transitive* graphs or *k-isoregular* graphs, as graphs where the number of common neighbors for any k-tuple of a given isomorphism type remains constant. In this context, we write *simple regular graphs* as 1-isoregular graphs, and *strongly regular graphs* are 2-isoregular. It is known that k-isoregular graphs are indistinguishable by the $(k+1)$-dimensional Weisfeiler–Leman test (Douglas, 2011).

Appendix E. Proofs

Lemma 1 *Let $u, v \in V(G)$ such that they are adjacent by an edge $e \in E(G)$. The edge e's corresponding node representation $w_e \in V(L(G))$ follows $d_{L(G)}(w_e) = d_G(u) + d_G(v) - 2$.*

Proof In the root graph G, u is adjacent to $d_G(u)$ edges, and v is adjacent to $d_G(v)$ edges. Among all the edges that u, v are adjacent to, e is connected to all other edges by u or v except for e itself. Thus, in the line graph $L(G)$, w_e is adjacent to $d_G(v) - 1 + d_G(u) - 1 = d_G(v) + d_G(u) - 2$ other nodes. ■

Corollary 3 *Let G be a simple and undirected graph, $L(G)$ does not contain $K_{1,3}$ as an induced subgraph.*

Proof This corollary is a direct result of Beineke's forbidden subgraphs. $K_{1,3}$ is one of the Beineke's forbidden subgraph, so line graphs does not contain $K_{1,3}$ as an induced subgraph. ■

Theorem 4 *Let $\mathcal{G}$ be a set of connected non-claw graphs and $\mathcal{C}$ be a collection of functions, such that $\forall G_1, G_2 \in \mathcal{G}$ such that $G_1 \not\cong G_2$, $\exists h \in \mathcal{C}$, $h(L(G_1)) \neq h(L(G_2))$. Then, $\mathcal{C}$ can universally approximate any permutation-invariant function $f : \mathcal{G} \to \mathbb{R}$.*

Proof Let $f' : \mathrm{L}(\mathcal{G}) \to \mathbb{R}$ be any permutation-invariant functions on $\mathrm{L}(\mathcal{G})$. By Theorem 11, since $\exists h \in \mathcal{C}$ can distinguish non-isomorphic graphs in $\mathrm{L}(\mathcal{G})$, $\mathcal{C}$ can universally approximate any permutation-invariant functions $f' : \mathrm{L}(\mathcal{G}) \to \mathbb{R}$. Since G_1, G_2 are connected and not a claw graph, by Theorem 2, there exists a function g such that $g \circ f' = f$ where g injectively maps f' to f. ■

Corollary 5 *Let $\mathcal{G}$ be a set of connected non-claw graphs and $\mathcal{C}$ be a collection of functions. If $\forall G_1, G_2 \in \mathcal{G}$, $G_1 \not\cong G_2$, and $\forall n \in \mathbb{N}$, $\exists h \in \mathcal{C}$ such that $h(L^{(n)}(G_1)) \neq h(L^{(n)}(G_2))$ Then, $\mathcal{C}$ can universally approximate any permutation-invariant function $f : \mathcal{G} \to \mathbb{R}$.*

Proof If $n = 1$, the proof is the same as Theorem 4.

For $n \geq 2$, we can safely assume $\mathrm{L}^{(n-1)}(G)$ is not isomorphic to $K_{1,3}$ by Corollary 3. By Theorem 11, since $\exists h \in \mathcal{C}$ can distinguish non-isomorphic graphs in $\mathrm{L}(\mathcal{G})$, $\mathcal{C}$ can universally approximate any permutation-invariant functions $f_n : \mathrm{L}^{(n)}(\mathcal{G}) \to \mathbb{R}$. Since $\mathrm{L}^{(n-1)}(G)$ is not

a claw graph, by Theorem 2, there exists a function g_n such that $g_n \circ f_n = f_{n-1}$ where g_n injectively maps f_n to f_{n-1}. By inductively applying Theorem 2 we can reach the base case. ■

Theorem 6 *Line graphs do not include CFI graphs constructed with X_k for $k \geq 3$.*

Proof It can be seen that for $k > 2$, by construction there exists no direct edge connecting a_i and b_i, where a single m_S connects to k nodes from $A_k \cup B_k$. Thus, there exists an induced subgraph that is isomorphic to $K_{1,3}$. By Corollary 3, CFI graphs with X_k, $k \geq 3$ are excluded by line graph transformation. ■

Lemma 14 *If a graph $G = \mathsf{srg}(v, k, \lambda, \mu)$ is connected and strongly regular, it holds that $k \leq 2$ if and only if G is one of K_2, C_3, C_4, C_5 and the graph with one node.*

Proof We show the proof by enumerating all the cases when $k \leq 2$. When $k = 0$, we have only the graph with only one node. When $k = 1$, for the graph to be connected it can only be the case when two nodes are connected by a single edge. When $k = 2$, regular graphs are cycles. Cycles with lengths 6 or more are not strongly regular because nodes that are a distance of 2 from each other share exactly one common neighbour, whereas nodes that are 3 or more distance apart do not share any common neighbours. This inconsistency in the number of common neighbours violates the definition of a strongly regular graph, where the number of common neighbours between nodes must be constant for both adjacent and non-adjacent pairs. ■

Theorem 7 *Let G be a connected, strongly regular graph which is not the complete graph K_n for $n \geq 2$, and G contains C_3 as a subgraph. Then $L(G)$ is not strongly regular.*

Proof Suppose G is a strongly regular graph denoted as $\mathrm{srg}(v, k, \lambda, \mu)$ (defined in Appendix D.2). Given that G is not a complete graph, it follows that $\mu \neq 0$, where μ denotes the number of mutual neighbours between every pair of non-adjacent nodes. Also, since G is triangle-containing and not K_3, by Lemma 14, we have $k \geq 3$. Suppose, for a contradiction, that $\mathrm{L}(G)$ is strongly regular, which can be denoted as $\mathrm{srg}(v^*, k^*, \lambda^*, \mu^*)$.

By the properties of line graphs, the nodes of $\mathrm{L}(G)$ correspond to the edges of G, hence $v^* = |E(G)|$. Based on the property of regular graphs, we have $|E(G)| = \frac{vk}{2} = v^*$. In $\mathrm{L}(G)$, each edge is adjacent to every other edge that shares a node in G, thus (also by Equation 1) we have $k^* = 2(k-1)$.

Given that G contains triangles, let two edges e_0 and e_1 in $E(G)$ be part of a triangle shown in Figure 8. By the definition of line graphs, there exists a node $u \in V(G)$ that is adjacent to e_0 and e_1. Thus, the third side e_2 of the triangle as well as all other $k-2$ edges connected with u are common neighbours in $\mathrm{L}(G)$. Therefore, for $\mathrm{L}(G)$ to be strongly regular, λ^* must be $k-1$. Given the $k \geq 3$, there exists an edge e_3 that does not share a node with e_2 on G, which corresponds to non-neighbouring nodes in $\mathrm{L}(G)$. This yields μ^* values ranging from 2 to 4 based on all possible configurations shown in Figure 8.

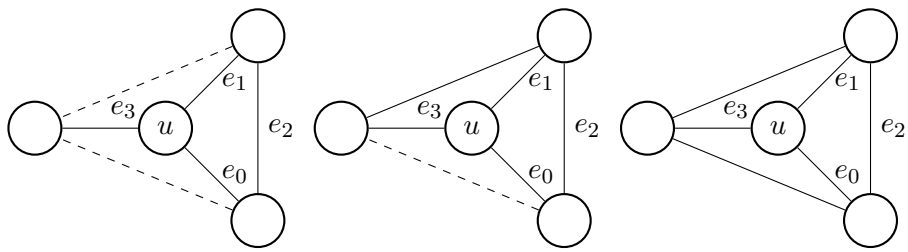


Figure 8: Three different cases in the triangle region in G where the dashed line represents no edge exists. In the first case, there is no edge connecting the other end node of e_3 to either end node of e_2. Only e_0 and e_1 are the common neighbours in $\mathrm{L}(G)$ for the pair e_2 and e_3.. The second and third cases represent if one or both edges exist, where the edges would also be neighbouring nodes in $\mathrm{L}(G)$.

We use the following property in Equation 1 where $v^* = \frac{vk}{2}$, $k^* = 2(k-1)$, $\lambda^* = k-1$, and $\mu^* = 2, 3, 4$. Evaluating this equation under $\mu^* = 2$ or $\mu^* = 3$ leads to the only positive integer solutions $v = 2, k = 1$ and $v = 3, k = 2$. Both solutions are invalid as we have $k \geq 3$. For $\mu^* = 4$, the solutions are $v = k + 1$, which contradicts the assumption that G is not complete. This shows that $\mathrm{L}(G)$ cannot be strongly regular.

■

Theorem 8 *Let G be a connected graph that is not the cycle graphs C_3, C_4, or C_5. Then, applying at most two line graph transformations to G yields a graph that is not strongly regular.*

Proof The proof is structured based on the characteristics of the graph G and proceeds in five cases:

Case 1: If G is not strongly regular, then by definition, no line graph transformation is required.

Case 2: If G is the graph with one node or K_2, we can easily see that $\mathrm{L}(G)$ and $\mathrm{L}(\mathrm{L}(G))$ respectively are not strongly regular.

Case 3: If G is a strongly regular graph that contains triangles and is not complete, we can apply Theorem 7 to show that $\mathrm{L}(G)$ is not strongly regular.

Case 4: If a strongly regular graph G is triangle-free with $k \geq 3$, each neighbourhood of nodes in G is star-like. Thus, for any two edges e and e' in G that are connected through a node u, they share exactly $k-2$ common neighbours (the other edges connected to u). Assuming $\mathrm{L}(G)$ is strongly regular, with parameters $v^*, k^*, \lambda^*, \mu^*$, it follows that $\lambda^* = k-2$, which is nonzero. This implies $\mathrm{L}(G)$ is triangle-containing. Given that only star graphs have line graphs that are complete, $\mathrm{L}(G)$ cannot be a complete graph. By Theorem 7, we need only one more line graph transformation to have $\mathrm{L}(\mathrm{L}(G))$ not strongly regular.

Case 5: By Lemma 14, we only have complete graphs with $v \geq 4$ left. If G is a complete graph with $v \geq 4$, it follows that the line graph of G, $\mathrm{L}(G)$, can be represented as a strongly regular graph with parameters $\mathsf{srg}\left(\binom{v}{2}, 2(v-2), v-2, 4\right)$ (Harary, 1969). In this scenario,

$\mathrm{L}(G)$ contains a triangle implied by the nonzero parameter $v-2$. Applying Theorem 7, we can show that the line graph of $\mathrm{L}(G)$, $\mathrm{L}(\mathrm{L}(G))$, is not strongly regular. ■

Corollary 9 *Let $\mathcal{G}$ be a set of graphs such that $\forall G \in \mathcal{G}$, for a fixed n, G does not have a component isomorphic to $K_{1,3}$ or P_k, $k \leq n$ and Let $\mathcal{C}$ be a collection of functions. If $\forall G_1, G_2 \in \mathcal{G}$ such that $G_1 \ncong G_2$, $\exists h \in \mathcal{C}$ such that $h(L^{(n)}(G_1)) \neq h(L^{(n)}(G_2))$. Then, $\mathcal{C}$ can universally approximate any permutation-invariant function $f : \mathcal{G} \to \mathbb{R}$.*

Proof We need to demonstrate that Corollary 5 applies to disconnected graphs as well. It suffices to show that Theorem 2 holds for G. For disconnected graphs, the line graph transformation is applied component-wise. Since G contains no component isomorphic to $K_{1,3}$, each component of G is uniquely mapped to a line graph component corresponding to the root component by Theorem 2. Also, under the assumption of no path with a length shorter than n is included in G as a connected component, every component would not vanish after repeated line graphs transformation (conversely, we see paths of length k or shorter would be turned to a graph with no nodes after k line graph transformations). Consequently, the line graph of the entire graph G is also unique to its root graph. The rest of the proof proceeds as in Corollary 5. ■

Corollary 10 *Let G be a strongly regular graph that is not the cycle graphs C_3, C_4, C_5, or a disjoint group of C_3. Then, applying at most two line graph transformations to G yields a graph that is not strongly regular.*

Proof For a graph G to be disconnected and strongly regular, it could only be the case where the graph is a set of disjoint K_n (Biggs, 1993). When $n \leq 2$, we can see that at most two line graphs would reduce G to an empty graph. When $n \geq 3$, each component would be reduced to $\mathsf{srg}\left(\binom{v}{2}, 2(v-2), v-2, 4\right)$. The overall graphs would not be strongly regular because one node in a component does not share any common neighbors with nodes in other components. When G is a set of C_3 (i.e. K_3), we have $\mathrm{L}^{(n)}(G) \cong G$. ■

Appendix F. Experimental details

F.1. Code availability

We open-sources our code to replicate the experiments on GitHub. The details about the experimental setup and training parameters can be found in the GitHub repository.

F.2. Dataset

Our experiments were conducted on the BREC dataset (Wang and Zhang, 2024) with the sections **Regular Graphs**, and **CFI Graphs**. A summary of each category is provided below.

We selected 120 pairs of regular graphs, which can be further divided into simple regular graphs, strongly regular graphs, 4-vertex condition graphs:

- **Simple regular graphs**: We selected 50 pairs of simple regular graphs, each with 6 to 10 nodes, by randomly choosing pairs with identical parameters.
- **Strongly regular graphs**: This subset includes 50 pairs of strongly regular graphs with node counts ranging from 16 to 35. The graphs were sourced from databases such as SR Graphs and BDM Graphs.
- **4-vertex condition graphs**: A set of 20 pairs of 4-vertex condition graphs was selected from the 4-vertex Condition Graph Database with parameters $\mathsf{srg}(63, 30, 13, 15)$.

Note that 4-vertex condition graphs are a specific subtype of strongly regular graphs, and as such, we classify them within the same category as other strongly regular graphs in our results.

We selected the 100 pairs of graphs in the BREC dataset. The backbone graphs ranged from 3 to 7 nodes. The dataset contains:

- 60 pairs of 1-WL-indistinguishable CFI graphs,
- 20 pairs of 3-WL-indistinguishable CFI graphs, and
- 20 pairs of 4-WL-indistinguishable CFI graphs.

F.3. Provably Powerful Neural Networks

To address the limitations of 1-WL expressiveness, recent research has focused on constructing GNNs that match or surpass the expressive power of higher-order WL tests. In this section, we discuss the provably powerful neural network (PPGN) proposed by Maron et al. (2019a) with 3-WL expressiveness can distinguish between non-isomorphic graphs that are indistinguishable by 1-WL.

The construction of a 3-WL expressive GNN involves three main components:

1. **Input representation**: The graph $G = (V, E, d)$, where V is the set of nodes, E is the set of edges, and d represents node features (or colors), is represented as a tensor $B \in \mathbb{R}^{n^2 \times (e+1)}$, where n is the number of nodes, e is the number of features, and the last channel of B encodes the graph adjacency matrix.
2. **Network layers**: The key operations of the GNN are organized into *blocks*, each consisting of:
   - A *Multilayer Perceptron (MLP)* applied independently to each feature of the input tensor. This is denoted as m_i where $i \in \{1, 2, 3\}$. Each MLP transforms the input features.
   - A *matrix multiplication* between the transformed feature tensors. Let $X \in \mathbb{R}^{n \times n \times a}$ be the input tensor to the block, where a is the feature dimension. The matrix multiplication is performed between the transformed tensors:

     $$W_{:,:,j} = m_1(X)_{:,:,j} \cdot m_2(X)_{:,:,j} \quad \forall j \in \{1, \ldots, b\}.$$

     The output tensor of the block consists of the concatenated MLP transformation $m_3(X)$ and the result of matrix multiplication W.

Proceedings Track

3. **Invariant and equivariant layers**: The model employs invariant and equivariant layers to ensure that its operations respect the permutation symmetry of the graph. The matrix multiplication operation is equivariant to permutations, ensuring that the model's output is invariant to node reordering. This structure ensures that the GNN can distinguish between non-isomorphic graphs that 1-WL cannot.

The GNN described above is provably as expressive as the 3-WL test. Formally, the following result holds:

- For any two graphs G and G' that can be distinguished by the 3-WL graph isomorphism test, there exists a 3-WL expressive PPGN F such that $F(G) \neq F(G')$.
- Conversely, if G and G' are isomorphic, then $F(G) = F(G')$ for any 3-WL expressive PPGN F.

F.4. Complexity analysis

In this section, we present the space and time complexity analysis associated with applying the line graph transformation in the context of k-WL tests. The k-WL test has a space complexity of $O(n^k)$ and a time complexity of $O(n^{k+1})$ (Feng et al., 2023), where n is the number of nodes in a graph G, i.e., $n = |V(G)|$. After performing the line graph transformation, the number of nodes in the transformed graph corresponds to the number of edges in the original graph. Consequently, for structures like paths and cycles, the space complexity remains $O(n^k)$ and the time complexity $O(n^{k+1})$.

For a d-regular graph, the number of nodes in the line graph is $\frac{dn}{2}$. Therefore, the space complexity becomes $O(d^k n^k)$, and the time complexity is $O(d^{k+1} n^{k+1})$. In the worst-case scenario, when the graph is dense—such as in a complete graph with $\frac{n(n-1)}{2}$ edges—the space complexity increases to $O(n^{2k})$ and the time complexity to $O(n^{2k+2})$.

The empirical comparison of time consumption on regular graphs is presented in Figure 9. A consistent increase in time consumption by orders of magnitude is observed. We present the subset of 4-vertex condition strongly regular graphs separately, as their significantly larger size distinguishes them from other strongly regular graphs in the BREC dataset. In the PPGN comparison, the line graph generally requires more time compared to the root graph. However, the results vary depending on the specific hyperparameter and epoch settings of the PGNN.

F.5. Experimental setup and parameters

For training the PPGN, we employed the Adam optimizer with a learning rate of 0.0001 and a weight decay of 0.0001. The loss function used was `CosineEmbeddingLoss`. The model architecture consisted of 5 layers with an inner embedding dimension of 32. For the root graph, a batch size of 32 was used, whereas for the line graph, a smaller batch size of 4 was applied due to memory limitations.

F.6. Compute resources

For 3-WL and 4-WL tests, we used 4-core CPU clusters to run the analysis. For the PPGN experiment, we used an additional 40 GB A100 GPU to accelerate the training.

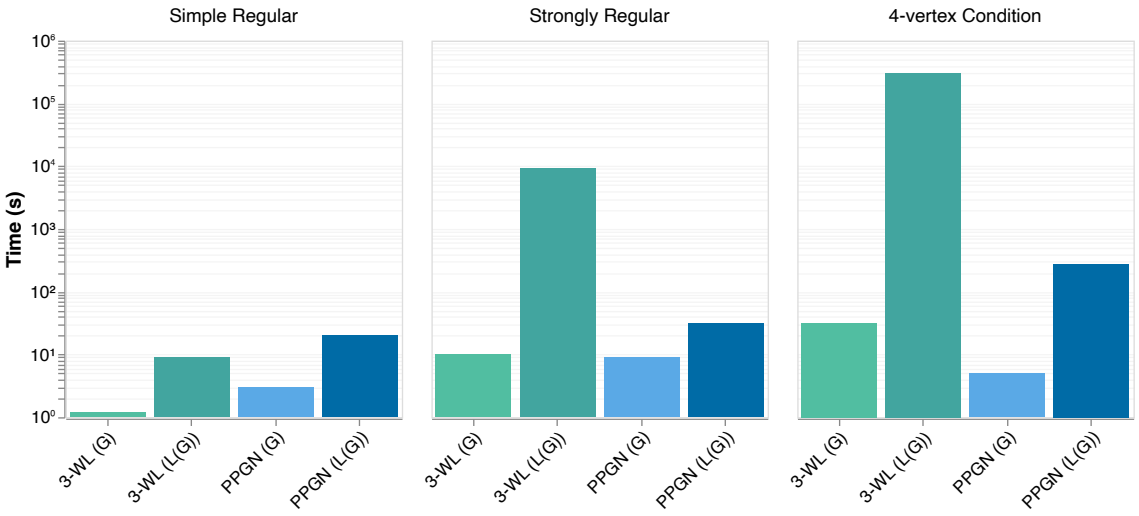


Figure 9: Time consumption on simple regular, strongly regular, and 4-vertex condition strongly regular subsets.