A Topological characterisation of Weisfeiler-Leman equivalence classes

Jacob Bamberger
Signal Processing Laboratory 2
EPFL,
Lausanne, Switzerland
`jacob.bamberger@epfl.ch`

Abstract

Graph Neural Networks (GNNs) are learning models aimed at processing graphs and signals on graphs. The most popular and successful GNNs are based on message passing schemes. Such schemes inherently have limited expressive power when it comes to distinguishing two non-isomorphic graphs. In this article, we rely on the theory of covering spaces to fully characterize the classes of graphs that GNNs cannot distinguish. We then generate arbitrarily many non-isomorphic graphs that cannot be distinguished by GNNs, leading to the GraphCovers dataset. We also show that the number of indistinguishable graphs in our dataset grows super-exponentially with the number of nodes. Finally, we test the GraphCovers dataset on several GNN architectures, showing that none of them can distinguish any two graphs it contains.

1 Introduction

Many modern Graph Neural Networks (GNNs) use a message passing scheme (Gilmer et al., 2017). Recent studies compared such message passing GNNs to the Weisfeiler-Leman (WL) graph isomorphism test (Weisfeiler & Leman, 1968), and used this analogy to show that such GNNs have limited expressive power (Morris et al., 2019; Xu et al., 2019). The challenge of finding architectures that overcome these limits is a very active research question, and has inspired much work, e.g. Morris et al. (2019); Garg et al. (2020); Papp et al. (2021); Bevilacqua et al. (2022).

In this paper, we build upon the WL analogy to fully characterize graphs that GNNs cannot distinguish. We do so by extending Leighton's Theorem (Leighton, 1982) to give a topological description of graphs that cannot be distinguished by the WL test. The description uses covering maps, a notion of local isomorphism between spaces. Our proofs being mainly constructive, we use them to generate a family of graphs that are non-isomorphic but indistinguishable by WL, leading to the GraphCovers dataset. We then show that the number of generated isomorphism classes scales super-exponentially with the desired number of nodes. Finally, we benchmark the GraphCovers dataset on the graph classification task with several common GNN architectures, showing that none of them can distinguish any two graphs it contains.

Before introducing the terms and concepts, we state the main "battle horse" of this paper, which is a rephrasing of Leighton's Theorem. See Figure 1 for an illustration.

Theorem 1.1. *(Leighton, 1982) Two connected graphs G and G' are indistinguishable by the WL test if and only if they have the same order and have isomorphic universal covers, or equivalently, if and only if they have the same order and have isomorphic finite covers.*

We give a self contained proof of a simpler version of Theorem 1.1, omitting the last statement. We namely prove that G and G' are indistinguishable by WL if and only if they have the same order and have isomorphic universal covers. The full theorem is not proven for two reasons: first, the partial result will be sufficiently strong for us to generate the GraphCovers dataset; second, we expect our proofs to generalize to simplicial and cellular message passing as defined in Bodnar et al. (2021a;b), whereas it is known that Leighton's Theorem does not generalize to arbitrary simplicial complexes, e.g. Wise (1996).
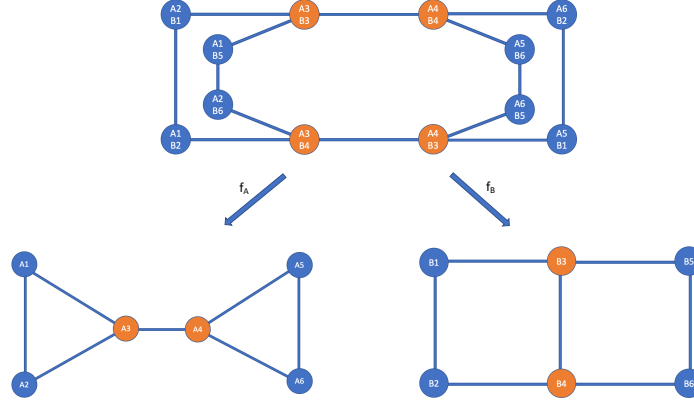
Figure 1: Two graphs G_A (left) and G_B (right) and their stable WL colourings which are indistinguishable, and a common covering space (top), with maps f_A and f_B defined by the vertex labels.

To fully characterize WL equivalence classes, we need to also consider disconnected graphs, since for example the WL test is fooled by the length 6 cycle and the disjoint union of two length 3 cycles. Hence we generalize Theorem 1.1 to get the following classification result for WL equivalence classes. The proof can be found in the Appendix B [1].

Theorem 1.2. *Two grphs G and G' of bounded degree are indistinguishable by the WL test if and only if they admit decompositions into disjoint unions of connected graphs:*

$$G = \bigoplus_{i \in \mathcal{I}} \bigoplus_{j \in \mathcal{J}_i} G_{ij} \text{ and } G' = \bigoplus_{i \in \mathcal{I}} \bigoplus_{j' \in \mathcal{J}_i'} G'_{ij'} \, ,$$

such that the following holds
- *for every $i \in \mathcal{I}$, $|\bigoplus_{j \in \mathcal{J}_i} G_{ij}| = |\bigoplus_{j' \in \mathcal{J}_i'} G'_{ij'}|$*
- *for every $i \in \mathcal{I}$, there is a graph H_i that covers both G_{ij} and $G'_{ij'}$ for all $j \in \mathcal{J}_i$ and $j' \in \mathcal{J}_i'$.*

Moreover, if $|\bigoplus_{j \in \mathcal{J}_i} G_{ij}| < \infty$, the cover H_i can be chosen to be finite.

1.1 RELATED WORK

Coverings: Covering maps for graphs are a special case of covering maps for topological spaces, see Chapter 1 in Hatcher (2000) for a classical introduction to coverings, and Chapter 1.A for the case of graphs. Covering graphs have been fruitful in many ways, both in applications and in pure mathematics: they were first applied to problems in multi-processor networks (Angluin, 1980), since then they have been at the core of the field of geometric and combinatorial group theory, e.g. Stallings (1983). They have also been studied from a purely graph theoretic perspective, e.g. Gross & Tucker (1977), from a probabilistic viewpoint Amit & Linial (2002), and from a spectral one Chung & Yau (1998). However, we are not aware of applications of coverings to GNNs.

GNN expressiveness: A successful research topic has been to understand how good GNNs are at distinguishing non-isomorphic graphs. To do so, GNNs have been compared to the WL test, an algorithm capable of distinguishing most, but not all, non-isomorphic graphs (Weisfeiler & Leman, 1968). Using the WL analogy, researchers have shown that GNNs have limited expressive power when it comes to distinguishing non-isomorphic graphs (Morris et al., 2019; Xu et al., 2019). These results have triggered a body of work that study the combinatorial structures GNNs cannot distinguish, e.g. Loukas (2020a;b); Chen et al. (2020); Garg et al. (2020). A line of work showed that GNN architectures gain in expressive power by encoding node ids either uniquely or randomly (Loukas, 2020a; Abboud et al., 2021; Sato et al., 2021), or through the use of port numbers - an

---

[1]https://arxiv.org/abs/2206.11876

idea adapted from the theory of distributed local algorithms (Sato et al., 2019). Another approach to gaining expressive power is to extend the WL analogy to the higher order k-WL tests (Immerman & Lander, 1990). This idea lead to several higher order architectures which are provably more expressive (Morris et al., 2019; Maron et al., 2019). Another approach is to encode substructures by treating graphs as a (possibly random) set of subgraphs (Papp et al., 2021; Bevilacqua et al., 2022), or by treating them as simplicial or cellular complexes (Bodnar et al., 2021a;b).

2 Proof of the main Theorem

2.1 Definitions

We start by defining graphs, (iso)morphisms and coverings between graphs, and (iso)morphisms between coverings.

Definition 2.1. An undirected graph G consists of a countable vertex set $\mathcal{V}(G)$ and an edge set $\mathcal{E}(G) \subseteq \mathcal{V}(G) \times \mathcal{V}(G)$ that is symmetric (i.e. $vw \in \mathcal{E}(G) \implies wv \in \mathcal{E}(G)$). When the underlying graph is clear from context, we refer to them as $\mathcal{V}$ and $\mathcal{E}$. The neighbors of a vertex v in a graph G is defined as $\mathcal{N}_G(v) := \{w \in \mathcal{V}(G) \mid vw \in \mathcal{E}(G)\}$. For each vertex v, we define it's neighborhood as the subgraph $G_v \subseteq G$ with vertices $\mathcal{V}(G_v) := \{v\} \bigcup \mathcal{N}_G(v)$ and edges being the edges uv and vu connecting v to a neighbor $u \in \mathcal{N}_v$. The order of G is $|\mathcal{V}(G)|$.

Definition 2.2. A morphism $\phi : H \to G$ between two graphs H, G is a map (denoted by the same symbol) $\phi : \mathcal{V}(H) \to \mathcal{V}(G)$ that preserves edges, i.e. $vw \in \mathcal{E}(H) \implies \phi(v)\phi(w) \in \mathcal{E}(G)$. It is an isomorphism if ϕ is bijective on vertices and if the induced map on edges is also bijective, i.e. $vw \in \mathcal{E}(H) \iff \phi(v)\phi(w) \in \mathcal{E}(G)$. It is a covering if it is surjective on vertices, and if ϕ restricts to an isomorphism on all neighbor graphs G_v. That is, if $\forall v \in \mathcal{V}(G)$, $\forall v' \in \phi^{-1}(v)$, and $\forall w' \in N_G(v')$, $\exists! v'w' \in \mathcal{E}(H)$ such that $\phi(v)\phi(w) = v'w'$. If ϕ is a covering map, we refer to the co-domain as the base graph.

Remark 2.3. Note that isomorphisms are a special case of coverings. See Figure 1 for examples of covering maps.

Definition 2.4. A morphism between covering spaces $\pi : H \to G$ and $\pi' : H' \to G$ of a same base graph G is a graph morphism $\phi : H \to H'$, such that $\pi' \circ \phi = \pi$, i.e. such that the following diagram commutes:

$$
\begin{array}{ccc}
H & \xrightarrow{\ \ \phi\ \ } & H' \\
& \searrow{\scriptstyle \pi} \quad \swarrow{\scriptstyle \pi'} & \\
& G &
\end{array}
$$

It is an isomorphism of covering space if ϕ is an isomorphism of graphs.

Remark 2.5. For any graph G considered, we assume that it has bounded degree (i.e. $\exists K \in \mathbb{N}$ such that $\forall v \in \mathcal{V}(G)$, $|\mathcal{N}(v)| < K$). Note that both finite graphs, and coverings of finite graphs satisfy this property. We refer to undirected graphs with this property simply as graphs.

We now introduce the notion of colorings and neighborhood colorings, which are essential to define the color refinement, itself used to define the WL test.

Definition 2.6. A coloring of a graph G is a mapping $c : \mathcal{V}(G) \to \mathbb{N}$, where each $n \in \mathbb{N}$ is thought of as a color. A colored neighborhood of $v \in \mathcal{V}(G)$ is a coloring of G_v. Given a coloring c of G, restricting it to G_v induces a neighborhood coloring $c|_{G_v}$ for every v. We let $\mathcal{G}$ be a set of representatives for the isomorphism classes of neighborhoods (that is, for each $v \in \mathcal{V}(G)$ there is a $K \in \mathcal{G}$ such that $G_v \cong K$). Define the set of neighborhood colourings $CN(G) = \{c : K \to \mathbb{N}, K \in \mathcal{G}\}$. For a graph G satisfying the finiteness assumption of Remark 2.5, we may take $\mathcal{G}$ to be finite, which ensures the existence of an injective map $HASH : CN(G) \to \mathbb{N}$.

Definition 2.7. The color refinement of a graph G is the coloring process described in Algorithm 1. It is therefore a sequence of colorings c_i of G. See Remark 2.8 for a comment on the classical approach using multisets, and Remark 2.9 for one on convergence of the algorithm. We abuse notation and pretend that it does not depend on $HASH$.

Algorithm 1 Color refinement

Input: Graph G
Fix c_0 a constant coloring of G, set $i = 0$
Fix an injective $HASH : CN(G) \to \mathbb{N}$
while c_i & c_{i-1} induce different partitions of $\mathcal{V}(G)$ **do**
    for $v \in \mathcal{V}(G)$ **do**

        set $c_i(v) = HASH(\mathcal{V}(G_v) \xrightarrow{c_{i-1}|_{G_v}} \mathbb{N})$
    end for
    i = i+1
end while
return c_i

Remark 2.8. The usual exposition to the WL test would replace our definition of $CN(G)$ by multisets, where at stage i of Algorithm 1 the multiset associated to $v \in \mathcal{V}(G)$ is $\{c_i(v), \{\{c_i(u), u \in \mathcal{N}_G(v)\}\}\}$, and the HASH function goes from the set of multisets, e.g. Morris et al. (2019).

Remark 2.9. For finite graphs, Algorithm 1 converges in $\mathcal{O}(|\mathcal{V}(G)|)$. We refer to this converged coloring, if it exists, as the stable coloring of a graph. For infinite graphs, the algorithm does not necessarily converge. However, for an infinite graph that covers a finite graph, the color refinement converges by Lemma 2.12, making Lemma 2.11 applicable to universal covers defined in Definition 2.16.

Definition 2.10. The WL test between G and G' consists of running the color refinement on the disjoint union $G \bigoplus G'$ until convergence, if both graphs have the same number of nodes of each color in the stable coloring, we say that G and G' are WL equivalent (they succeed WL) otherwise we say that they are not WL equivalent (they fail WL).

2.2 LEMMAS AND PROOFS

In this subsection, we present the main Lemmas and proofs needed for Theorem 1.1. We first recall that the WL test solves the isomorphism problem for trees, whose proof can be found in Appendix A [2].

Lemma 2.11. *Let T and T' be (possibly infinite) trees with a stable coloring, then T and T' are indistinguishable by WL if and only if they are isomorphic.*

We show that coverings keep the color refinement invariant:

Lemma 2.12. *The color refinement of a graph is fully determined by the color refinement of any graph it covers or any graph covering it. That is, let G, H be graphs, and $\pi : H \to G$ be a covering map. If $g \in G$ has a given color at step i, then any $h \in \pi^{-1}(g)$ has the same color at step i. Conversely, if $\pi(h)$ has a given color at step i, then $h \in H$ has the same color at step i.*

Proof. We prove the result inductively. At the first stage, both graphs are colored by one color so the statement holds. Suppose the statement holds at stage i, then by definition of a cover, the subgraph $H_h \subseteq H$ is mapped isomorphically to the subgraph $G_{\pi(h)} \subseteq G$. By assumption the ith coloring is fixed by π, so the neighborhood coloring is the same, and so is the updated colors. $\qquad\square$

We have seen that covering maps respect the color refinement of graphs. Hence a graph and its cover look alike from the WL perspective. But for finite graphs, the WL test always distinguishes two graphs with different number of nodes. The dichotomy between order and covers is in fact sufficient to fully characterise WL equivalence classes. To formalize the interplay between covers and order, we introduce the degree of covers, for which we need the following result:

Lemma 2.13. *Let H and G be connected graphs, and let $\pi : H \to G$ be a covering map. For any two v, $v' \in \mathcal{V}(G)$, the fibers of v and v' are of the same cardinality, i.e. $|\pi^{-1}(v)| = |\pi^{-1}(v')|$. In particular, $|\mathcal{V}(H)| = |\pi^{-1}(v)| \cdot |\mathcal{V}(G)|$.*

---

[2]https://arxiv.org/abs/2206.11876

Proof. We show that the size of fibers is locally constant which implies the result since the graphs are path connected. Let $v \in \mathcal{V}(G)$, assume that $d = |\pi^{-1}(v)|$, and let $u \in \mathcal{N}_G(v)$. Since π is a cover, each $\tilde{v} \in \pi^{-1}(v)$ has precisely one neighbor $\tilde{u} \in \mathcal{N}_H(\tilde{v})$ with $\pi(\tilde{u}) = u$, showing that $|\pi^{-1}(u)| \geq |\pi^{-1}(v)|$. By symmetry, swapping u and v gives $|\pi^{-1}(v)| \geq |\pi^{-1}(u)|$ and concludes the proof. □

Definition 2.14. Let H and G be connected graphs, and let $\pi : H \to G$ be a covering map. The degree $d_\pi \in \mathbb{N}$ (or number of sheets) of π is the size of any pre-image of a node in G (i.e. $d_\pi = |\pi^{-1}(v))|)$, which is well defined by Lemma 2.13.

Using the above definition, we can prove a useful statement about the WL test and coverings:

Corollary 2.15. *Let G and G' be two connected graphs of equal order. Let $\pi : H \to G$ and $\pi' : H' \to G'$ be two covers such that H and H' are isomorphic. Then G and G' are WL equivalent.*

Proof. Since H and H' are isomorphic, their stable coloring must be the same. By Lemma 2.12, the coloring of G and G' are therefore the same. Since G and G' have the same order, both covers must have the same degree. We can conclude that the number of nodes of a given color in G (resp G') is given by the number of nodes of that color in H (resp H') divided by d. □

At last, we introduce the universal cover of a graph G:

Definition 2.16. The universal cover of a connected graph G is a cover $\pi : T \to G$ where T is a connected tree. This map is unique up to cover isomorphism, and covers all other covers of G (Hatcher, 2000). See Figure 2 for an illustration.
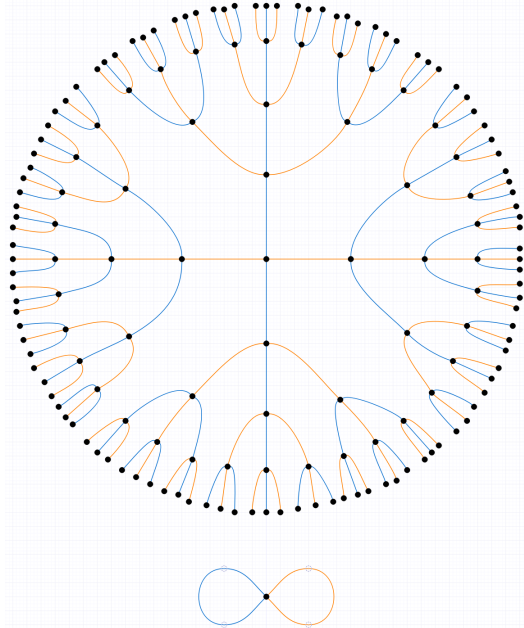

Figure 2: Finite subgraph of the infinite universal cover (top) of the rose with 2 petals (bottom). We apologies since the rose is not a graph as per Definition 2.1. To make it into a graph we would have to subdivide each edge twice, and do the same each edge of the universal cover. The finite subgraph of the universal cover is the 4 hop neighborhood of the central node.

Remark 2.17. If the Euler characteristic $\chi(G) := |\mathcal{V}| - |\mathcal{E}|$, of a finite connected graph G is ≤ 0 (i.e. G has a loop), then the universal cover is infinite, otherwise G is a tree.

Finally, we prove the main Lemma for the second direction of Theorem 1.1, and then prove the main theorem.

Lemma 2.18. *Two connected graphs that are not distinguishable by WL have isomorphic universal covers.*

Proof. Let G and G' be two connected graphs which are not distinguishable by WL. Let $\pi : T \to G$ and $\pi' : T' \to G'$ be their respective universal covers. By Lemma 2.12, since G and G' are indistinguishable by WL, so are T and T'. By Lemma 2.11, T and T' must therefore be isomorphic. $\square$

Proof of Theorem 1.1. One direction is obtained from Lemma 2.18, the other from Corollary 2.15. $\square$

3 THE GRAPHCOVERS DATASET

Our goal is to efficiently generate graphs that are indisinguishable by the WL test, but are pairwise non-isomorphic. To do so, Theorem 1.1 indicates that same degree covers of a common base graph are good candidates since they all share the same universal cover, and hence fool the WL test. However, we still need a way to distinguish non-isomorphic covers, and two covers can be non-isomorphic as covers but still be isomorphic as graphs. After showing that under mild assumptions on the base graph, both notions of isomorphism agree, we provide an efficient algorithm to test isomorphism of covers. We combine these into Algorithm 2 which returns GraphCovers(G, d), a set of representatives for the isomorphism classes of connected, degree d covers of a base graph G.

Theorem 3.1. *If color refinement distinguishes every vertex of a base graph G (i.e. the stable coloring $c : \mathcal{V}(G) \to \mathbb{N}$ is injective) then two covers $\pi : H \to G$ and $\pi' : H' \to G$ are isomorphic as covers if and only if they are isomorphic as graphs.*

Proof. If two covers are isomorphic, then the underlying graphs are isomorphic. We thus focus on the other direction. Let $\phi : H \to H'$ be an isomorphism of graphs. By Lemma 2.12, all nodes in $\pi^{-1}(v)$, and all nodes in $\pi'^{-1}(v)$ have the same color as v. Since isomorphisms are covers, colorings have to be preserved by the ϕ per Lemma 2.12. Since every node in G have different colors, we must have that $\phi : H \to H'$ maps $\pi^{-1}(v)$ to $\pi'^{-1}(v)$, thus making the diagram in Definition 2.4 commute. $\square$

In order to build the promised algorithm, we need a way to check if two covers are isomorphic as covers. Lemma 3.2 shows that this problem can be reduced to checking if we can extend a partial map defined on a single point to an isomorphism. In Proposition 3.3 we show that extending such maps can be done in polynomial time.

Lemma 3.2. *A morphism $\phi : H \to H'$ between connected coverings $\pi : H \to G$ and $\pi' : H' \to G'$ of a connected base graph G is uniquely determined by its value at any vertex.*

Proof. As for Lemma 2.13 we work locally, by showing that there is no choice when extending a partial $\tilde{v} \mapsto \tilde{v}'$ to the neighbors $\tilde{u} \in \mathcal{N}_H(\tilde{v})$. Let $\tilde{v} \in \mathcal{V}(H)$, and $\tilde{v}' := \phi(\tilde{v})$, and let $\tilde{u} \in \mathcal{N}_H(\tilde{v})$. Since π' is a covering map, there is a unique $\tilde{u}' \in \mathcal{N}_{H'}(v') \cap \pi'^{-1}(\pi(\tilde{u}))$, and thus we must send $\tilde{u}$ to this $\tilde{u}'$ to preserve commutativity of the diagram. $\square$

Proposition 3.3. *Let G be a connected graph satisfying the conditions of Theorem 3.1. There is a $\mathcal{O}(d^2 \cdot \mathcal{E}(G))$ algorithm that checks if two degree d covers of the base graph G are isomorphic as graphs.*

Proof. Let $\pi : H \to G$ and $\pi' : H' \to G$ be two covers of degree d. By Theorem 3.1, it suffices to check if there exists a cover isomorphism between π and π'. Fix a $\tilde{v} \in \mathcal{V}(H)$. By Lemma 3.2, it suffices to check if for some $\tilde{v}' \in \pi'^{-1}(\pi(v))$, the map $\tilde{v} \mapsto \tilde{v}'$ can be extended to a covering isomorphism from H to H'. This can be done by extending the map locally edge by edge, and checking that at each stage it is an isomorphism onto its image. Looping through every such $\tilde{v}'$ gives a $\mathcal{O}(d \cdot \mathcal{E}(H))$ algorithm. $\square$

Algorithm 2 Generate GraphCovers

    Inputs: Graph G as in Theorem 3.1, Degree $d > 0$
    Initialize GraphCovers$(G, d) = \emptyset$.
    for every connected degree d cover $\pi : H \to G$ **do**
        if $H \cong H'$ for some $H' \in$ GraphCovers(G, d) **then**
            do nothing
        else
            GraphCovers(G, d).add(H)
        end if
    end for
    return GraphCovers(G, d)

To implement Algorithm 2, we enumerate the degree d covers of G by selecting one permutation $\sigma_e \in S_d$ for each edge $e \in \mathcal{E}$, where S_d is the group of permutations on d elements. The cover H is then constructed by creating d copies $v_1, v_2, \ldots v_d$ for each $v \in \mathcal{V}$, and for each $e = vw \in \mathcal{E}$ we add edges connecting v_i to $w_{\sigma_e(i)}$ for $i = 1 \ldots d$. This procedure is used in Gross & Tucker (1977) (see Theorem 2 in that paper), and in Amit & Linial (2002). To capture all isomorphism classes it is enough to fix $\sigma_e = id$ except for certain distinguished edges, where the number of distinguished edges is equal to $1 - \chi(G)$. We use the algorithm described in Proposition 3.3 to check if two generated covers are isomorphic.

The limits of Algorithm 2 is that it generates only connected covers, and cannot generate a pair of graph with common cover that do not cover a common base graph (such as ones in Figure 1). Whereas the first point can be addressed, hence generalizing the construction of the CYCLES dataset from Horn et al. (2022); the second cannot.

4   A POLY-FACTORIAL LOWER BOUND TO THE SIZE OF WL EQUIVALENCE CLASSES

In the previous section, we gave an efficient algorithm that generates GraphCovers(G, d), a collection of graphs that fool the WL test but that are pairwise non-isomorphic. However, we do not know how big this collection is. In the case where G is a length n cycle, any degree d covering of G isomorphic to the cycle of degree $d \cdot n$, so our dataset would consist of only one graph, making it useless. This suggests that for GraphCovers(G, d) to be useful, the topology of the graph G has to be more complicated than that of a cycle. We thus show that if G has two or more cycles, then GraphCovers(G, d) grows poly-factorially with d:

Theorem 4.1. *Let G be a connected graph that satisfies the conditions of Theorem 3.1. The number $C_{d,r} = |GraphCovers(G, d)|$ of non-isomorphic connected degree d covers of G satisfies*

$$d \cdot C_{d,r} \geq N_{d,r}$$

where $N_{d,r}$ is the number of index d subgroups of the free group of rank $r = 1 - \chi(G)$. In particular,

$$C_{d,r} \geq d^{r-2}((d-1)!)^{r-1} .$$

The proof can be found in Appendix C [3]. The exact number of connected degree d covers was computed in Kwak & Lee (1996-10), but for our purpose we simply need a good lower bound. We combine Theorem 4.2 and Proposition 4.3 to obtain a lower bound on the number of non-isomorphic degree d connected covers of a connected base graph G, and therefore a lower bound on the size of GraphCovers(G, d). In Theorem 4.1 we reduce the problem to the one of counting index d subgroups in a rank r free group, where $r = 1 - \chi(G)$, and $\chi(G) = |\mathcal{V}| - |\mathcal{E}|$ is the Euler characteristic of the graph. This problem was solved in Hall (1949) leading to the following recursive formula:

Theorem 4.2. *Hall (1949) The number $N_{d,r}$ of subgroups of index d in the rank r free group F_r is given recursively by $N_{1,r} = 1$,*

$$N_{d,r} = d(d!)^{r-1} - \sum_{i=1}^{d-1} [(d-i)!]^{r-1} N_{i,r} .$$

---

[3]https://arxiv.org/abs/2206.11876

Table 1: Training accuracy of 2-layer deep nonlinear GNNs on 3 non-isomorphic, degree 5 covers of a graph with $\chi(G) = -1$.

MODEL	UNIQUE ID	RANDOM	DEGREE	CONSTANT
WL	N/A	N/A	33%	33%
GCN	100%	100%	33%	33%
CHEB	100%	100%	33%	33%
GIN	100%	100%	33%	33%
SAGE	100%	100%	33%	33%
GAT	100%	100%	33%	33%

The following is a well know fact from algebraic topology that we need, but do not prove in this paper. See Chapter 1 in Hatcher (2000) for the definition of the fundamental group, and Proposition 1A.2 for a proof of the following.

Proposition 4.3. *The fundamental group $\pi_1(G, p)$ based at $p \in \mathcal{V}(G)$ of a connected graph G is isomorphic to the free group F_r of rank $r = 1 - \chi(G)$.*

5 EXPERIMENTS

We now design experiments to benchmark our dataset. First, we take three connected non-isomorphic degree 5 covers of a connected base graph G with $|\mathcal{V}(G)| = 9$ and $\chi(G) = -1$ that satisfy the conditions of Theorem 3.1, see Figure 5 in Appendix D [4] for an illustration of the chosen G. All three resulting connected graphs have 45 nodes and 50 edges. The choices of the 3 graphs and the degree 5 was arbitrary, and repeating the experiment with different choices always gave the same results. We can now benchmark the dataset by measuring the performance of popular GNN architectures on the supervised graph classification task.

We consider five GNN architectures, each differing only by their message passing scheme. They consist of two message passing layers with ReLU nonlinearity, followed by global poolings (stack of mean and max pooling) and an MLP with output dimension 3. The five message passing schemes are ChebConv (Defferrard et al., 2016), GCN (Kipf & Welling, 2016), GraphSAGE (Hamilton et al., 2017), GAT (Veličković et al., 2018), and GIN (Xu et al., 2019).

We consider four different node features $\mathbf{x} : \mathcal{V} \to \mathbb{R}^c$. The constant signal equal to 1 everywhere, the degree signal mapping v to its degree, a random signal sampled from a 1-dimensional Gaussian $\mathcal{N}(0, 1)$ distribution, and a unique identifier signal where each node is mapped to a 45-dimensional binary vector of all 0s except 1 at a position uniquely determined by the node.

To make sure that the GNNs have the capacity to distinguish these graphs, the hyperparameters are chosen such that all architectures reach 100% training accuracy on the random and unique ID signals. The existence of such a choice of hyperparameters is expected by Loukas (2020a;b); Chen et al. (2020); Garg et al. (2020). The chosen hyperparameters are: cross entropy loss; Adam optimizer with learning rate 0.01 and weight decay $5 \cdot 10^{-4}$; hidden dimension 100; and each architecture is trained for 400 epochs on the 3 graphs. We report training accuracy in Table 1, to which we have also added the accuracy of the WL test on these graphs.

The architectures all perform poorly, as they learn only one of the three isomorphism classes. In fact, all the GNNs give constant predictions, since every graph's embedding in the final layer is the same, as expected by Theorem 1.1.

6 CONCLUSION

After giving a novel description of WL equivalence classes to the GNN community, we materialized the result to provide an arbitrarily large dataset on which message passing GNNs cannot do graph classification.

Our work strengthens the existing link between Algebraic Topology, Group Theory, and Geometric Deep Learning. We showed that GNNs capture two orthogonal structures of graphs: size (num-

---

[4]https://arxiv.org/abs/2206.11876

ber of nodes) and shape (isomorphism type of covers). We also showed that messages propagate equivalently on a graph and its universal cover which is a tree.

Future work includes benchmarking more complex architectures on GraphCovers, and studying the performance of higher order WL tests on GraphCovers.

GraphCovers can now be used to benchmark and study novel GNN architectures. We consider the contribution of a concrete dataset for which message passing does not work to be an essential step in improving GNNs expressiveness, and to go beyond message passing. We thus conclude by challenging the community to solve graph classification on the GraphCovers dataset for large degree covers.

SOFTWARE AND DATA

Code to generate GraphCovers(G, d) and to reproduce our experiments can be found on our github repository [5]. Experiments were mainly done using pytorch geometric (Fey & Lenssen, 2019) and Networkx (Hagberg et al., 2008).

ACKNOWLEDGEMENTS

REFERENCES

Ralph Abboud, Ismail Ilkan Ceylan, Martin Grohe, and Thomas Lukasiewicz. The surprising power of graph neural networks with random node initialization. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 2112–2118. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track.

Alon Amit and Nathan Linial. Random graph coverings i: General theory and graph connectivity. *Combinatorica*, 22(1):1–18, 2002.

Dana Angluin. Local and global properties in networks of processors. In *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*, pp. 82–93. Association for Computing Machinery, 1980. ISBN 0897910176.

Beatrice Bevilacqua, Fabrizio Frasca, Derek Lim, Balasubramaniam Srinivasan, Chen Cai, Gopinath Balamurugan, Michael M. Bronstein, and Haggai Maron. Equivariant subgraph aggregation networks. In *International Conference on Learning Representations*, 2022.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Liò, Guido F Montufar, and Michael Bronstein. Weisfeiler and lehman go cellular: Cw networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 2625–2640. Curran Associates, Inc., 2021a.

Cristian Bodnar, Fabrizio Frasca, Yu Guang Wang, Nina Otter, Guido Montufar, Pietro Liò, and Michael M. Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks. In *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, 2021b.

Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. Can graph neural networks count substructures? In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 10383–10395. Curran Associates, Inc., 2020.

Fan Chung and S.-T. Yau. Coverings, heat kernels and spanning trees. *The Electronic journal of combinatorics*, 6(1), 1998. ISSN 1077-8926.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *CoRR*, abs/1903.02428, 2019.

Vikas K. Garg, Stefanie Jegelka, and Tommi Jaakkola. Generalization and representational limits of graph neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 06–11 Aug 2017.

Jonathan L. Gross and Thomas W. Tucker. Generating all graph coverings by permutation voltage assignments. *Discrete Mathematics*, 18(3):273–283, 1977.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman (eds.), *Proceedings of the 7th Python in Science Conference*, pp. 11 – 15, Pasadena, CA USA, 2008.

Marshall Hall. Subgroups of finite index in free groups. *Canadian Journal of Mathematics*, 1(2): 187–190, 1949.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Allen Hatcher. *Algebraic topology*. Cambridge Univ. Press, Cambridge, 2000.

Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten Borgwardt. Topological graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2022.

Neil Immerman and Eric Lander. Describing graphs: A first-order approach to graph canonization. In *Complexity theory retrospective*, pp. 59–81. Springer, 1990.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2016.

Jin Ho Kwak and Jaeun Lee. Enumeration of connected graph coverings. *Journal of graph theory*, 23(2), 1996-10. ISSN 0364-9024.

Frank Thomson Leighton. Finite common coverings of graphs. *Journal of Combinatorial Theory, Series B*, 33(3):231–238, 1982.

Andreas Loukas. What graph neural networks cannot learn: depth vs width. In *International Conference on Learning Representations*, 2020a.

Andreas Loukas. How hard is to distinguish graphs with graph neural networks? In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3465–3476. Curran Associates, Inc., 2020b.

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4602–4609, Jul. 2019.

Pál András Papp, Karolis Martinkus, Lukas Faber, and Roger Wattenhofer. DropGNN: Random dropouts increase the expressiveness of graph neural networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

R. Sato, Makoto Yamada, and Hisashi Kashima. Random features strengthen graph neural networks. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, 2021.

Ryoma Sato, Makoto Yamada, and Hisashi Kashima. Approximation ratios of graph neural networks for combinatorial problems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

John R. Stallings. Topology of finite graphs. *Inventiones mathematicae*, 71(3):p551–565, 1983.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Boris Yu. Weisfeiler and Andrei A. Leman. The reduction of a graph to canonical form and the algebra which appears therein. *Nauchno-Technicheskaya Informatsia, Seriya 22 9, 12–16. In Russian*, 1968.

Daniel T. Wise. *Non-Positively Curved Squared Complexes: Aperiodic Tilings and Non-Residually Finite Groups*. PhD thesis, USA, 1996.

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