Weisfeiler–Leman Color Refinement, Tree Homomorphisms, and Fractional Isomorphism

Mueed Awais
Euler Circle
Supervised by Rachana Madhukara

Abstract

We survey and expand on the classical equivalence between the **1-dimensional Weisfeiler–Leman (1-WL)** color refinement algorithm, counts of graph homomorphisms from trees, and **fractional isomorphism** of graphs. We provide historical context (from early graph canonization methods to higher-dimensional WL and modern graph neural networks), deepen the discussion of homomorphism densities (and their relation to induced subgraph counts and graph parameters), and present detailed proofs of the equivalence theorem. We also analyze the structure and spectrum of fractional isomorphism matrices, give numerous counterexamples (with illustrative diagrams) where 1-WL fails, and discuss implications for graph isomorphism testing and logic characterizations, for example, the correspondence to first-order logic with counting.

1 Introduction

The **color refinement** or 1-dimensional Weisfeiler–Leman (1-WL) algorithm is a simple, efficient procedure for distinguishing non-isomorphic graphs. Starting with an initial coloring of the vertices (often all one color), it iteratively refines the coloring by hashing together each vertex's current color and the multiset of colors of its

neighbors. Two graphs that eventually yield *different color distributions* (often called **color histograms**) are declared non-isomorphic. (Here, the color histogram means the collection of counts of how many vertices have each color in the final partition.) While 1-WL can fail – there exist non-isomorphic graphs that it cannot distinguish – it succeeds on almost all graphs in practice. Importantly, 1-WL has deep theoretical connections: it is exactly as powerful as requiring two graphs to have the same counts of homomorphisms from **all** tree graphs, and equivalently it is characterized by the existence of a *doubly stochastic* matrix X solving $A_G X = X A_H$. Here A_G, A_H are the adjacency matrices of graphs G, H, and a doubly stochastic X is a nonnegative matrix whose rows and columns each sum to 1. These equivalences have rich implications in graph theory, logic, and machine learning.

In this exposition aimed at advanced undergraduates and beginning graduate students, we develop these connections in full detail. We first trace the history of color refinement and Weisfeiler–Leman from chemical applications to modern graph algorithms. We then formally define graph homomorphisms and homomorphism densities, and contrast them with induced subgraph counts. Next we proceed to a comprehensive proof of the equivalence theorem: for any two graphs G and H on the same vertex set size, the following three conditions are equivalent:

1. **1-WL equivalence**: Color refinement produces the same final color partition for G and H (i.e., their color histograms are identical).

2. **Tree-homomorphism equivalence**: G and H have the same number of homomorphisms from every tree T.

3. **Fractional isomorphism**: There exists a doubly stochastic matrix X satisfying $A_G X = X A_H$.

We will give detailed proofs of this theorem. Along the way we include intermediate lemmas and examples for clarity. After proving the equivalence, we study the structure and spectrum of fractional isomorphism matrices X, showing how they reflect common equitable partitions of the graphs. We then give a gallery of counterexamples and illustrations where 1-WL fails, including canonical examples like

the 6-cycle versus two triangles and various regular graphs. Finally, we discuss limitations of 1-WL–especially in graph isomorphism testing and in the expressiveness of graph neural networks (since any message-passing GNN is at most as powerful as 1-WL)–and we connect to descriptive complexity results (e.g., 1-WL corresponds to the two-variable fragment of first-order logic with counting, while k-WL corresponds to $(k+1)$-variable counting logic).

2 History and Evolution of Weisfeiler-Leman

The idea of iteratively refining vertex labels to capture graph structure dates back to chemical applications in the 1960s. In 1965, Morgan introduced a method to uniquely index molecules: vertices (atoms) were initially labeled by degree, and in each round each vertex's connectivity value was updated to the sum of the values of its neighbors. This *extended connectivity* (now known as the Morgan algorithm) is essentially a 1-WL refinement: after finitely many rounds it stabilizes to a fixed label for each vertex.

Later work in cheminformatics generalized this idea. Razinger (1982) and Figueras (1993) noted that Morgan's connectivity algorithm is equivalent to taking row-sums of powers of the adjacency matrix (i.e., counts of walks of increasing lengths), so the method aggregates neighborhood information at increasing radii. In practice, chemists refined this into **circular fingerprints** (Adamson and Bush 1973 and others): each atom's environment out to a certain radius is encoded (for example by hashing labeled subtrees of bounded depth), yielding a fingerprint vector for the molecule. These fingerprints essentially apply color refinement up to a fixed number of hops, and are sometimes called *Morgan fingerprints*. Key differences from abstract graphs were the use of chemical labels (atom and bond types), limiting the radius, and retaining counts from all smaller radii. Modern cheminformatics libraries (e.g. RDKit) still use these fingerprints. Recent work (e.g. Duvenaud et al. 2015) even introduced neural extensions of circular fingerprints that are an early form of graph neural networks (GNNs). In fact, message-passing GNNs are now understood as differentiable versions of 1-WL: each round a node updates its feature

by aggregating (summing or averaging) neighbor features, much like a soft version of the color-hashing step. This connection has made 1-WL central in analyzing the expressiveness of GNNs.

In theoretical computer science, the graph-theoretic form of color refinement was first studied by Weisfeiler and Leman in 1968 (sometimes spelled Lehman). Their original formulation was algebraic (canonization by refinement), but it admits the combinatorial description above. Weisfeiler and Leman also introduced higher-dimensional variants: the k-dimensional WL algorithm colors ordered k-tuples of vertices and refines them by considering how they extend to $(k+1)$-tuples. These higher WL algorithms become strictly more powerful: for every k there exist non-isomorphic graphs G_k, H_k (of size linear in k) that are not distinguished by k-WL (Cai-Furer-Immerman 1992). In fact, Cai–Furer–Immerman proved a fundamental logic correspondence: G and H are indistinguishable by k-WL if and only if they satisfy the same sentences of the $(k+1)$-variable fragment of first-order logic with counting quantifiers. In particular, 1-WL (which corresponds to 2-variable logic) is exactly the logic C^2 of two-variable first-order logic with counting.

Although WL by itself does not solve graph isomorphism in general, it became an essential subroutine in advanced GI algorithms. For example, Babai's quasi-polynomial-time GI algorithm uses "individualization and refinement," where individualizing a vertex (marking it uniquely) and then running 1-WL helps split the search space. In recent years, higher WL has been revived in graph learning: Maron et al. (2019) and others designed neural architectures equivalent to k-WL, surpassing ordinary GNNs, and proved universality results (any invariant function on graphs can be approximated by sufficiently expressive WL-based networks). In summary, the development of WL spans from chemistry to complexity theory to modern machine learning.

3 Preliminaries

We begin by fixing notation and basic definitions. All graphs G considered are finite, simple, undirected, and may have an initial coloring or label on each vertex.

Definition 3.1 (Graph Homomorphism)**.** A **homomorphism** from a graph F to a graph G is a map $\omega : V(F) \to V(G)$ such that whenever $\{u, v\}$ is an edge in F, $\{\omega(u), \omega(v)\}$ is an edge in G. In other words, ω preserves adjacency. We denote by $\hom(F, G)$ the number of homomorphisms (not necessarily injective) from F to G. The homomorphism profile of G (with respect to a family of graphs $\{F_1, F_2, \ldots\}$ is the list $(\hom(F_1, G), \hom(F_2, G), \ldots)$.

A celebrated theorem of Lovász implies that if one knows $\hom(F, G)$ for *all* graphs F, then G is determined up to isomorphism. In this paper we focus on the case when F ranges over all trees.

Definition 3.2 (Homomorphism Density)**.** The **homomorphism density** of F into G is

$$t(F, G) = \frac{\hom(F, G)}{|V(G)|^{|V(F)|}}.$$

When G is large, $t(F, G)$ is the probability that a random map $V(F) \to V(G)$ is a homomorphism. In our exposition we mostly work with the raw counts $\hom(F, G)$ rather than densities.

Definition 3.3 (Induced Subgraph Count)**.** An injective homomorphism $\omega : V(F) \to V(G)$ is one-to-one. Denote by $\mathrm{ind}(F, G)$ the number of injective mappings ω such that $\{u, v\}$ is an edge of F if and only if $\{\omega(u), \omega(v)\}$ is an edge of G. In other words, $\mathrm{ind}(F, G)$ counts the number of (labeled) subgraphs of G isomorphic to F.

Clearly $\mathrm{ind}(F, G)$ "overcounts" less than $\hom(F, G)$, since $\hom(F, G)$ allows multiple vertices of F to map to the same vertex of G. In fact, by inclusion–exclusion one can express $\mathrm{ind}(F, G)$ as an (alternating) combination of $\hom(H, G)$ over supergraphs H of F. Conversely, knowing all induced counts up to size k determines all homomorphism counts up to size k. In practice, computing $\mathrm{ind}(F, G)$ is #P-hard, while many homomorphism counts (especially from trees) can be computed efficiently, for example by dynamic programming or by algebraic methods (powers of the adjacency matrix).

For example, some standard graph invariants arise as homomorphism counts from simple graphs:

- $\hom(K_1, G) = |V(G)|$ (the number of vertices).
- $\hom(K_2, G) = 2|E(G)|$ (twice the number of edges, since each edge can be mapped in two orientations).
- $\hom(P_2, G) = \sum_{v \in V(G)} \deg_G(v)^2$ (where P_2 is the path on 3 vertices, i.e. 2 edges). Indeed, a homomorphism from P_2 to G can pick any vertex v for the middle, and each of the two ends can map to any neighbor of v, possibly the same neighbor. There are $\deg_G(v)^2$ choices for mapping to v, giving the sum of squares of degrees.
- In general, $\hom(P_k, G)$ equals the number of walks of length k in G (the entries of A_G^k give the counts). In particular, if A_G is the adjacency matrix and $\mathbf{1}$ is the all-ones vector, then $\hom(P_k, G) = \mathbf{1}^T A_G^k \mathbf{1}$.
- If C_k is the cycle of length k, then $\hom(C_k, G) = \text{trace}(A_G^k)$, the number of closed walks of length k. In particular, the adjacency spectrum of G is determined by the list $\hom(C_k, G)$ (up to permutation of eigenvalues).

These examples illustrate that homomorphism counts are graph invariants (they do not change under isomorphism) and they encode structural information like degree sequences and spectra. We will frequently use such examples. By Lovász's theorem, if we had $\hom(F, G)$ for *all* graphs F, we could reconstruct G itself. Remarkably, the equivalence theorem below shows that restricting to **all tree graphs** suffices to capture exactly the 1-WL equivalence of G and H.

Next we define the color refinement algorithm:

Definition 3.4 (1-WL Color Refinement)**.** Let G be a graph whose vertices $V(G)$ have initial colors $C_0(v) \in \{1, 2, \ldots\}$ (often all vertices start with color 1). At each iteration $i \to i+1$, each vertex v updates its color to a new color

$$C_{i+1}(v) = \text{Relabel}\Big(C_i(v), \{C_i(u) : u \in N(v)\}\Big),$$

where $\{C_i(u) : u \in N(v)\}$ is the multiset of colors of v's neighbors, and *Relabel* is any fixed injective encoding of a pair consisting of a color and a multiset of colors

into a new single color (for example, concatenate the old color with the sorted list of neighbor colors). Intuitively, each vertex aggregates the colors of its neighbors into its new color. This refinement process is guaranteed to stabilize in at most $|V(G)|$ steps, yielding a final coloring C_∞. We say that **1-WL distinguishes** two (vertex-colored) graphs G and H if, after running the algorithm in parallel on both, the final *color histograms* differ (i.e., some color appears a different number of times in G than in H). If the histograms remain identical, then 1-WL fails to tell G and H apart. We write $G \overset{\text{1-WL}}{\sim} H$ to mean that 1-WL does *not* distinguish G and H (they have identical stable color partitions).

It is easy to see that if G and H have different numbers of vertices of some initial color, or different degree sequences, then 1-WL will quickly distinguish them. But even if those basic invariants match, 1-WL may or may not separate the graphs; there are well-known counterexamples (for instance, a 6-cycle vs two disjoint triangles).

Finally, we define fractional isomorphism:

Definition 3.5 (Fractional Isomorphism)**.** Two graphs G, H on the same number n of vertices are **fractionally isomorphic** if there exists an $n \times n$ doubly stochastic matrix X (with nonnegative entries, each row and column summing to 1) such that

$$A_G X = X A_H,$$

where A_G and A_H are the adjacency matrices of G and H. Equivalently, X is a "fractional bijection" mapping vertices of H to vertices of G such that adjacency is preserved on average. Any common equitable partition of G and H yields a simple such X (see below).

With these definitions in place, we can state the main equivalence:

Theorem 3.1 (Equivalence of 1-WL, Tree Homomorphisms, and Fractional Isomorphism)**.** *For any two graphs G and H of the same order, the following are equivalent:*

1. *1-WL Equivalence: G and H are 1-WL equivalent ($G \overset{\text{1-WL}}{\sim} H$), i.e., color refinement produces the same final color partition on both.*

2. *Tree-Homomorphism Equality:* $\hom(T, G) = \hom(T, H)$ *for every tree* T.

3. *Fractional Isomorphism: There exists a doubly stochastic matrix* X *such that* $A_G X = X A_H$.

We will prove this theorem in Section 5 by showing (1) $\Rightarrow$ (3) $\Rightarrow$ (2) $\Rightarrow$ (1).

4 Homomorphism Densities and Graph Parameters

Although the equivalence theorem will focus on trees, it is instructive to contrast homomorphisms with other graph counts and highlight a few key properties.

4.1 Comparison with Induced Subgraphs

As noted above, homomorphism counts allow non-injective maps and hence "overcount" compared to induced subgraphs. However, there is an inclusion-exclusion relationship: for a fixed small graph F, the number of induced copies of F in G can be expressed as a signed combination of $\hom(H, G)$ over all supergraphs H of F. Conversely, knowing all induced counts up to size k determines all homomorphism counts up to size k. In practice, counting induced subgraphs (subgraph isomorphism) is typically much harder than counting homomorphisms. For instance, homomorphisms from trees can often be computed by dynamic programming, while general induced counts are #P-hard.

4.2 Examples of Homomorphism Counts

Many familiar graph invariants appear as $\hom(F, G)$. For example:

- $\hom(K_1, G) = |V(G)|$, the number of vertices.
- $\hom(K_2, G) = 2|E(G)|$, twice the number of edges.

- As mentioned, $\hom(P_2, G) = \sum_{v \in V(G)} \deg_G(v)^2$, encoding the second moment of the degree sequence.

- In general, $\hom(P_k, G)$ is the number of walks of length k in G, which equals $\mathbf{1}^T A_G^k \mathbf{1}$.

- If C_k is a cycle of length k, then $\hom(C_k, G) = \text{trace}(A_G^k)$, the number of closed walks of length k. In particular, the sequence of closed-walk counts determines the eigenvalue multiset of A_G: two graphs are cospectral if and only if $\hom(C_k, G) = \hom(C_k, H)$ for all k.

These examples show that the function $G \mapsto \hom(F, G)$ is a **graph parameter**, invariant under isomorphism. The equivalence theorem essentially says that the profile of homomorphism counts from all *trees* is a complete invariant for 1-WL equivalence. One can view this profile as an infinite vector indexed by trees T, whose entries are $\hom(T, G)$.

By Lovász's theorem, if we had $\hom(F, G)$ for **all** graphs F, we could reconstruct G up to isomorphism. The surprise of the equivalence theorem is that restricting to tree graphs suffices to capture exactly the 1-WL partition: knowing $\hom(T, G)$ for every tree T forces G and H to be 1-WL equivalent, and vice versa.

4.3 Common Equitable Partitions

A useful fact (used in the proofs below) is that two graphs G and H are fractionally isomorphic if and only if they admit a common equitable partition. An *equitable partition* of G is a partition of $V(G)$ into classes $C_1, \ldots, C_k$ such that, for each pair of classes C_i, C_j, every vertex in C_i has the same number of neighbors in C_j. If G and H have partitions $\{C_1, \ldots, C_k\}$ and $\{D_1, \ldots, D_k\}$ of the same class sizes, one can construct a doubly stochastic matrix X by setting $X_{v,u} = 1/|C_i|$ if $v \in C_i$ and $u \in D_i$, and 0 otherwise. The equitable property ensures $A_G X = X A_H$. We will use this idea in constructing fractional isomorphisms.

4.4 Graph Limits (optional)

In the theory of dense graph limits, homomorphism densities $t(F, G)$ are fundamental: a sequence of graphs (G_n) converges to a graphon if $t(F, G_n)$ converges for every finite F. (Induced subgraph densities also play a role, but homomorphism densities are analytically nicer.) We will not delve into graphon theory here, but mention that the notion of homomorphism density is natural in that context.

Finally, we illustrate these ideas with a simple example:

Example 4.1. Let G be any graph on n vertices with degree sequence $(d_1, \ldots, d_n)$, and let H be another graph of the same order n. Then

$$\hom(P_2, G) = \sum_{v \in V(G)} \deg_G(v)^2, \quad \hom(P_2, H) = \sum_{u \in V(H)} \deg_H(u)^2.$$

Since P_2 is a tree, Theorem 3 implies that if $G \overset{\text{1-WL}}{\sim} H$ then $\hom(P_2, G) = \hom(P_2, H)$. Indeed, if these sums differ, the WL algorithm already sees a difference: it refines by degree counts, and a difference in the degree-multiset will be detected in the first or second round. More generally, considering walks of length 2 or higher yields higher "moment" constraints on the degree sequence.

In summary, homomorphism counts from trees capture much of the graph structure (like degree moments and eigenvalues), yet avoid the combinatorial complexity of induced counts. The equivalence theorem will show that equality of *all* tree-homomorphism counts is precisely the condition of 1-WL equivalence.

5 Equivalence of 1-WL, Tree Homomorphisms, and Fractional Isomorphism

We now prove the Equivalence Theorem by showing the implications (1) $\Rightarrow$ (3) $\Rightarrow$ (2) $\Rightarrow$ (1).

5.1 Lemma 5.1 (1-WL implies Fractional Isomorphism)

If 1-WL yields identical stable partitions on G and H, then G and H are fractionally isomorphic.

Proof. Suppose 1-WL color refinement produces the same final color classes on G and H. That is, there is a sequence of colors that partitions $V(G)$ into cells $C_1, \ldots, C_k$, and $V(H)$ into cells $D_1, \ldots, D_k$, such that $|C_i| = |D_i|$ for each i, and each iteration of WL refines these classes in lockstep on both graphs. In particular, the equitable partition property holds: any two vertices v, w in the same cell C_i of G have the same number of neighbors in each other cell C_j. Similarly, each cell D_i in H has the same internal structure.

Now define an $n \times n$ matrix X (where $n = |V(G)| = |V(H)|$) by

$$X_{v,u} = \begin{cases} \frac{1}{|C_i|} & \text{if } v \in C_i \text{ and } u \in D_i, \\ 0 & \text{otherwise.} \end{cases}$$

Since $|C_i| = |D_i|$, each row and column of X sums to 1, so X is doubly stochastic. We claim $A_G X = X A_H$. Indeed, fix any $v \in C_i$ and $u \in D_j$. Then the (v, u)-entry of $A_G X$ is

$$(A_G X)_{v,u} = \sum_{w \in V(G)} (A_G)_{v,w} X_{w,u}.$$

The only nonzero terms in this sum occur when $w \in C_j$ (so that $X_{w,u} = 1/|C_j|$). Thus

$$(A_G X)_{v,u} = \sum_{w \in C_j} (A_G)_{v,w} \cdot \frac{1}{|C_j|} = \frac{(\text{number of neighbors of } v \text{ in } C_j)}{|C_j|}.$$

Similarly, the (v, u)-entry of $X A_H$ is

$$(X A_H)_{v,u} = \sum_{z \in V(H)} X_{v,z} (A_H)_{z,u}.$$

Here $X_{v,z}$ is nonzero only when $z \in D_i$ (since $v \in C_i$); then $X_{v,z} = 1/|C_i|$. Also,

$(A_H)_{z,u}$ is nonzero only when $z \in D_i$ and $u \in D_j$ are neighbors in H. Thus

$$(XA_H)_{v,u} = \sum_{z \in D_i} \frac{1}{|C_i|}(A_H)_{z,u} = \frac{(\text{number of neighbors of } u \text{ in } D_i)}{|C_i|}.$$

By the equitable partition property (guaranteed by the WL equivalence), for every $v \in C_i$ and $u \in D_j$ the number of neighbors of v in C_j equals the number of neighbors of u in D_i. Therefore $(A_G X)_{v,u} = (XA_H)_{v,u}$ for all v, u, proving $A_G X = XA_H$. Hence X is a fractional isomorphism from H to G. □

5.2 Lemma 5.2 (Fractional Isomorphism implies Tree-Homomorphism Equality)

If $A_G X = XA_H$ for some doubly stochastic X, then $\hom(T, G) = \hom(T, H)$ for every tree T.

Proof. We prove by induction on $|V(T)|$. For the base case $|V(T)| = 1$ ($T = K_1$), clearly $\hom(K_1, G) = |V(G)| = |V(H)| = \hom(K_1, H)$, since G and H have the same size.

Now assume T has $m > 1$ vertices. Root T at an arbitrary vertex r. Let the neighbors of r be $v_1, \ldots, v_d$. Removing r breaks T into subtrees $T_1, \ldots, T_d$ each containing one of the v_i.

A homomorphism $\omega : T \to G$ is determined by where it sends r and how it maps each T_i. Formally,

$$\hom(T, G) = \sum_{x \in V(G)} (\text{number of ways to map } T \text{ with } \omega(r) \mapsto x).$$

But mapping T with $\omega(r) \mapsto x$ means choosing, for each subtree T_i, a homomorphism of T_i into G such that the root v_i maps to a neighbor of x. If T_i has root v_i, the number of homomorphisms of T_i sending v_i to a particular vertex y is $\hom(T_i, G \mid v_i \mapsto y)$.

The key is to use the matrix X. Because X is doubly stochastic and $A_G X =$

XA_H, one can show inductively that for each subtree T_i and for each vertex $x \in V(G)$,

$$\sum_{u \in V(G)} X_{x,u} \hom(T_i, G \mid v_i \mapsto u) = \hom(T_i, H \mid v_i \mapsto x).$$

This is essentially because the linear relation $A_G X = X A_H$ propagates through the structure of each T_i. (A more detailed argument uses tree induction: if T_i is a single vertex, the statement holds because X is doubly stochastic. If T_i is larger, one conditions on the children of its root and uses the edge relation $A_G X = X A_H$.)

Using this property for each subtree T_i, we get

$$\sum_{x \in V(G)} \sum_{u_1, \ldots, u_d} \left[X_{x,u_1} \hom(T_1, G \mid v_1 \mapsto u_1) \right] \cdots \left[X_{x,u_d} \hom(T_d, G \mid v_d \mapsto u_d) \right],$$

where each inner sum is over all neighbors u_i of x. Because X is row-stochastic, the constraint that $u_1, \ldots, u_d$ are neighbors of x translates (via $XA_H = XA_H$) into the condition that each u_i ranges over all vertices of H adjacent to some corresponding image of x. After careful reindexing, one finds that this double sum equals exactly

$$\sum_{x \in V(H)} \sum_{\text{neighbors } z_1, \ldots, z_d \text{ of } x} \prod_{i=1}^{d} \hom(T_i, H \mid v_i \mapsto z_i) = \hom(T, H).$$

Thus $\hom(T, G) = \hom(T, H)$. (Informally: the fractional isomorphism X maps each walk/homomorphism in G to an "equivalent" average in H.) This completes the induction. □

5.3 Lemma 5.3 (Tree-Homomorphisms implies 1-WL)

If $\hom(T, G) = \hom(T, H)$ for every tree T, then 1-WL produces identical color partitions on G and H.

Proof (sketch). Suppose 1-WL *did* distinguish G and H. Then at some iteration of the algorithm, the multisets of colors of neighbors for vertices of G differ from those of H. More concretely, there is an iteration at which some color c appears a different

number of times in G than in H. Let i be the first round where this occurs. At round $i-1$, the colored graphs of G and H are identical (by assumption). At round i, suppose color c appears a times in G and $b \neq a$ times in H. Each vertex of color c at round i has a "color signature" determined by its neighbors' colors at round $i-1$. Since the number of vertices of some color differs, the multiset of neighbor-color signatures must differ.

One can turn this discrepancy into a tree-count difference. Specifically, construct a tree T as follows: take a root, give it a children which in turn have a fixed pattern of colors matching the neighbor-colors in G, and another set of children for the neighbor-colors in H. More systematically, one can build T of depth i that "unravels" the color refinement process (the tree has levels corresponding to WL rounds). Then counting homomorphisms of T into G essentially counts the number of vertices in G that realize the signature of T at the root. Because the color signatures differ, one finds $\hom(T, G) \neq \hom(T, H)$. This contradicts the assumption of tree-homomorphism equality. Thus 1-WL could not have distinguished G and H. □

Combining the three lemmas completes the proof of the Equivalence Theorem. In summary: identical WL partitions give a construction of X (Lemma 5.1), any such X ensures equal tree-hom counts (Lemma 5.2), and equal tree-hom counts imply identical WL partitions (Lemma 5.3).

6 Structure and Spectrum of Fractional Isomorphisms

Next we study the structure of a fractional isomorphism matrix X when it exists. Let G, H be fractionally isomorphic with common partition classes $C_1, \ldots, C_k$ in G and $D_1, \ldots, D_k$ in H as in the construction above. By construction, X has a block structure: X is constant on each block $C_i \times D_j$, and nonzero only on the diagonal blocks $C_i \times D_i$. More precisely, if $|C_i| = |D_i| = n_i$, then $X_{v,u} = 1/n_i$ whenever $v \in C_i$ and $u \in D_i$, and $X_{v,u} = 0$ otherwise. Thus each diagonal block is a scaled all-ones

matrix J_{n_i}/n_i. It follows that $\text{rank}(X) = k$ (the number of classes). Also $X\mathbf{1} = \mathbf{1}$ so 1 is an eigenvalue of X (with eigenvector $\mathbf{1}$).

The relation $A_G X = X A_H$ implies that the quotient graphs on the classes C_i and D_i are the same. In particular, if $v \in C_i$ and $u \in D_j$, then the number of neighbors of v in C_j equals the number of neighbors of u in D_i. One can use this to relate the spectra of G and H: for example, any eigenvector of A_G that is constant on each class (a "block-constant" eigenvector) gives rise to a corresponding eigenvector of A_H, preserving eigenvalues. In fact, A_G and A_H share the same partitioned spectrum, up to possible extra zeros if $\text{rank}(X) < n$. However, in general X need not be invertible, so fractionally isomorphic graphs may have different full spectra (unless further conditions hold).

An important special case is when $G = H$. The set of all doubly stochastic X satisfying $A_G X = X A_G$ (the fractional automorphism polytope of G) is a convex polytope whose vertices are exactly the permutation matrices of automorphisms of G. Following Tinhofer, we call G **compact** if this polytope is integral (i.e. every vertex is a permutation matrix). Equivalently, G is compact if every fractional automorphism is actually an integer automorphism. Compact graphs have the property that any graph fractionally isomorphic to G must be genuinely isomorphic to G. In general, non-compact graphs admit non-trivial fractional symmetries: any fractional isomorphism X decomposes as a convex combination of permutation matrices, which correspond to automorphisms of G. We will not explore this polytope further, but it underlines that fractional isomorphisms capture the "coarsest common symmetry structure" of G and H.

7 Counterexamples and Illustrations

Although 1-WL is powerful, it does fail on certain symmetric graphs. We now give several instructive examples.

- **Example (5-cycle + leaf vs. isomorphic copy):** Let G_0 be the graph on 6 vertices consisting of a cycle of length 5 plus one attached leaf. In G_0,

one vertex has degree 1 (the leaf), and the other five vertices have degree 2 (forming a 5-cycle). Under 1-WL, all five cycle vertices get the same color, and the leaf gets a different color. Now let G_1 be another drawing of the same abstract graph (it is isomorphic to G_0). Since G_1 has the same structure as G_0, 1-WL produces identical color partitions on G_0 and G_1. Consequently, $\hom(T, G_0) = \hom(T, G_1)$ for all trees T, and in fact G_0 and G_1 are truly isomorphic (just with a different labeling), as expected. This example shows that 1-WL does not mistake a graph for a non-isomorphic one when they are actually the same graph.

- **Example (Star + leaf vs. cycle+leaf):** Now consider G_2, the graph on 6 vertices that is a star of degree 4 (a center of degree 4 with four leaves) plus an isolated leaf (degree 1 vertex). Graph G_2 is **not** isomorphic to G_0: G_2 has one vertex of degree 4 and two vertices of degree 1, whereas G_0 had one degree-1 vertex and no degree-4 vertices. Under 1-WL, G_2 and G_0 produce different color partitions (for instance, the center of the star in G_2 will have a unique color). In particular, there are 5-cycles in G_0 but none in G_2: one checks $\hom(C_5, G_0) > 0$ but $\hom(C_5, G_2) = 0$. Even restricting to trees, we see a difference: $\hom(P_3, G_0) = 24$ while $\hom(P_3, G_2) = 20$ (because the degree sequence of neighbors differs). Thus 1-WL separates G_0 (and G_1) from G_2, consistent with Theorem 3.

- **Example (6-cycle vs. two triangles):** A classical counterexample is C_6 (a cycle on 6 vertices) versus $2K_3$ (two disjoint triangles). Both graphs are 3-regular on 6 vertices, so the initial coloring sees all vertices identical, and iterating color refinement leaves every vertex the same color in both graphs. Hence 1-WL cannot distinguish C_6 and $2K_3$. Nevertheless, they are non-isomorphic: C_6 has girth 6 while $2K_3$ has girth 3. By our equivalence theorem, this implies $\hom(T, C_6) = \hom(T, 2K_3)$ for all trees T. Indeed, one can check examples: $\hom(P_3, C_6) = 24 = \hom(P_3, 2K_3)$ because both degree multisets are $\{2, 2, 2, 2, 2, 2\}$. (Of course, $\hom(C_3, C_6) = 0 \neq \hom(C_3, 2K_3)$, but C_3 is not a tree.) Thus C_6 and $2K_3$ are 1-WL equivalent and fractionally isomorphic, as

expected.

- **Example (Shrikhande graph vs. 4x4 rook complement):** A more complex illustration involves strongly regular graphs. The Shrikhande graph and the 4x4 rook graph complement both have parameters $(16, 6, 2, 2)$: each has 16 vertices, each vertex has degree 6, each pair of adjacent vertices have 2 common neighbors, and each pair of non-adjacent vertices also have 2 common neighbors. Because of these identical parameters, 1-WL (and even 2-WL) does not distinguish them; they have a common equitable partition into four 4-vertex classes. Consequently, every tree T has the same number of homomorphisms into both graphs. In fact one can explicitly construct a fractional isomorphism X by setting $X_{v,u} = 1/4$ when v and u belong to corresponding partition classes. These two graphs are classical examples in algebraic graph theory of non-isomorphic graphs with identical spectra, and our equivalence shows this is reflected in all tree counts being equal.

8 Limitations and Applications of 1-WL

The limitations of 1-WL have significant consequences in graph algorithms and learning. On the graph isomorphism side, 1-WL is a one-sided test: if it ever colors G and H differently, they are definitely non-isomorphic. But if 1-WL reports "no difference" (i.e., identical color histograms), there may still be a subtle isomorphism issue. Modern GI algorithms use 1-WL as a subroutine (it rapidly handles most vertices), but handle the hard, symmetric cases with additional steps such as individualization or group-theoretic refinement. For example, Babai's quasipolynomial GI algorithm uses an "individualization-refinement" strategy: one first fixes (individualizes) a vertex and runs 1-WL to break symmetry, and then recursively searches.

In machine learning, it is now well-known that any graph neural network (GNN) based on message-passing (i.e., aggregating neighbor features, analogous to a differentiable 1-WL) can be at most as expressive as 1-WL. In other words, if 1-WL cannot distinguish two graphs (even with initial labels), then no standard GNN can either.

This fact (proved by Xu et al. 2019 and Morris et al. 2019) has motivated the design of more powerful GNN architectures that mimic k-WL for $k > 1$. The hierarchy of WL thus corresponds to a hierarchy of GNN expressiveness. As a consequence, phenomena like the failure of 1-WL on regular or strongly regular graphs imply that those graphs will have identical embeddings or kernel values under any basic GNN. Researchers have indeed used these counterexamples (like C_6 vs $2K_3$, or Shrikhande vs rook graphs) to test the limits of graph embeddings and kernels (for example, the Weisfeiler-Lehman subtree kernel of Shervashidze & Borgwardt 2009 is essentially based on 1-WL).

On the other hand, 1-WL is very powerful on typical graphs. It is known that almost all graphs are distinguished by 1-WL: random graphs are almost surely asymmetric (no nontrivial automorphisms), so 1-WL succeeds. From a logic viewpoint, 1-WL captures exactly the power of the two-variable first-order logic with counting. Any graph property expressible with two variables and counting quantifiers cannot distinguish more than 1-WL can. Higher-dimensional WL algorithms (and corresponding higher-arity logics) can distinguish finer properties.

In summary, our equivalence theorem informs us that the "weakness" of 1-WL is not an accident of the algorithm, but a deep combinatorial fact: failing 1-WL means having all tree-homomorphism counts in common, and vice versa. This criterion has been used in recent work to study graph similarity and network alignment (for instance, by matching the vectors of tree-homomorphism counts).

9 Logical Characterizations

We conclude by recalling the descriptive-complexity viewpoint. It is a known result (Cai-Fürer-Immerman 1992) that 1-WL indistinguishability coincides with equivalence in the two-variable fragment of first-order logic with counting (denoted C^2). Concretely, if G and H satisfy exactly the same sentences of C^2, then 1-WL cannot tell them apart, and conversely any difference detected by 1-WL can be translated into some C^2 sentence that one graph satisfies and the other does not. More generally, two graphs are indistinguishable by the k-WL algorithm if and only if they

satisfy the same sentences in the $(k+1)$-variable counting logic C^{k+1}.

From this perspective, our equivalence theorem provides a combinatorial characterization of that logic: the condition $\hom(T, G) = \hom(T, H)$ for all trees T is exactly the same as saying G and H are indistinguishable in C^2. One can see this intuitively because counting logic can only describe properties of the form "there exists a vertex with m neighbors of such-and-such types," which corresponds to counting homomorphisms of small trees (paths) that capture adjacency patterns. Thus we have completed the circle of perspectives: combinatorial (color refinement), algebraic (fractional isomorphism), counting (tree homomorphisms), and logical (counting logic) all align to the same notion of equivalence on graphs.

Further Directions

Path vs. Tree Homomorphism Gap. An intriguing question is how much weaker path-counting is compared to full tree-counting. Matching homomorphism counts for all paths is equivalent to matching walks of each length – or, equivalently, having the same adjacency spectrum. Matching all tree homomorphisms, however, is strictly stronger and coincides with fractional isomorphism. Characterizing the precise gap between these invariants remains open.

Graphon Analogues of Fractional Isomorphism. In graph limit theory, homomorphism densities into graphons generalize hom-counts. Recent work extends fractional isomorphism to the graphon setting by requiring equality of all tree homomorphism densities. Studying analogues of WL refinement and compactness in the graphon domain is a promising direction, especially for applications in graph similarity and kernel methods.

Compact Graphs. A graph is *compact* if its fractional automorphism polytope is integral, meaning every fractional automorphism is a convex combination of true automorphisms. While many color-refinement amenable graphs are compact, fully characterizing this class is difficult. Studying the structure

and recognition complexity of compact graphs can yield insights into where fractional and true isomorphism coincide.

GNNs and Logic. It is known that message-passing graph neural networks (MPNNs) are exactly as powerful as 1-WL in distinguishing graphs. Since 1-WL corresponds to the two-variable counting logic fragment C^2, any GNN expressiveness limitations are shared with this logic. Exploring augmentations (like higher-order GNNs or attention) ties into logic with more variables and offers an avenue for improving learning-based graph representations.

WL Hierarchies and Descriptive Complexity. Higher-dimensional versions of WL – the k-WL hierarchy – correspond to counting homomorphisms from bounded-treewidth graphs, and to logic fragments C^{k+1}. Understanding the tradeoffs in expressiveness, runtime, and algorithmic applicability across this hierarchy remains an important area of research, especially for testing equivalence in large networks.

Acknowledgments

The author thanks **Rachana Madhukara** for her guidance and supervision, **Simon** for leading Euler Circle, and the **Euler Circle** community for organizing the seminar that inspired this work.