On the Expressive Power of Spectral Invariant Graph Neural Networks

Bohang Zhang [1]   **Lingxiao Zhao** [2]   **Haggai Maron** [3] [4]

Abstract

Incorporating spectral information to enhance Graph Neural Networks (GNNs) has shown promising results but raises a fundamental challenge due to the inherent ambiguity of eigenvectors. Various architectures have been proposed to address this ambiguity, referred to as spectral invariant architectures. Notable examples include GNNs and Graph Transformers that use spectral distances, spectral projection matrices, or other invariant spectral features. However, the potential expressive power of these spectral invariant architectures remains largely unclear. The goal of this work is to gain a deep theoretical understanding of the expressive power obtainable when using spectral features. We first introduce a unified message-passing framework for designing spectral invariant GNNs, called Eigenspace Projection GNN (EPNN). A comprehensive analysis shows that EPNN essentially unifies all prior spectral invariant architectures, in that they are either strictly less expressive or equivalent to EPNN. A fine-grained expressiveness hierarchy among different architectures is also established. On the other hand, we prove that EPNN itself is bounded by a recently proposed class of Subgraph GNNs, implying that all these spectral invariant architectures are strictly less expressive than 3-WL. Finally, we discuss whether using spectral features can gain additional expressiveness when combined with more expressive GNNs.

1. Introduction

Recent works have demonstrated the promise of using spectral graph features, particularly the eigenvalues and eigenvectors of the graph Laplacian or functions thereof, as positional and structural encodings in Graph Neural Networks (GNNs) and Graph Transformers (GTs) (Dwivedi et al., 2020; Dwivedi & Bresson, 2021; Kreuzer et al., 2021; Rampasek et al., 2022; Kim et al., 2022). These spectral features encapsulate valuable information about graph connectivity, inter-node distances, node clustering patterns, and more. When using eigenvectors as inputs for machine learning models, a major challenge arises due to the inherent eigenspace symmetry (Lim et al., 2023) — eigenvectors are not unique. Specifically, for any eigenvector v, $-v$ is also a valid eigenvector. The ambiguity becomes worse in the case of repeated eigenvalues; here, any orthogonal transformation of the basis vectors in a particular eigenspace yields alternative but equivalent input representations.

To address the ambiguity problem, a major line of recent works leverages *invariant* features derived from eigenvectors and eigenvalues to design spectral invariant architectures. Popular choices for such features include eigenspace projection matrices[1] (Lim et al., 2023; Huang et al., 2024), spectral node distances (e.g., those associated with random walks or graph diffusion) (Li et al., 2020; Zhang et al., 2023b; Feldman et al., 2023), or other invariant spectral characteristics (Wang et al., 2022). All of these features can be easily integrated into GNNs to enhance edge features or function as relative positional encoding of GTs. However, on the theoretical side, while the expressive power of GNNs has been studied extensively (Xu et al., 2019; Maron et al., 2019a; Morris et al., 2021; Geerts & Reutter, 2022; Zhang et al., 2024), there remains little understanding of the important category represented by spectral invariant GNNs/GTs.

Current work. The goal of this work is to gain deep insights into the expressive power of spectral invariant architectures and establish a complete expressiveness hierarchy. We begin by presenting *Eigenspace Projection GNN* (EPNN), a novel GNN framework that unifies the study of all the aforementioned spectral invariant methods. EPNN is very simple: it encodes all spectral information for a node pair (u, v) as a *set* containing the values of all projection matrices on that node pair, along with the associated eigenvalues. It then computes and refines node representations using the spectral information as edge features within a standard message-passing framework on a fully connected graph.

[1]Peking University [2]Carnegie Mellon University [3]Technion [4]NVIDIA Research. Correspondence to: Bohang Zhang <zhang-bohang@pku.edu.cn>, Haggai Maron <hmaron@nvidia.com>.

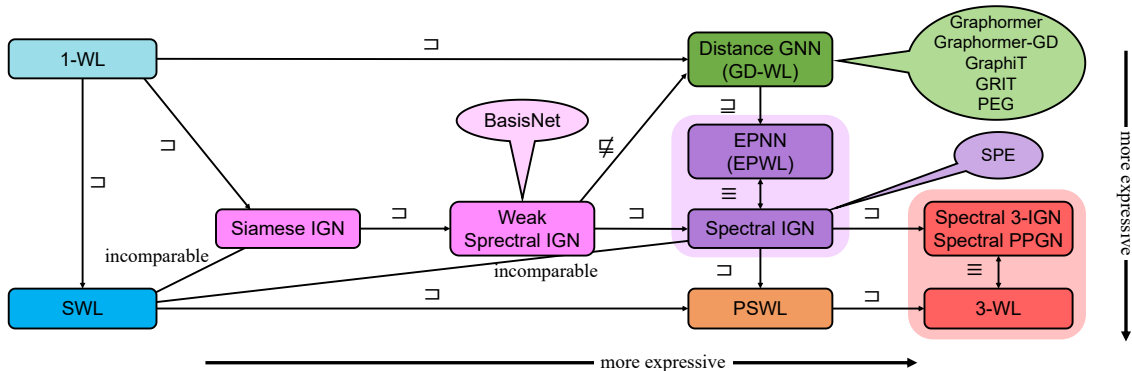[1]See Section 3 for a formal definition of projection matrices.

Figure 1. Expressive hierarchy for all GNN architectures studied in this paper. Here, the symbol "≡" means that the two GNNs being compared have the same expressive power; "⊐" means that the latter GNN is *strictly* more expressive than the former one; "⊒" means that the latter GNN is either strict more expressive than or as expressive as the former one; "⊄̸" means that the latter GNN is (strictly) not less expressive than the former one. Finally, "incomparable" means that either GNN is (strictly) not more expressive than the other. The dialog bubbles list literature architectures that can be seen as instantiations of the corresponding GNN class.

Our first theoretical result establishes a tight expressiveness upper bound for EPNN, showing that it is strictly less expressive than an important class of Subgraph GNNs proposed in Zhang et al. (2023a), called PSWL. This observation is intriguing for two reasons. First, it connects spectral invariant GNNs and GTs with the seemingly unrelated research direction of Subgraph GNNs (Cotta et al., 2021; Bevilacqua et al., 2022; Frasca et al., 2022; Qian et al., 2022; Zhao et al., 2022) — a line of research studying expressive GNNs from a structural and permutation symmetry perspective. Second, combined with recent results (Frasca et al., 2022; Zhang et al., 2023a), it implies that EPNN is strictly bounded by 3-WL. As an implication, bounding previously proposed spectral invariant methods by EPNN would readily indicate that they are all strictly less expressive than 3-WL.

We then explore how EPNNs are related to GNNs/GTs that employ *spectral distances* as positional encoding (Ying et al., 2021; Mialon et al., 2021; Zhang et al., 2023b; Ma et al., 2023b; Wang et al., 2022; Li et al., 2020). We prove that under the general framework proposed in Zhang et al. (2023b), all commonly used spectral distances give rise to models with an expressive power bounded by EPNNs. This highlights an inherent expressiveness limitation of distance-based approaches in the literature. Moreover, our analysis underscores the crucial role of message-passing in enhancing the expressive power of spectral features.

Our next step aims to draw connections between EPNNs and two important spectral invariant architectures that utilize projection matrices, known as Basisnet (Lim et al., 2023) and SPE (Huang et al., 2024). This is achieved by a novel symmetry analysis for eigenspace projections, which yields a theoretically-inspired architecture called Spectral IGN. Surprisingly, we prove that Spectral IGN is *as expressive as* EPNN. On the other hand, SPE and BasisNet can be easily upper bounded by either Spectral IGN or its weaker variant.

Finally, we discuss the potential of using spectral features to boost the expressive power of higher-order GNNs. We show using the projection matrices alone does not provide any additional expressive power advantage when combined with highly expressive GNNs such as PPGN and k-IGN (Maron et al., 2019b;a). Nevertheless, we propose a possible solution towards further expressiveness gains: we hypothesize that stronger expressivity could be achieved through higher-order extensions of graph spectra, such as projection tensors. Overall, our theoretical results characterize an expressiveness hierarchy across basis invariant GNNs, distance-based GNNs, GTs with spectral encoding, subgraph-based GNNs, and higher-order GNNs. The resulting hierarchy is illustrated in Figure 1.

2. Related GNN Models

2.1. Spectrally-enhanced GNNs

In recent years, a multitude of research has emerged to develop spectrally-enhanced GNNs/GTs, integrating graph spectral information into either GNN node features or the subsequent message-passing process. These endeavors can be categorized into the following three groups.

Laplacian eigenvectors as absolute positional encoding. One way to design spectrally-enhanced GNNs involves encoding Laplacian eigenvectors. This approach treats each eigenvector as a 1-dimensional node feature and incorporates the top k eigenvectors as a type of absolute positional encoding, which can be used to enhance any message-passing GNNs and GTs (Dwivedi et al., 2020; Dwivedi & Bresson, 2021; Kreuzer et al., 2021; Rampasek et al., 2022; Maskey et al., 2022; Dwivedi et al., 2022; Kim et al., 2022). However, one main drawback of using Laplacian eigenvectors arises from the *ambiguity* problem. Such ambiguity creates severe issues regarding training instability

and poor generalization (Wang et al., 2022). While this problem can be partially mitigated through techniques like randomly flipping eigenvector signs or employing a canonization method (Ma et al., 2023a), it becomes much more complicated when eigenvalues have higher multiplicities.

Spectral invariant architectures. A better approach would be to design GNNs that are *invariant* w.r.t. the choice of eigenvectors. For example, SignNet (Lim et al., 2023) transforms each eigenvector $\boldsymbol{v}$ to $\phi(\boldsymbol{v}) + \phi(-\boldsymbol{v})$ for some permutation equivariant function ϕ, which guarantees the invariance when all eigenvalues have a multiplicity of 1. In case of higher multiplicity, BasisNet (Lim et al., 2023) achieves spectral invariance for the first time by utilizing the projection matrix. Specifically, given an eigenvalue λ with multiplicity k, the projection matrix defined as $\sum_{i=1}^{k} \boldsymbol{v}_i \boldsymbol{v}_i^\top$ is invariant w.r.t. the choice of (unit) eigenvectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k$ as long as they form an orthogonal basis of the eigenspace associated with λ. Therefore, BasisNet simply feeds the projection matrix into a permutation equivariant model $\rho : \mathbb{R}^{n \times n} \to \mathbb{R}^n$ (e.g., 2-IGN (Maron et al., 2019b)) to generate spectral invariant node features $\rho(\sum_{i=1}^{k} \boldsymbol{v}_i \boldsymbol{v}_i^\top)$. The node features generated for different eigenspaces are concatenated together. While the authors proved that Basis-Net can universally represent any graph functions when ρ is universal (e.g., using n-IGN), the empirical performance is generally unsatisfactory when employing a practical model ρ (i.e., 2-IGN). Recently, Huang et al. (2024) further generalized BasisNet by proposing SPE, which performs a soft aggregation across different eigenspaces rather than a hard separation implemented in BasisNet. Specifically, let $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n$ be an orthogonal basis of (unit) eigenvectors associated with eigenvalues $\lambda_1, \cdots, \lambda_n$, respectively; then, each 1-dimensional node feature generated by SPE has the form $\rho(\sum_{i=1}^{n} \psi_j(\lambda_i) \boldsymbol{v}_i \boldsymbol{v}_i^\top)$, where $\psi_j : \mathbb{R} \to \mathbb{R}$ is a parameterized function associated with feature dimension j. The authors demonstrated that SPE can enhance the stability and generalization of GNNs, yielding much better empirical performance compared with BasisNet.

Spectral distances as invariant relative positional encoding. In contrast to encoding Laplacian eigenvectors, an alternative approach to achieving spectral invariance involves utilizing (spectral) distances. Previous studies have identified various distances, spanning from the basic shortest path distance (Feng et al., 2022; Abboud et al., 2022) to more advanced ones such as PageRank distance, resistance distance, and distances associated with random walks and graph diffusion (Li et al., 2020; Zhang et al., 2023b; Mialon et al., 2021; Feldman et al., 2023). Notably, all of these distances have a deep relation to the graph Laplacian while being more interpretable than eigenvectors and not suffering from ambiguity problems. The work of PEG (Wang et al., 2022) designed an invariant *relative* positional encoding based on Laplacian eigenvectors, which can also be treated as a dis-

tance between nodes. Distances can be easily encoded in GNN models by either serving as edge features in message-passing aggregations (Wang et al., 2022; Velingker et al., 2023) or as relative positional encoding in Graph Transformers (Ying et al., 2021; Zhang et al., 2023a; Ma et al., 2023b).

2.2. Expressive GNNs

The expressive power of GNNs has been studied in depth in the recent few years. Early works (Xu et al., 2019; Morris et al., 2019) have pointed out a fundamental limitation of GNNs by establishing an equivalence between message-passing neural networks and the 1-WL graph isomorphism test (Weisfeiler & Lehman, 1968). To develop more expressive models, several studies leveraged high-dimensional variants of the WL test (Cai et al., 1992; Grohe, 2017). Representative models include k-IGN (Maron et al., 2019b), PPGN (Maron et al., 2019a), and k-GNN (Morris et al., 2019; 2020). However, these models suffer from severe computational costs and are generally not suitable in practice. Currently, one mainstream approach to designing simple, efficient, practical, and expressive architectures is the Subgraph GNNs (Cotta et al., 2021; Bevilacqua et al., 2022; 2023; You et al., 2021; Zhang & Li, 2021; Zhao et al., 2022; Kong et al., 2023). In particular, the expressive power of Subgraph GNNs as well as their relation to the WL tests are well-understood in recent studies (Frasca et al., 2022; Qian et al., 2022; Zhang et al., 2023a; 2024). These results will be used to analyze spectrally-enhanced GNNs in this paper.

2.3. Expressive power of spectral invariant GNNs

While spectrally-enhanced GNNs have been extensively studied in the literature, much less is known about their expressive power. Balcilar et al. (2021); Wang & Zhang (2022) delved into the expressive power of specific spectral filtering GNNs, but their expressive power is inherently limited by 1-WL. Another line of works studied the expressive power of the *raw* spectral invariants (e.g., projection matrices) in relation to the Weisfeiler-Lehman algorithms (Fürer, 1995; 2010; Rattan & Seppelt, 2023). However, their analysis does not consider any aggregation or refinement procedures over spectral invariants, and thus, it does not provide explicit insights into the expressive power of the corresponding GNNs. Lim et al. (2023) proposed a concrete spectral invariant GNN called BasisNet, but their expressiveness analysis still largely focuses on *raw* eigenvectors and projection matrices. To our knowledge, none of the prior works addresses the crucial problem of whether/how the design of *GNN layers* contributes to the model's expressiveness. In this paper, we will answer this question by showing that (i) a suitable aggregation procedure can strictly improve the expressive power beyond raw spectral features, and (ii) different aggregation schemes can lead to considerable variations in the models' expressiveness.

3. Preliminaries

We use $\{\ \}$ and $\{\!\{\ \}\!\}$ to denote sets and multisets, respectively. Given a (multi)set S, its cardinality is denoted as $|S|$. In this paper, we consider finite, undirected, simple graphs with no isolated vertices. Let $G = (V_G, E_G)$ be a graph with vertex set V_G and edge set E_G, where each edge in E_G is represented as a set $\{u, v\} \subset V_G$ of cardinality two. The *neighbors* of a vertex $u \in V_G$ is denoted as $N_G(u) = \{v \in V_G : \{u, v\} \in E_G\}$, and the degree of u is denoted as $\deg_G(u) = |N_G(u)|$. Given vertex pair $(u, v) \in V_G^2$, denote by $\mathrm{atp}_G(u, v)$ its atomic type, which encodes whether $u = v$, $\{u, v\} \in E_G$, or u and v are not adjacent. Given vertex tuple $\boldsymbol{u} \in V_G^k$, the rooted graph $G^{\boldsymbol{u}}$ is a graph obtained from G by marking vertices $u_1, \cdots, u_k$ sequentially. We denote by $\mathcal{G}$ the set of all graphs and by $\mathcal{G}_k$ the set of all rooted graphs marking k vertices. It follows that $\mathcal{G}_0 = \mathcal{G}$.

Graph invariant. Two (rooted) graphs $G^{\boldsymbol{u}}, H^{\boldsymbol{v}} \in \mathcal{G}_k$ are called *isomorphic* (denoted by $G^{\boldsymbol{u}} \simeq H^{\boldsymbol{v}}$) if there is a bijection $f : V_G \to V_H$ such that $f(u_i) = v_i$ for all $i \in [k]$, and for all vertices $w_1, w_2 \in V_G$, $\{w_1, w_2\} \in E_G$ iff $\{f(w_1), f(w_2)\} \in E_H$. A function f defined on graphs $\mathcal{G}_k$ is called a graph invariant if it is invariant under isomorphism, i.e., $f(G^{\boldsymbol{u}}) = f(H^{\boldsymbol{v}})$ if $G^{\boldsymbol{u}} \simeq H^{\boldsymbol{v}}$. In the context of graph learning, any GNN that outputs a graph representation should be a graph invariant over $\mathcal{G}_0$; similarly, any GNN that outputs a representation for each node/each pair of nodes should be a graph invariant over $\mathcal{G}_1/\mathcal{G}_2$, respectively.

Graph vectors and matrices. Any real-valued graph invariant x defined over $\mathcal{G}_1$ corresponds to a graph vector $\boldsymbol{x}_G : V_G \to \mathbb{R}$ when restricting on a specific graph $G \in \mathcal{G}$. Without ambiguity, we denote the elements in $\boldsymbol{x}_G$ as $\boldsymbol{x}_G(u)$ for each $u \in V_G$, which is equal to $x(G^u)$. Similarly, any real-valued graph invariant M defined over $\mathcal{G}_2$ corresponds to a graph matrix $\boldsymbol{M}_G : V_G^2 \to \mathbb{R}$ when restricting on $G \in \mathcal{G}$, where element $\boldsymbol{M}_G(u, v)$ equals to $M(G^{uv})$. For ease of reading, we will drop the subscript G when there is no ambiguity of the graph used in context. One can generalize all basic linear algebras from classic vectors/matrices to those defined on graphs. For example, the matrix product is defined as $(\boldsymbol{M}_1 \boldsymbol{M}_2)(u, v) = \sum_{w \in V_G} \boldsymbol{M}_1(u, w) \boldsymbol{M}_2(w, v)$. Several basic graph matrices include the adjacency matrix $\boldsymbol{A}$, degree matrix $\boldsymbol{D}$, Laplacian matrix $\boldsymbol{L} := \boldsymbol{D} - \boldsymbol{A}$, and normalized Laplacian matrix $\hat{\boldsymbol{L}} := \boldsymbol{D}^{-1/2} \boldsymbol{L} \boldsymbol{D}^{-1/2}$. Note that all these matrices are symmetric.

Graph spectra and projection. Let $\boldsymbol{M}$ be any symmetric graph matrix (e.g., $\boldsymbol{A}$, $\boldsymbol{L}$, or $\hat{\boldsymbol{L}}$). The graph spectrum is the set of all eigenvalues of $\boldsymbol{M}$, which is a graph invariant over $\mathcal{G}_0$. In addition to eigenvalues, the spectral information of a graph also includes eigenvectors or eigenspaces, which contain much more fine-grained information. Unfortunately, eigenvectors have inherent ambiguity and cannot serve as a valid graph invariant over $\mathcal{G}_1$. Instead, we fo-

cus on the eigenspaces characterized by their projection matrices. Concretely, there is a *unique* projection matrix $\boldsymbol{P}_i$ for each eigenvalue λ_i, which can be obtained via the eigen-decomposition $\boldsymbol{M} = \sum_{i \in [m]} \lambda_i \boldsymbol{P}_i$, where m is the number of different eigenvalues. It follows that these projection matrices are symmetric, idempotent ($\boldsymbol{P}_i^2 = \boldsymbol{P}_i$), "orthogonal" ($\boldsymbol{P}_i \boldsymbol{P}_j = \boldsymbol{O}$ for all $i \neq j$), and sum to identity ($\sum_{i \in [m]} \boldsymbol{P}_i = \boldsymbol{I}$). There is a close relation between projection matrix $\boldsymbol{P}_i$ and any orthogonal basis of unit eigenvectors $\{\boldsymbol{z}_{i,1}, \cdots, \boldsymbol{z}_{i,J_i}\}$ that spans the eigenspace associated with λ_i: specifically, $\boldsymbol{P}_i = \sum_{j=1}^{J_i} \boldsymbol{z}_{i,j} \boldsymbol{z}_{i,j}^\top$. The projection matrices naturally define a graph invariant $\mathcal{P}^{\boldsymbol{M}}$ over $\mathcal{G}_2$:

$$\mathcal{P}_G^{\boldsymbol{M}}(u, v) := \{\!\{(\lambda_1, \boldsymbol{P}_1(u, v)), \cdots, (\lambda_m, \boldsymbol{P}_m(u, v))\}\!\}.$$

We call $\mathcal{P}^{\boldsymbol{M}}$ the eigenspace projection invariant (associated with graph matrix $\boldsymbol{M}$).

4. Eigenspace Projection Network

This section introduces a simple GNN design paradigm based on the invariant $\mathcal{P}^{\boldsymbol{M}}$ defined above, called Eigenspace Projection GNN (EPNN). The idea of EPNN is very simple: $\mathcal{P}_G^{\boldsymbol{M}}(u, v)$ essentially encodes the relation between vertices u and v in graph G and can thus be treated as a form of "edge feature". In light of this, one can naturally define a message-passing GNN that updates the node representation of each vertex u by iteratively aggregating the representations of other nodes v along with edge features associated with (u, v). Formally, consider a K-layer EPNN and denote by $\boldsymbol{h}_G^{(l)}(u)$ the node representation of $u \in V_G$ computed by an EPNN after the l-th layer. Then, we can write the update rule of each EPNN layer as follows:

$$\begin{aligned}
\boldsymbol{h}_G^{(l+1)}(u) = g^{(l+1)}(\boldsymbol{h}_G^{(l)}(u), \\
\{\!\{(\boldsymbol{h}_G^{(l)}(v), \mathcal{P}_G^{\boldsymbol{M}}(u, v)) : v \in V_G\}\!\}),
\end{aligned} \tag{1}$$

where all node representations $\boldsymbol{h}_G^{(0)}(u)$ are the same at initialization. Here, $g^{(l+1)}$ can be any parameterized function representing the $(l+1)$-th layer. In practice, it can be implemented in various ways such as GIN-based aggregation (Xu et al., 2019) or Graph Transformers (Ying et al., 2021), and we highlight that it is particularly suited for Graph Transformers as the graph becomes fully connected with this edge feature. Finally, after the K-th layer, a global pooling is performed over all vertices in the graph to obtain the graph representation $\mathsf{POOL}(\{\!\{\boldsymbol{h}_G^{(K)}(u) : u \in V_G\}\!\})$.

EPNN is well-defined. First, the graph representation computed by an EPNN is permutation invariant w.r.t. vertices, as $\mathcal{P}^{\boldsymbol{M}}$ is a graph invariant over $\mathcal{G}_2$. Second, EPNN does not suffer from the eigenvector ambiguity problem, as $\mathcal{P}_G^{\boldsymbol{M}}$ is uniquely determined by graph G. Later, we will show that EPNN can serve as a simple yet unified framework for studying the expressive power of spectral invariant GNNs.

4.1. Expressive power of EPNN

The central question we would like to study is: *what is the expressive power of EPNN?* To formally study this question, this subsection introduces EPWL (Eigenspace Projection Weisfeiler-Lehman), an abstract color refinement algorithm tailored specifically for graph isomorphism test. Compared with Equation (1), in EPWL the node representation $h_G^{(l+1)}(u)$ is replaced by a color $\chi_G^{(l+1)}(u)$, and the aggregation function $g^{(l+1)}$ is replaced by a perfect *hash function* hash, as presented below:

$$\chi_G^{(l+1)}(u) = \mathsf{hash}(\chi_G^{(l)}(u), \{\!\!\{(\chi_G^{(l)}(v), \mathcal{P}_G^{\boldsymbol{M}}(u,v)) : v \in V_G\}\!\!\}). \tag{2}$$

Initially, all node colors $\chi_G^{(0)}(u)$ are the same. For each iteration l, the color mapping $\chi_G^{(l)}$ induces an equivalence relation over vertex set V_G, and the relation gets *refined* with the increase of l. Therefore, with a sufficiently large number of iterations $l \leq |V_G|$, the relations get *stable*. The graph representation is then defined to be the multiset of stable colors. EPWL distinguishes two non-isomorphic graphs iff the computed graph representations are different. We have the following result (which can be easily proved following standard techniques, see e.g., Zhang et al. (2023a)):

Proposition 4.1. *The expressive power of EPNN is bounded by EPWL in terms of graph isomorphism test. Moreover, with sufficient layers and proper functions $g^{(l)}$, EPNN can be as expressive as EPWL.*

In subsequent analysis, we will bound the expressive power of EPWL by building relations to the standard Weisfeiler-Lehman hierarchy. First, it is easy to see that EPWL is *lower bounded* by the classic 1-WL defined below:

$$\tilde{\chi}_G^{(l+1)}(u) = \mathsf{hash}(\tilde{\chi}_G^{(l)}(u), \{\!\!\{(\tilde{\chi}_G^{(l)}(v), \mathsf{atp}_G(u,v)) : v \in V_G\}\!\!\}), \tag{3}$$

where $\tilde{\chi}_G^{(l)}(u)$ is the 1-WL color of u in graph G after l iterations. This result follows from the fact that the EPWL color mapping $\chi^{(l+1)}$ always induces a *finer* relation than the 1-WL color mapping $\tilde{\chi}^{(l+1)}$, as $\mathsf{atp}_G(u,v)$ is fully encoded in $\mathcal{P}_G^M(u,v)$ (see Lemma A.5). Besides, it is easy to give 1-WL indistinguishable graphs that can be distinguished via spectral information (see Figure 2). Putting these together, we arrive at the following conclusion:

Proposition 4.2. *For any graph matrix $\boldsymbol{M} \in \{\boldsymbol{A}, \boldsymbol{L}, \hat{\boldsymbol{L}}\}$, the corresponding EPWL is strictly more expressive than 1-WL in distinguishing non-isomorphic graphs.*

On the other hand, the question becomes more intriguing when studying the *upper bound* of EPWL. Below, we will approach the problem by building fundamental connections between EPWL and an important class of expressive GNNs known as Subgraph GNNs. The basic form of Subgraph GNN is very simple: given a graph G, it treats G as a set of

rooted graphs $\{\!\!\{G^u : u \in V_G\}\!\!\}$ (known as node marking), independently runs 1-WL for each G^u, and finally merges their graph representations. We call the above algorithm SWL, and the refinement rule can be formally written as

$$\chi_G^{\mathsf{S},(l+1)}(u,v) = \mathsf{hash}(\chi_G^{\mathsf{S},(l)}(u,v),$$
$$\{\!\!\{(\chi_G^{\mathsf{S},(l)}(u,w), \mathsf{atp}_G(v,w)) : w \in V_G\}\!\!\}). \tag{4}$$

where $\chi_G^{\mathsf{S},(l)}(u,v)$ is the SWL color of vertex v in graph G^u after l iterations, and the initial color $\chi_G^{\mathsf{S},(0)}(u,v) = \mathbb{I}[u = v]$ distinguishes the marked vertex u in G^u. Recently, Zhang et al. (2023a); Frasca et al. (2022) significantly generalized Subgraph GNNs by enabling interactions among different subgraphs and built a complete design space. Among them, Zhang et al. (2023a) proposed the PSWL algorithm, which adds a cross-graph aggregation to SWL as shown below:

$$\chi_G^{\mathsf{PS},(l+1)}(u,v) = \mathsf{hash}(\chi_G^{\mathsf{PS},(l)}(u,v), \chi_G^{\mathsf{PS},(l)}(v,v),$$
$$\{\!\!\{(\chi_G^{\mathsf{PS},(l)}(u,w), \mathsf{atp}_G(v,w)) : w \in V_G\}\!\!\}), \tag{5}$$

where $\chi_G^{\mathsf{PS},(l)}(u,v)$ is the PSWL color of $(u,v) \in V_G^2$ after l iterations. We now present our main result, which reveals a fundamental connection between PSWL and EPWL:

Theorem 4.3. *For any graph matrix $\boldsymbol{M} \in \{\boldsymbol{A}, \boldsymbol{L}, \hat{\boldsymbol{L}}\}$, the expressive power of EPWL is strictly bounded by PSWL in distinguishing non-isomorphic graphs.*

The proof of Theorem 4.3 is deferred to Appendix A.3, which is based on the recent graph theory result established by Rattan & Seppelt (2023). Specifically, given any graph G and vertices $u, v \in V_G$, each projection element $\boldsymbol{P}_i(u,v)$ in $\mathcal{P}^{\boldsymbol{M}}$ is determined by the SWL stable color $\chi_G^{\mathsf{S}}(u,v)$ for any symmetric "equitable" matrix $\boldsymbol{M}$ defined in Rattan & Seppelt (2023), and the eigenvalues are also determined by the SWL graph representation. Notably, all matrices studied in this paper (e.g., $\boldsymbol{A}, \boldsymbol{L}, \hat{\boldsymbol{L}}$) are equitable. Based on this result, one may guess that EPWL can be bounded by SWL. However, we show this is actually not the case when further taking the message-passing aggregation into account. The key technical contribution in our proof is to relate the refinement procedure in EPWL to the additional cross-graph aggregation $\chi_G^{\mathsf{PS},(l)}(v,v)$ in PSWL. To this end, we show the stable color $\chi_G^{\mathsf{PS}}(u,u)$ is strictly finer than the stable color $\chi_G(u)$, thus concluding the proof.

Remark 4.4. Based on Theorem 4.3, one can also bound the expressiveness of EPWL by other popular GNNs in literature, such as SSWL (Zhang et al., 2023a), Local 2-GNN (Morris et al., 2020; Zhang et al., 2024), RelGN(2) (Frasca et al., 2022), ESAN (Bevilacqua et al., 2022), and GNN-AK (Zhao et al., 2022), as all these architectures are more expressive than PSWL (Zhang et al., 2023a). However, EPWL is incomparable to the vanilla SWL, where we give counterexamples in Appendix A.8.

The significance of Theorem 4.3 is twofold. First, it reveals a surprising relation between GNNs augmented with spectral information and the ones grounded in structural message-passing, which represents two seemingly unrelated research directions. Our result thus offers insights into how previously proposed expressive GNNs can encode spectral information. Second, Theorem 4.3 points out a fundamental limitation of EPNN. Indeed, combined with the result that PSWL is strictly bounded by 3-WL (Zhang et al., 2023a; 2024), we obtain the concluding corollary:

Corollary 4.5. *For any graph matrix $M \in \{A, L, \hat{L}\}$, the expressive power of EPWL is strictly bounded by 3-WL.*

5. Distance GNNs and Graph Transformers

In this section, we will show that EPNN unifies all distance-based spectral invariant GNNs. Here, we adopt the framework proposed in Zhang et al. (2023b), known as Generalized Distance (GD) GNNs. The aggregation formula of the corresponding GD-WL can be written as follows:

$$\chi_G^{\text{D},(l+1)}(u) = \text{hash}(\chi_G^{\text{D},(l)}(u), \tag{6}$$
$$\{\!\{(\chi_G^{\text{D},(l)}(v), d_G(u,v)) : v \in V_G\}\!\}).$$

where $\chi_G^{\text{D},(l)}(u)$ is the GD-WL color of vertex $u \in V_G$ after l iterations, and d can be any valid distance metric. By choosing different distances, GD-WL incorporates various prior works listed below.

- **Shortest-path distance (SPD).** This is the most basic distance metric and has been extensively used in designing expressive GNNs (e.g., Li et al., 2020; Ying et al., 2021; Abboud et al., 2022; Feng et al., 2022).

- **Resistance distance (RD).** It is defined to be the effective resistance between two nodes when treating the graph as an electrical network where each edge has a resistance of 1Ω. Recently, Zhang et al. (2023b) showed that incorporating RD can significantly improve the expressive power of GNNs for biconnectivity problems such as identifying cut vertices/edges. Besides, RD has been extensively studied in other areas in the GNN community, such as alleviating oversquashing problems (Arnaiz-Rodríguez et al., 2022).

- **Distances based on random walk.** The hitting-time distance (HTD) between two vertices u and v is defined as the expected number of steps in a random walk starting from u and reaching v for the first time, which is an asymmetric distance. Instead, the commute-time distance (CTD) is defined as the expected number of steps for a round-trip starting at u to reach v and then return to u, which is a symmetrized version of HTD. These distances are fundamental in graph theory and have been used to develop/understand expressive GNNs (Velingker et al., 2023; Zhang et al., 2023a).

- **PageRank distance (PRD).** Given weight sequence $\gamma_0, \gamma_1, \cdots$, the PageRank distance between vertices u and v is defined as $\sum_{k=0}^{\infty} \gamma_k W^k(u,v)$, where $W = D^{-1}A$ is the random walk probability matrix. It is a generalization of the p-step landing probability distance, which corresponds to setting $\gamma_p = 1$ and $\gamma_k = 0$ for all $k \neq p$. Li et al. (2020) first proposed to use PRD-WL to boost the expressive power of GNNs.

- **Other distances.** We also study the (normalized) diffusion distance (Coifman & Lafon, 2006) and the biharmonic distance (Lipman et al., 2010). Due to space limit, please refer to Appendix A.4 for more details.

Our main result is stated as follows:

Theorem 5.1. *For any distance listed above, the expressive power of GD-WL is upper bounded by EPWL with the normalized graph Laplacian matrix $\hat{L}$.*

The proof of Theorem 4.3 is highly technical and is deferred to Appendix A.4, with several important remarks made as follows. For the case of SPD, the proof is based on the key finding that $\mathcal{P}_G^{\hat{L}}(u,v)$ determines the shortest path distance between u and v for any graph G and vertices $u, v \in V_G$. Unfortunately, this property does not transfer to other distances listed above. For general distances, the reason why EPWL is still more expressive lies in the entire *message-passing process* (or color refinement procedure). Concretely, the refinement continuously enriches the information embedded in node colors $\chi_G^{(l)}(v)$, so that the tuple $(\chi_G^{(l)}(u), \chi_G^{(l)}(v), \mathcal{P}_G^{\hat{L}}(u,v))$ eventually encompasses sufficient information to determine any distance $d_G(u,v)$ (although $\mathcal{P}_G^{\hat{L}}(u,v)$ alone may not determine it). Our proof thus emphasizes the critical role of message-passing aggregation in enhancing the expressiveness of spectral information. Note that this is also justified in the proof of Theorem 4.3, where the message-passing process boosts the expressive power of EPWL beyond SWL. Moreover, we emphasize that these distances listed above cannot be well-encoded when using weaker message-passing aggregations, as will be elucidated in Section 6.2.

Implications. Theorem 5.1 has a series of consequences. First, it implies that all the power of distance information is possessed by EPWL. As an example, we immediately have the following corollary based on the relation between distance and biconnectivity of a graph established in Zhang et al. (2023b), which significantly extends the classic result that Laplacian spectrum encodes graph connectivity (Brouwer & Haemers, 2011).

Corollary 5.2. *EPWL is fully expressive for encoding graph biconnectivity properties, such as identifying cut vertices and cut edges, determining the number of biconnected components, and distinguishing graphs with non-isomorphic block cut-vertex trees and block cut-edge trees.*

Remark 5.3. Corollary 5.2 offers a novel understanding of the work by Zhang et al. (2023b) on why ESAN can encode graph biconnectivity, thereby thoroughly unifying their analysis. Essentially, this is just because ESAN is more powerful than EPWL (Remark 4.4), and EPWL itself is already capable of encoding both SPD and RD.

Remark 5.4. Combining Theorems 4.3 and 5.1 resolves an open question posed by Zhang et al. (2023a), confirming that PSWL can encode resistance distance.

As a second implication, combined with Corollary 4.5, Theorem 5.1 highlights a fundamental limitation of all distance-based GNNs as stated below:

Corollary 5.5. *For any distance defined above, the expressive power of GD-WL is strictly bounded by 3-WL.*

Distances as positional encoding. Distances have also found extensive application as positional encoding in GNNs and GTs. For instance, Graphormer (Ying et al., 2021), Graphormer-GD (Zhang et al., 2023b), and GraphiT (Mialon et al., 2021) employ various distances as relative positional encoding in Transformer's attention layers. Similarly, GRIT (Ma et al., 2023b) employs multi-dimensional PRD and a novel attention mechanism to further boost the performance of GTs. The positional encoding devised in PEG (Wang et al., 2022) can also be viewed as a function of a distance, with the architecture representing an instantiation of GD-WL. These architectures are analyzed in Appendix A.7, where we have the following concluding corollary:

Corollary 5.6. *The expressive power of Graphormer, Graphormer-GD, GraphiT, GRIT, and PEG are all bounded by EPWL and strictly less expressive than 3-WL.*

6. Spectral Invariant Graph Network

To gain an in-depth understanding of the expressive power of EPNN, in this section we will switch our attention to a more principled perspective by studying how to model spectral invariant GNNs based on the *symmetry* of projection matrices. Consider a graph G with vertex set $V_G = \{1, \cdots, n\}$. We can group all spectral information of G into a 4-dimensional tensor $\mathbf{P} \in \mathbb{R}^{m \times n \times n \times 2}$, where $\mathbf{P}_{i,u,v,1} = \lambda_i$ encodes the i-th eigenvalue and $\mathbf{P}_{i,u,v,2} = P_i(u, v)$ encodes the (u, v)-th element of the projection matrix P_i associated with λ_i. From this representation, one can easily figure out the symmetry group associated with $\mathbf{P}$, which is the *product group* $S_m \times S_n$ of two permutation groups S_m and S_n representing two independent symmetries — eigenspace symmetry and graph symmetry. For any element $(\sigma, \tau) \in S_m \times S_n$, it acts on $\mathbf{P}$ in the following way:

$$[(\sigma, \tau) \cdot \mathbf{P}]_{i,u,v,j} = \mathbf{P}_{\sigma^{-1}(i), \tau^{-1}(u), \tau^{-1}(v), j}. \quad (7)$$

We would like to design a GNN model f that is invariant under $S_m \times S_n$, i.e., $f((\sigma, \tau) \cdot \mathbf{P}) = f(\mathbf{P})$ for all $(\sigma, \tau) \in S_m \times S_n$. Interestingly, this setting precisely corresponds to a specific case of the DSS framework proposed in Maron et al. (2020); Bevilacqua et al. (2022). This offers a theoretically inspired approach to designing powerful invariant architectures, as we will detail below.

6.1. Spectral IGN

In the DSS framework, the neural network f is formed by stacking a series of equivariant linear layers $L^{(i)}$ interleaved with elementwise non-linear activation ϕ, culminating in final pooling layers:

$$f = M \circ \phi \circ g \circ \phi \circ L^{(K)} \circ \phi \circ \cdots \circ \phi \circ L^{(1)}, \quad (8)$$

where each $L^{(l)} : \mathbb{R}^{m \times n \times n \times d_{l-1}} \to \mathbb{R}^{m \times n \times n \times d_l}$ is equivariant w.r.t. $S_m \times S_n$, i.e., for all $(\sigma, \tau) \in S_m \times S_n$,

$$L^{(l)}((\sigma, \tau) \cdot \mathbf{X}) = (\sigma, \tau) \cdot L^{(l)}(\mathbf{X}) \quad \forall \mathbf{X} \in \mathbb{R}^{m \times n \times n \times d_{l-1}};$$

$g : \mathbb{R}^{m \times n \times n \times d_K} \to \mathbb{R}^{d_K}$ is an invariant pooling layer (e.g., average pooling or max pooling), and $M : \mathbb{R}^{d_K} \to \mathbb{R}^{d_{K+1}}$ is a multi-layer perceptron. Here, the key question lies in designing linear layers $L^{(l)}$ equivariant to the product group $S_m \times S_n$. Maron et al. (2020) theoretically showed that $L^{(l)}$ can be decomposed in the following way:

$$[L^{(l)}(\mathbf{X})]_i = \tilde{L}_1^{(l)}(\mathbf{X}_i) + \tilde{L}_2^{(l)}\left(\sum_{i \in [m]} \mathbf{X}_i\right), \quad (9)$$

where $\tilde{L}_1^{(l)}, \tilde{L}_2^{(l)} : \mathbb{R}^{n \times n \times d_{l-1}} \to \mathbb{R}^{n \times n \times d_l}$ are two linear functions equivariant to the graph symmetry modeled by S_n. This decomposition is significant as the design space of equivariant linear layers for graphs has been fully characterized in Maron et al. (2019b), known as 2-IGN. We thus call our model (Equations (8) and (9)) Spectral IGN.

The central question we would like to study is: what is the expressive power of Spectral IGN? Surprisingly, we have the following main result:

Theorem 6.1. *The expressive power of Spectral IGN is bounded by EPWL. Moreover, with sufficient layers and proper network parameters, Spectral IGN is as expressive as EPWL in distinguishing non-isomorphic graphs.*

The proof of Theorem 6.1 is given in Appendix A.5. It offers an interesting and alternative view for theoretically understanding the EPNN designing framework and justifying its expressiveness. Moreover, the connection to Spectral IGN allows us to bridge EPNN with important architectural variants as we will discuss in the next subsection.

Remark 6.2. We remark that there is a variant of Spectral IGN, where the pooling layer g is decomposed into two pooling layers via $g = g^{(2)} \circ \phi \circ g^{(1)}$ with $g^{(1)} : \mathbb{R}^{m \times n \times n \times d_K} \to \mathbb{R}^{n \times n \times d_K}$ and $g^{(2)} : \mathbb{R}^{n \times n \times d_K} \to \mathbb{R}^{d_K}$ pooling layers for symmetry groups S_m and S_n, respectively. This variant has the same expressive power and Theorem 6.1 still holds.

6.2. Siamese IGN

Let us consider an interesting variant of Spectral IGN, dubbed Siamese IGN (Maron et al., 2020), where the network processes each eigenspace $\mathbf{P}_i$ ($i \in [m]$) *independently* using a "Siamese" 2-IGN without aggregating over different eigenspaces. This corresponds to replacing Equation (9) by $[L^{(l)}(\mathbf{X})]_i = \tilde{L}^{(l)}(\mathbf{X}_i)$ with $\tilde{L}^{(l)} : \mathbb{R}^{n \times n \times d_{l-1}} \to \mathbb{R}^{n \times n \times d_l}$ a 2-IGN layer. Siamese IGN is interesting because it is a special type of Spectral IGN that is invariant to a *strictly larger* group formed by the wreath product $S_m \wr S_n$ (see Maron et al., 2020; Wang et al., 2020). Moreover, it is closely related to a prior architecture called BasisNet (Lim et al., 2023, see Appendix A.6 for a detailed description), as BasisNet also processes each eigenspace independently without interaction. To elaborate on this connection, consider a variant of Siamese IGN, dubbed Weak Spectral IGN, which is obtained from Siamese IGN by decomposing the pooling layer according to Remark 6.2. Note that unlike Siamese IGN, Weak Spectral IGN is no longer invariant to group $S_m \wr S_n$. We have the following result:

Proposition 6.3. *Let the top graph encoder used in BasisNet be a 1-layer message-passing GNN. Then, the expressive power of BasisNet is bounded by Weak Spectral IGN.*

On the other hand, a more fundamental question lies in the relation between Siamese IGN, Weak Spectral IGN, and Spectral IGN. Our main result states that there are strict expressivity gaps between them:

Theorem 6.4. *Siamese IGN is strictly less expressive than Weak Spectral IGN. Moreover, Weak Spectral IGN is strictly less expressive than Spectral IGN.*

Discussions with BasisNet. (i) Combined with previous results, one can prove that *EPNN is strictly more expressive than BasisNet*[2]. This result is particularly striking, as EPNN only stores node representations while BasisNet stores a representation for each 3-tuple $(i, u, v) \in [m] \times V_G \times V_G$, whose memory complexity scales like $O(|V_G|^3)$. (ii) Combined with Corollary 4.5, we conclude that the expressive power of BasisNet is also strictly bounded by 3-WL.

Discussions with SPE. We next turn to the SPE architecture (Huang et al., 2024). Surprisingly, while SPE was originally designed to improve the stability and generalization of spectral invariant GNNs (see Section 2.1), we found the soft aggregation across different eigenspaces simultaneously enhances the network's *expressive power*. Indeed, we have:

Proposition 6.5. *When 2-IGN is used to generate node features in SPE and the top graph encoder is a message-passing GNN, the expressive power of the whole SPE architecture is as expressive as Spectral IGN.*

---

[2]This holds when using multi-layer message-passing graph encoder by following the construction in Theorem 6.4.

Proposition 6.5 theoretically justifies the design of SPE. Combined with previous results, we conclude that SPE is strictly more expressive than BasisNet when using the same 2-IGN backbone, while being strictly bounded by 3-WL.

Delving more into the gap. We remark that the gap between Siamese IGN and Spectral IGN is not just theoretical; it also reveals significant limitations of the siamese design in practical aspects. Specifically, we identify that both Siamese IGN and Weak Spectral IGN *cannot* fully encode any graph distance listed in Section 5 (even the basic SPD), as stated in the following theorem:

Theorem 6.6. *For any distance listed in Section 5, there exist two non-isomorphic graphs which GD-WL can distinguish but Weak Spectral IGN (applied to matrix $\hat{L}$) cannot.*

On the other hand, we have proved that EPNN applied to matrix $\hat{L}$ is more powerful than GD-WL. This contrast reveals the crucial role of allowing interaction between eigenspaces for enhancing model's expressiveness.

6.3. Extending to higher-order spectral invariant GNNs

The DSS framework presented in Section 6.1 is quite general. In principle, any S_n-equivariant graph layer E can be used to build a GNN model f invariant to $S_m \times S_n$. This can be achieved by making $\tilde{L}_1^{(l)}, \tilde{L}_2^{(l)}$ in Equation (9) two instantiations of E. In this subsection, we will study higher-order spectral invariant GNNs where the used graph encoders are beyond 2-IGN. We consider two standard settings for choosing highly expressive graph encoders: the k-IGN and the k-order Folklore GNN (Maron et al., 2019b;a; Azizian et al., 2021). We call the resulting models Spectral k-IGN and Spectral k-FGNN, respectively. Unfortunately, our results are negative for all of these higher-order spectral invariant GNNs:

Proposition 6.7. *For all $k > 2$, Spectral k-IGN is as expressive as k-WL. Similarly, for all $k \geq 2$, Spectral k-FGNN is as expressive as k-FWL.*

We give a proof in Appendix A.7. Combined with the results that k-IGN is already as expressive as k-WL and k-FGNN is already as expressive as k-FWL (Maron et al., 2019a; Azizian et al., 2021; Geerts & Reutter, 2022), we conclude that commonly-used spectral information does not help when combined with highly powerful GNN designs.

Discussions on higher-order spectral features. The above negative result further inspires us to think about the following question: is it still possible to use spectral information to enhance the expressive power of higher-order GNNs? Here, we offer some possible directions towards this goal. The crux here is to use *higher-order* spectral features. Specifically, all the spectral information considered in previous sections (e.g., distance or projection matrices) is at most

2-dimensional. Can we generalize these spectral features into multi-dimensional tensors? This is indeed possible: for example, a simple approach is to use symmetric powers of a graph (also called the *token graph*), which has been widely studied in literature (Audenaert et al., 2007; Alzaga et al., 2010; Barghi & Ponomarenko, 2009; Fabila-Monroy et al., 2012). The k-th symmetric power of graph G, denoted by $G^{\{k\}}$, is the graph formed by vertex set $V_{G^{\{k\}}} := \{S \subset V_G : |S| = k\}$ and edge set $E_{G^{\{k\}}} := \{\{S_1, S_2\} : S_1, S_2 \in V_{G^{\{k\}}}, S_1 \triangle S_2 \in E_G\}$. Here, each element in $V_{G^{\{k\}}}$ is a multiset of cardinality k, and two multisets are connected if their symmetric difference is an edge. In this way, one can easily define higher-order spectral information $(\lambda_i, \mathbf{P}_i) \in \mathbb{R} \times \mathbb{R}^{n^{2k}}$ based on $\mathcal{P}_{G^{\{k\}}}^M = \{(\lambda_i, \mathbf{P}_i)\}_{i=1}^m$ (the eigenspace projection invariant associated with the k-th token graph), e.g., by setting $\mathbf{P}_i(u_1, \cdots, u_k, v_1, \cdots, v_k) = \mathbf{P}_i(\{\!\{u_1, \cdots, u_k\}\!\}, \{\!\{v_1, \cdots, v_k\}\!\})$. The higher-order spectral information can then serve as initial features of any higher-order GNN that computes representations for each vertex tuple, such as $2k$-IGN.

Several works have pointed out the strong expressive power of higher-order spectral features. Audenaert et al. (2007) verified that the spectra of the 3rd symmetric power are already not less expressive than 3-WL. Moreover, Alzaga et al. (2010); Barghi & Ponomarenko (2009) upper bounds the expressive power of the spectra of the k-th symmetric power by $2k$-FWL. These results imply that using higher-order projection tensors is a promising approach to further boosting the expressive power of higher-order GNNs. We leave the corresponding architectural design and expressiveness analysis as an open direction for future study.

7. Experiments

In this section, we empirically evaluate the expressive power of various GNN architectures studied in this paper. We adopt the BREC benchmark (Wang & Zhang, 2023), a comprehensive dataset for comparing the expressive power of GNNs. We focus on the following GNNs that are closely related to this paper: (i) Graphormer (Ying et al., 2021) (a distance-based GNN that uses SPD, see Section 5); (ii) NGNN (Zhang & Li, 2021) (a variant of subgraph GNN, see Section 4.1); (ii) ESAN (Bevilacqua et al., 2022) (an advanced subgraph GNN that adds cross-graph aggregations, see Section 4.1); (iv) PPGN (Maron et al., 2019a) (a higher-order GNN, see Section 6.3); (v) EPNN (this paper). We follow the same setup as in Wang & Zhang (2023) in both training and evaluation. For all baseline GNNs, the reported numbers are directly borrowed from Wang & Zhang (2023); For EPNN, we run the model 10 times with different seeds and report the average performance[3].

---

[3]Our code can be found in the following github repo: https://github.com/LingxiaoShawn/EPNN-Experiments

Table 1. Empirical performance of different GNNs on BREC.

Model	WL class	Basic	Reg	Ext	CFI	Total
Graphormer	SPD-WL	26.7	10.0	41.0	10.0	19.8
NGNN	SWL	98.3	34.3	59.0	0	41.5
ESAN	GSWL	96.7	34.3	100.0	15.0	55.2
PPGN	3-WL	100.0	35.7	100.0	23.0	58.2
EPNN	EPWL	100.0	35.7	100.0	5.0	53.8

The results are presented in Table 1. From these results, one can see that the empirical performance of EPNN matches its theoretical expressivity in our established hierarchy. Concretely, EPNN performs much better than Graphormer (SPD-WL) and NGNN (SWL), while underperforming ESAN (GS-WL) and PPGN (3-WL).

8. Conclusion

This paper investigates the expressive power of spectral invariant GNNs and related models. It establishes an expressiveness hierarchy between current models using a unifying framework we propose. We also draw a surprising connection to a recently proposed class of highly expressive GNNs, demonstrating that specific instances from this class (e.g., PSWL) upper bound all spectral invariant GNNs. This implies spectral invariant GNNs are strictly less expressive than the 3-WL test. Furthermore, we show spectral projection features and spectral distances do not provide additional expressivity benefits when combined with more powerful high-order architectures. We give a graphical illustration of all of these results in Figure 1.

Open questions. There are still several promising directions that are not fully explored in this paper. First, an interesting question lies in how the choice of graph matrix M affects the expressive power of spectral invariant GNNs. We suspect that using (normalized) Laplacian matrix can be more beneficial than using the adjacency matrix, and the former is *strictly* more expressive. We have given implicit evidence showing that EPNN with matrix $\hat{L}$ can encode all distances studied in this paper; however, we were unable to demonstrate the same result for other graph matrices. Besides, another important open question is the expressive power of higher-order spectral features, such as the one obtained by using token graphs. It is still unknown about a tight lower/upper bound of their expressive power in relation to higher-order WL tests. Moreover, investigating the *refinements* over higher-order spectral features and the corresponding GNNs could be a fantastic open direction.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgements

References

Abboud, R., Dimitrov, R., and Ceylan, I. I. Shortest path networks for graph property prediction. In *The First Learning on Graphs Conference*, 2022.

Alzaga, A., Iglesias, R., and Pignol, R. Spectra of symmetric powers of graphs and the weisfeiler–lehman refinements. *Journal of Combinatorial Theory, Series B*, 100(6):671–682, 2010.

Arnaiz-Rodríguez, A., Begga, A., Escolano, F., and Oliver, N. M. Diffwire: Inductive graph rewiring via the lovász bound. In *The First Learning on Graphs Conference*, 2022.

Audenaert, K., Godsil, C., Royle, G., and Rudolph, T. Symmetric squares of graphs. *Journal of Combinatorial Theory, Series B*, 97(1):74–90, 2007. ISSN 0095-8956.

Azizian, W. et al. Expressive power of invariant and equivariant graph neural networks. In *International Conference on Learning Representations*, 2021.

Balcilar, M., Renton, G., Héroux, P., Gaüzère, B., Adam, S., and Honeine, P. Analyzing the expressive power of graph neural networks in a spectral perspective. In *International Conference on Learning Representations*, 2021.

Barghi, A. R. and Ponomarenko, I. Non-isomorphic graphs with cospectral symmetric powers. *the electronic journal of combinatorics*, pp. R120–R120, 2009.

Bevilacqua, B., Frasca, F., Lim, D., Srinivasan, B., Cai, C., Balamurugan, G., Bronstein, M. M., and Maron, H. Equivariant subgraph aggregation networks. In *International Conference on Learning Representations*, 2022.

Bevilacqua, B., Eliasof, M., Meirom, E., Ribeiro, B., and Maron, H. Efficient subgraph gnns by learning effective selection policies. *arXiv preprint arXiv:2310.20082*, 2023.

Brouwer, A. E. and Haemers, W. H. *Spectra of graphs*. Springer Science & Business Media, 2011.

Cai, J.-Y., Fürer, M., and Immerman, N. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.

Coifman, R. R. and Lafon, S. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

Cotta, L., Morris, C., and Ribeiro, B. Reconstruction for powerful graph representations. *Advances in Neural Information Processing Systems*, 34:1713–1726, 2021.

Dwivedi, V. P. and Bresson, X. A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

Dwivedi, V. P., Joshi, C. K., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.

Dwivedi, V. P., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022.

Fabila-Monroy, R., Flores-Peñaloza, D., Huemer, C., Hurtado, F., Urrutia, J., and Wood, D. R. Token graphs. *Graphs and Combinatorics*, 2012.

Feldman, O., Boyarski, A., Feldman, S., Kogan, D., Mendelson, A., and Baskin, C. Weisfeiler and leman go infinite: Spectral and combinatorial pre-colorings. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Feng, J., Chen, Y., Li, F., Sarkar, A., and Zhang, M. How powerful are k-hop message passing graph neural networks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 4776–4790, 2022.

Frasca, F., Bevilacqua, B., Bronstein, M. M., and Maron, H. Understanding and extending subgraph gnns by rethinking their symmetries. In *Advances in Neural Information Processing Systems*, 2022.

Fürer, M. Graph isomorphism testing without numerics for graphs of bounded eigenvalue multiplicity. In *Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms*, pp. 624–631, 1995.

Fürer, M. Weisfeiler-lehman refinement requires at least a linear number of iterations. In *International Colloquium on Automata, Languages, and Programming*, pp. 322–333. Springer, 2001.

Fürer, M. On the power of combinatorial and spectral invariants. *Linear algebra and its applications*, 432(9):2373–2380, 2010.

Geerts, F. and Reuter, J. L. Expressiveness and approximation properties of graph neural networks. In *International Conference on Learning Representations*, 2022.

Grohe, M. *Descriptive complexity, canonisation, and definable graph structure theory*, volume 47. Cambridge University Press, 2017.

Huang, Y., Lu, W., Robinson, J., Yang, Y., Zhang, M., Jegelka, S., and Li, P. On the stability of expressive positional encodings for graph neural networks. In *International Conference on Learning Representations*, 2024.

Kim, J., Nguyen, D., Min, S., Cho, S., Lee, M., Lee, H., and Hong, S. Pure transformers are powerful graph learners. In *Advances in Neural Information Processing Systems*, volume 35, pp. 14582–14595, 2022.

Klein, D. J. and Randic, M. Resistance distance. *Journal of Mathematical Chemistry*, 12:81–95, 1993.

Kong, L., Feng, J., Liu, H., Tao, D., Chen, Y., and Zhang, M. Mag-gnn: Reinforcement learning boosted graph neural network. *arXiv preprint arXiv:2310.19142*, 2023.

Kreuzer, D., Beaini, D., Hamilton, W., Létourneau, V., and Tossou, P. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021.

Li, P., Wang, Y., Wang, H., and Leskovec, J. Distance encoding: design provably more powerful neural networks for graph representation learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 4465–4478, 2020.

Lim, D., Robinson, J. D., Zhao, L., Smidt, T., Sra, S., Maron, H., and Jegelka, S. Sign and basis invariant networks for spectral graph representation learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Lipman, Y., Rustamov, R. M., and Funkhouser, T. A. Biharmonic distance. *ACM Transactions on Graphics (TOG)*, 29(3):1–11, 2010.

Ma, G., Wang, Y., and Wang, Y. Laplacian canonization: A minimalist approach to sign and basis invariant spectral embedding. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a.

Ma, L., Lin, C., Lim, D., Romero-Soriano, A., Dokania, P. K., Coates, M., Torr, P., and Lim, S.-N. Graph inductive biases in transformers without message passing. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 23321–23337. PMLR, 2023b.

Maron, H., Ben-Hamu, H., Serviansky, H., and Lipman, Y. Provably powerful graph networks. In *Advances in neural information processing systems*, volume 32, 2019a.

Maron, H., Ben-Hamu, H., Shamir, N., and Lipman, Y. Invariant and equivariant graph networks. In *International Conference on Learning Representations*, 2019b.

Maron, H., Litany, O., Chechik, G., and Fetaya, E. On learning sets of symmetric elements. In *International conference on machine learning*, pp. 6734–6744. PMLR, 2020.

Maskey, S., Parviz, A., Thiessen, M., Stärk, H., Sadikaj, Y., and Maron, H. Generalized laplacian positional encoding for graph representation learning. In *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations*, 2022.

Mialon, G., Chen, D., Selosse, M., and Mairal, J. Graphit: Encoding graph structure in transformers. *arXiv preprint arXiv:2106.05667*, 2021.

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 4602–4609, 2019.

Morris, C., Rattan, G., and Mutzel, P. Weisfeiler and leman go sparse: Towards scalable higher-order graph embeddings. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21824–21840, 2020.

Morris, C., Lipman, Y., Maron, H., Rieck, B., Kriege, N. M., Grohe, M., Fey, M., and Borgwardt, K. Weisfeiler and leman go machine learning: The story so far. *arXiv preprint arXiv:2112.09992*, 2021.

Qian, C., Rattan, G., Geerts, F., Niepert, M., and Morris, C. Ordered subgraph aggregation networks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 21030–21045, 2022.

Rampasek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*, 2022.

Rattan, G. and Seppelt, T. Weisfeiler-leman and graph spectra. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2268–2285. SIAM, 2023.

Velingker, A., Sinop, A. K., Ktena, I., Veličković, P., and Gollapudi, S. Affinity-aware graph networks. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Wang, H., Yin, H., Zhang, M., and Li, P. Equivariant and stable positional encoding for more powerful graph neural networks. In *International Conference on Learning Representations*, 2022.

Wang, R., Albooyeh, M., and Ravanbakhsh, S. Equivariant maps for hierarchical structures. In *Proceedings of the*

34th International Conference on Neural Information Processing Systems, pp. 13806–13817, 2020.

Wang, X. and Zhang, M. How powerful are spectral graph neural networks. In *International Conference on Machine Learning*, pp. 23341–23362. PMLR, 2022.

Wang, Y. and Zhang, M. Towards better evaluation of gnn expressiveness with brec dataset. *arXiv preprint arXiv:2304.07702*, 2023.

Wei, Y., Li, R.-h., and Yang, W. Biharmonic distance of graphs. *arXiv preprint arXiv:2110.02656*, 2021.

Weisfeiler, B. and Lehman, A. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series*, 2(9):12–16, 1968.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T.-Y. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34, 2021.

You, J., Gomes-Selman, J. M., Ying, R., and Leskovec, J. Identity-aware graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 10737–10745, 2021.

Zhang, B., Feng, G., Du, Y., He, D., and Wang, L. A complete expressiveness hierarchy for subgraph gnns via subgraph weisfeiler-lehman tests. *arXiv preprint arXiv:2302.07090*, 2023a.

Zhang, B., Luo, S., Wang, L., and He, D. Rethinking the expressive power of gnns via graph biconnectivity. In *The Eleventh International Conference on Learning Representations*, 2023b.

Zhang, B., Gai, J., Du, Y., Ye, Q., He, D., and Wang, L. Beyond weisfeiler-lehman: A quantitative framework for GNN expressiveness. In *The Twelfth International Conference on Learning Representations*, 2024.

Zhang, M. and Li, P. Nested graph neural networks. *Advances in Neural Information Processing Systems*, 34: 15734–15747, 2021.

Zhao, L., Jin, W., Akoglu, L., and Shah, N. From stars to subgraphs: Uplifting any gnn with local structure awareness. In *International Conference on Learning Representations*, 2022.

A. Proofs

This section presents all the missing proofs in this paper. First, Appendix A.1 defines basic concepts and introduces our proof technique that will be frequently used in subsequent analysis. Then, Appendix A.2 gives several basic results and proves Proposition 4.2. The formal proofs of our main theorems, including Theorems 4.3, 5.1 and 6.1 and Proposition 6.3, are presented in Appendices A.3 to A.6, respectively. Discussion with other architectures, such as GRIT, PEG, Spectral PPGN, and Spectral k-IGN are presented in Appendix A.7. Finally, Appendix A.8 reveals the gaps between each pair of architectures in our paper, leading to the proofs of Theorems 6.4 and 6.6.

A.1. Preliminary

We first introduce some basic terminologies and concepts for general color refinement algorithms.

Color mapping. Any graph invariant over $\mathcal{G}_k$ is called a k-dimensional color mapping. We use $\mathcal{M}_k$ to denote the family of all k-dimensional color mappings. For a color mapping $\chi \in \mathcal{M}_k$, our interest lies not in the specific values of the function (say $\chi_G(\boldsymbol{u})$ for some $\boldsymbol{u} \in V_G^k$), but in the equivalence relations among different values. Formally, each color mapping $\chi \in \mathcal{M}_k$ defines an equivalence relation $\overset{\chi}{\sim}$ between rooted graphs $G^{\boldsymbol{u}}$, $H^{\boldsymbol{v}}$ marking k vertices, where $G^{\boldsymbol{u}} \overset{\chi}{\sim} H^{\boldsymbol{v}}$ iff $\chi_G(\boldsymbol{u}) = \chi_H(\boldsymbol{v})$. For any graph $G \in \mathcal{G}$, the equivalence relation $\overset{\chi}{\sim}$ induces a partition $Q_G(\chi)$ over the set V_G^k.

Given two color mappings $\chi_1, \chi_2 \in \mathcal{M}_k$, we say χ_1 is equivalent to χ_2, denoted as $\chi_1 \equiv \chi_2$, if $G^{\boldsymbol{u}} \overset{\chi_1}{\sim} H^{\boldsymbol{v}} \iff G^{\boldsymbol{u}} \overset{\chi_2}{\sim} H^{\boldsymbol{v}}$ for all graphs $G, H \in \mathcal{G}$ and vertices $\boldsymbol{u} \in V_G^k$, $\boldsymbol{v} \in V_H^k$. One can see that "$\equiv$" forms an equivalence relation over $\mathcal{M}_k$. We say χ_1 is *finer* than χ_2, denoted as $\chi_1 \preceq \chi_2$, if $G^{\boldsymbol{u}} \overset{\chi_1}{\sim} H^{\boldsymbol{v}} \implies G^{\boldsymbol{u}} \overset{\chi_2}{\sim} H^{\boldsymbol{v}}$ for all graphs G, H and vertices $\boldsymbol{u} \in V_G^k$, $\boldsymbol{v} \in V_H^k$. One can see that "$\preceq$" forms a partial relation on $\mathcal{M}_k$. We say χ_1 is *strictly finer* than χ_2, denoted as $\chi_1 \prec \chi_2$, if $\chi_1 \preceq \chi_2$ and $\chi_1 \not\equiv \chi_2$.

Color refinement. A function $T : \mathcal{M}_k \to \mathcal{M}_{k'}$ that maps from one color mapping to another is called a color transformation. Throughout this paper, we assume that all color transformations are *order-preserving*, i.e., for all $\chi_1, \chi_2 \in \mathcal{M}_k$, $T(\chi_1) \preceq T(\chi_2)$ if $\chi_1 \preceq \chi_2$. An order-preserving color transformation $T : \mathcal{M}_k \to \mathcal{M}_k$ is further called a *color refinement* if $T(\chi) \preceq \chi$ for all $\chi \in \mathcal{M}_k$. For any color refinement T, we denote by T^t the t-th function power of T, i.e., the function composition $T \circ \cdots \circ T$ with t occurrences of T. Note that if T is a color refinement, so is T^t for all $t \geq 0$.

Given two color transformations $T_1, T_2 : \mathcal{M}_k \to \mathcal{M}_{k'}$, we say T_1 is as expressive as T_2, denoted by $T_1 \equiv T_2$, if $T_1(\chi) \equiv T_2(\chi)$ for all $\chi \in \mathcal{M}_k$. We say T_1 is more expressive than T_2, denoted by $T_1 \preceq T_2$, if $T_1(\chi) \preceq T_2(\chi)$ for all $\chi \in \mathcal{M}_k$. We say T_1 is strictly more expressive than T_2, denoted by $T_1 \prec T_2$, if T_1 is more expressive than T_2 and not as expressive as T_2. As will be clear in our subsequent proofs, the expressive power of GNNs is compared through an examination of their color transformations.

For any color refinement $T : \mathcal{M}_k \to \mathcal{M}_k$, we define the corresponding stable refinement $T^\infty : \mathcal{M}_k \to \mathcal{M}_k$ as follows. For any $\chi \in \mathcal{M}_k$, define the color mapping $T^\infty(\chi)$ such that $G^{\boldsymbol{u}} \overset{T^\infty(\chi)}{\sim} H^{\boldsymbol{v}}$ iff $G^{\boldsymbol{u}} \overset{T^t(\chi)}{\sim} H^{\boldsymbol{v}}$ where $t \geq 0$ is the minimum integer satisfying $Q_G(T^t(\chi)) = Q_G(T^{t+1}(\chi))$ and $Q_H(T^t(\chi)) = Q_H(T^{t+1}(\chi))$. Note that T^∞ is well-defined since t always exists (one can see that $Q_G(T^t(\chi)) = Q_G(T^{t+1}(\chi))$ and $Q_H(T^t(\chi)) = Q_H(T^{t+1}(\chi))$ holds for all $t \geq \max(|V_G|^k, |V_H|^k)$ when T is a color refinement), and it is easy to see that T^∞ is a color refinement. We call T^∞ stable because $(T \circ T^\infty)(\chi) \equiv T^\infty(\chi)$ holds for all $\chi \in \mathcal{M}_k$, i.e., $T \circ T^\infty \equiv T^\infty$.

A color refinement algorithm A is formed by the composition of a stable refinement $T^\infty : \mathcal{M}_k \to \mathcal{M}_k$ with a color transformation $U : \mathcal{M}_k \to \mathcal{M}_0$, called the *pooling* transformation. It can be formally written as $A := U \circ T^\infty$. Below, we will derive several useful properties for general color refinement algorithms. These properties will be frequently used to compare the expressive power of different algorithms.

Proposition A.1. *Let $T_1, T_2 : \mathcal{M}_{k_1} \to \mathcal{M}_{k_2}$, $U_1, U_2 : \mathcal{M}_{k_2} \to \mathcal{M}_{k_3}$ be color transformations. If $T_1 \preceq T_2$ and $U_1 \preceq U_2$, then $U_1 \circ T_1 \preceq U_2 \circ T_2$.*

Proof. Since $T_1 \preceq T_2$, $T_1(\chi) \preceq T_2(\chi)$ holds for all $\chi \in \mathcal{M}_{k_1}$. Since U_2 is order-preserving, $U_2(T_1(\chi)) \preceq U_2(T_2(\chi))$ holds for all $\chi \in \mathcal{M}_{k_1}$. Finally, since $U_1 \preceq U_2$, $U_1(T_1(\chi)) \preceq U_2(T_1(\chi))$ holds for all $\chi \in \mathcal{M}_{k_1}$. Combining the above inequalities yields the desired result. $\square$

Proposition A.2. *Let $T_1 : \mathcal{M}_{k_1} \to \mathcal{M}_{k_1}$ and $T_2 : \mathcal{M}_{k_2} \to \mathcal{M}_{k_2}$ be color refinements, and let $U_1 : \mathcal{M}_{k_0} \to \mathcal{M}_{k_1}$ and $U_2 : \mathcal{M}_{k_1} \to \mathcal{M}_{k_2}$ be color transformations. If $T_2 \circ U_2 \circ T_1^\infty \circ U_1 \equiv U_2 \circ T_1^\infty \circ U_1$, then $U_2 \circ T_1^\infty \circ U_1 \preceq T_2^\infty \circ U_2 \circ U_1$.*

Proof. If $T_2 \circ U_2 \circ T_1^\infty \circ U_1 \equiv U_2 \circ T_1^\infty \circ U_1$, then by definition of stable refinement, $T_2^\infty \circ U_2 \circ T_1^\infty \circ U_1 \equiv U_2 \circ T_1^\infty \circ U_1$. Since T_1^∞ is a refinement, $U_2 \circ T_1^\infty \circ U_1 \preceq T_2^\infty \circ U_2 \circ U_1$ according to Proposition A.1. We thus obtain the desired result. $\square$

Corollary A.3. *Let $T_1, T_2 : \mathcal{M}_k \to \mathcal{M}_k$ be color refinements. Then, $T_1 \preceq T_2$ implies that $T_2 \circ T_1^\infty \equiv T_1^\infty$.*

Proof. Since $T_1 \preceq T_2$, $T_1 \circ T_1^\infty \preceq T_2 \circ T_1^\infty$ by Proposition A.1. Namely, $T_1^\infty \preceq T_2 \circ T_1^\infty$. On the other hand, $T_2 \circ T_1^\infty \preceq T_1^\infty$ since T_2 is a color refinement. Combined the two directions yields the desired result. $\square$

The above two propositions will play a crucial role in our subsequent proofs. Below, we give a simple example to illustrate how these results can be used to give a proof that a GNN model M_1 is more expressive than another model M_2. Suppose $T_1, T_2 : \mathcal{M}_k \to \mathcal{M}_k$ are color refinements corresponding to one GNN layer of M_1 and M_2, respectively, and let $U : \mathcal{M}_k \to \mathcal{M}_0$ be the color transformation corresponding to the final pooling layer in M_1 and M_2. Then, the color refinement algorithms associated with M_1 and M_2 can be represented by $U \circ T_1^\infty$ and $U \circ T_2^\infty$, respectively. Concretely, denote by χ^0 the initial color mapping in the two algorithms, e.g., the constant mapping where $\chi_G^0(\boldsymbol{u})$ is the same for all $G^{\boldsymbol{u}} \in \mathcal{G}_k$. It follows that the graph representation of a graph G computed by the two algorithms is $[U(T_1(\chi^0))](G)$ and $[U(T_2(\chi^0))](G)$, respectively. Then, The statement "M_1 is more expressive than M_2" means that $[U(T_1(\chi^0))](G) = [U(T_1(\chi^0))](H) \implies [U(T_2(\chi^0))](G) = [U(T_2(\chi^0))](H)$ for all graphs $G, H \in \mathcal{G}$.

To prove that M_1 is more expressive than M_2, it suffices to prove that $T_2 \circ T_1^\infty$ is as expressive as T_1^∞. Indeed, this is actually a simple consequence of Propositions A.1 and A.2. If $T_2 \circ T_1^\infty$ is as expressive as T_1^∞, then T_1^∞ is more expressive than T_2^∞ (Proposition A.2), and thus $U \circ T_1^\infty$ is more expressive than $U \circ T_2^\infty$ (Proposition A.1), yielding the desired result.

A.2. Basic results

This subsection proves several basic results for EPWL. We begin by proving that EPWL is strictly more expressive than 1-WL (Proposition 4.2). We will first restate Proposition 4.2 using the color refinement terminologies defined in Appendix A.1. We need the following color transformations:

- **EPWL color refinement.** Define $T_{\mathsf{EP},\boldsymbol{M}} : \mathcal{M}_1 \to \mathcal{M}_1$ such that for any color mapping $\chi \in \mathcal{M}_1$ and rooted graph G^u,

$$[T_{\mathsf{EP},\boldsymbol{M}}(\chi)]_G(u) = \mathsf{hash}(\chi_G(u), \{\!\{(\chi_G(v), \mathcal{P}_G^{\boldsymbol{M}}(u,v)) : v \in V_G\}\!\}). \tag{10}$$

- **1-WL color refinement.** Define $T_{\mathsf{WL}} : \mathcal{M}_1 \to \mathcal{M}_1$ such that for any color mapping $\chi \in \mathcal{M}_1$ and rooted graph G^u,

$$[T_{\mathsf{WL}}(\chi)]_G(u) = \mathsf{hash}(\chi_G(u), \{\!\{(\chi_G(v), \mathsf{atp}_G(u,v)) : v \in V_G\}\!\}). \tag{11}$$

- **Global pooling.** Define $T_{\mathsf{GP}} : \mathcal{M}_1 \to \mathcal{M}_0$ such that for any color mapping $\chi \in \mathcal{M}_1$ and graph G,

$$[T_{\mathsf{GP}}(\chi)](G) = \mathsf{hash}(\{\!\{\chi_G(u) : u \in V_G\}\!\}). \tag{12}$$

Equipped with the above color transformations, Proposition 4.2 is equivalent to the following:

Proposition A.4. *For any graph matrix $\boldsymbol{M} \in \{\boldsymbol{A}, \boldsymbol{L}, \hat{\boldsymbol{L}}\}$, $T_{\mathsf{GP}} \circ T_{\mathsf{EP},\boldsymbol{M}}^\infty \preceq T_{\mathsf{GP}} \circ T_{\mathsf{WL}}^\infty$.*

Based on Propositions A.1 and A.2, it suffices to prove that $T_{\mathsf{EP},\boldsymbol{M}} \preceq T_{\mathsf{WL}}$.

Lemma A.5. *For any graph matrix $\boldsymbol{M} \in \{\boldsymbol{A}, \boldsymbol{L}, \hat{\boldsymbol{L}}\}$, $\mathcal{P}^{\boldsymbol{M}} \preceq \mathsf{atp}$. Here, the atomic type operator $\mathsf{atp} \in \mathcal{M}_2$ is regarded as a 2-dimensional color mapping. This readily implies that $T_{\mathsf{EP},\boldsymbol{M}} \preceq T_{\mathsf{WL}}$.*

Proof. It suffices to prove that, for any two graphs $G, H \in \mathcal{G}$ and vertices $u, v \in V_G$, $x, y \in V_H$, if $\mathcal{P}_G^{\boldsymbol{M}}(u,v) = \mathcal{P}_H^{\boldsymbol{M}}(x,y)$, then (a) $u = v \iff x = y$; (b) $\{u, v\} \in E_G \iff \{x, y\} \in E_H$.

Item (a) simply follows from the fact that $u = v$ iff $\sum_{(\lambda, \boldsymbol{P}(u,v)) \in \mathcal{P}_G^{\boldsymbol{M}}(u,v)} \boldsymbol{P}(u,v) = 1$ (by definition of eigen-decomposition). Item (b) simply follows from the fact that $\{u, v\} \in E_G$ iff $\sum_{(\lambda, \boldsymbol{P}(u,v)) \in \mathcal{P}_G^{\boldsymbol{M}}(u,v)} \lambda \boldsymbol{P}(u,v) \neq 0$ and $u \neq v$ (which holds for all matrices $\boldsymbol{M} \in \{\boldsymbol{A}, \boldsymbol{L}, \hat{\boldsymbol{L}}\}$). $\square$

14

Strict sepatation. It is easy to give a pair of counterexample graphs G, H that are indistinguishable by 1-WL but can be distinguished via EPWL. We give such a pair of graphs in Figure 2. One can check that the two graphs has difference set of eigenvalues no matter what graph matrix $M \in \{A, L, \hat{L}\}$ is used in EPWL.
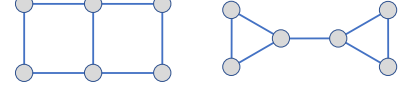

Figure 2. A pair of counterexample graphs that are indistinguishable by 1-WL but can be distinguished via EPWL.

We next show that EPWL is more expressive than spectral positional encoding using Laplacian eigenvectors. We will prove the following result:

Proposition A.6. *Given any graph matrix $M \in \{A, L, \hat{L}\}$ and any initial color mapping $\chi^0 \in \mathcal{M}$, let $\chi^1 = T_{\mathsf{EP},M}(\chi^0)$ be the EPWL color mapping after the first iteration. Then, for any graphs $G, H \in \mathcal{G}$ and vertices $u \in V_G, x \in V_H$, if $\chi_G^1(u) = \chi_H^1(v)$, then $\mathcal{P}_G^M(u, u) = \mathcal{P}_H^M(x, x)$.*

Proof. Let $G, H \in \mathcal{G}$ and vertices $u \in V_G, x \in V_H$ satisfy that $\chi_G^1(u) = \chi_H^1(v)$. Then,

$$\{\!\{(\chi_G^0(v), \mathcal{P}_G^M(u, v)) : v \in V_G\}\!\} = \{\!\{(\chi_H^0(y), \mathcal{P}_H^M(x, y)) : y \in V_H\}\!\}. \tag{13}$$

Note that for any $u, v \in V_G$ and $x, y \in V_H$, if $\mathcal{P}_G^M(u, v) = \mathcal{P}_H^M(x, y)$, then $u = v \iff x = y$ (Lemma A.5). Therefore, Equation (13) implies that $\mathcal{P}_G^M(u, u) = \mathcal{P}_H^M(x, x)$. $\square$

A.3. Proof of Theorem 4.3

This subsection aims to prove Theorem 4.3. We will first restate Theorem 4.3 using the color refinement terminologies defined in Appendix A.1. We need the following color transformations:

- **EPWL color refinement.** Define $T_{\mathsf{EP},M} : \mathcal{M}_1 \to \mathcal{M}_1$ such that for any color mapping $\chi \in \mathcal{M}_1$ and rooted graph G^u,

$$[T_{\mathsf{EP},M}(\chi)]_G(u) = \mathsf{hash}(\chi_G(u), \{\!\{(\chi_G(v), \mathcal{P}_G^M(u, v)) : v \in V_G\}\!\}). \tag{14}$$

- **SWL color refinement.** Define $T_{\mathsf{S}} : \mathcal{M}_2 \to \mathcal{M}_2$ such that for any color mapping $\chi \in \mathcal{M}_2$ and rooted graph G^{uv},

$$[T_{\mathsf{S}}(\chi)]_G(u, v) = \mathsf{hash}(\chi_G(u, v), \{\!\{(\chi_G(u, w), \mathsf{atp}_G(v, w)) : w \in V_G\}\!\}). \tag{15}$$

- **PSWL color refinement.** Define $T_{\mathsf{PS}} : \mathcal{M}_2 \to \mathcal{M}_2$ such that for any color mapping $\chi \in \mathcal{M}_2$ and rooted graph G^{uv},

$$[T_{\mathsf{PS}}(\chi)]_G(u, v) = \mathsf{hash}(\chi_G(u, v), \chi_G(v, v), \{\!\{(\chi_G(u, w), \mathsf{atp}_G(v, w)) : w \in V_G\}\!\}). \tag{16}$$

- **Global refinement.** Define $T_{\mathsf{Gu}}, T_{\mathsf{Gv}} : \mathcal{M}_2 \to \mathcal{M}_2$ such that for any color mapping $\chi \in \mathcal{M}_2$ and rooted graph G^{uv},

$$[T_{\mathsf{Gu}}(\chi)]_G(u, v) = \mathsf{hash}(\chi_G(u, v), \{\!\{\chi_G(u, w) : w \in V_G\}\!\}), \tag{17}$$

$$[T_{\mathsf{Gv}}(\chi)]_G(u, v) = \mathsf{hash}(\chi_G(u, v), \{\!\{\chi_G(w, v) : w \in V_G\}\!\}). \tag{18}$$

- **Diagonal refinement.** Define $T_{\mathsf{Du}}, T_{\mathsf{Dv}} : \mathcal{M}_2 \to \mathcal{M}_2$ such that for any color mapping $\chi \in \mathcal{M}_2$ and rooted graph G^{uv},

$$[T_{\mathsf{Du}}(\chi)]_G(u, v) = \mathsf{hash}(\chi_G(u, v), \chi_G(u, u)), \tag{19}$$

$$[T_{\mathsf{Dv}}(\chi)]_G(u, v) = \mathsf{hash}(\chi_G(u, v), \chi_G(v, v)). \tag{20}$$

- **Node marking refinement.** Define $T_{\mathsf{NM}} : \mathcal{M}_2 \to \mathcal{M}_2$ such that for any color mapping $\chi \in \mathcal{M}_2$ and rooted graph G^{uv},

$$[T_{\mathsf{NM}}(\chi)]_G(u, v) = \mathsf{hash}(\chi_G(u, v), \mathbb{I}[u = v]). \tag{21}$$

- **Lift transformation.** Define $T_{\uparrow} : \mathcal{M}_1 \to \mathcal{M}_2$ such that for any color mapping $\chi \in \mathcal{M}_1$ and rooted graph G^{uv},

$$[T_{\uparrow}(\chi)]_G(u, v) = \chi_G(v). \tag{22}$$

- **Subgraph pooling.** Define $T_{\mathsf{SP}} : \mathcal{M}_2 \to \mathcal{M}_1$ such that for any color mapping $\chi \in \mathcal{M}_2$ and rooted graph G^u,

$$[T_{\mathsf{SP}}(\chi)]_G(u) = \mathsf{hash}(\{\!\{\chi_G(u, v) : v \in V_G\}\!\}). \tag{23}$$

- **Global pooling.** Define $T_{\mathsf{GP}} : \mathcal{M}_1 \to \mathcal{M}_0$ such that for any color mapping $\chi \in \mathcal{M}_1$ and graph G,

$$[T_{\mathsf{GP}}(\chi)](G) = \mathsf{hash}(\{\!\{\chi_G(u) : u \in V_G\}\!\}). \tag{24}$$

Note that $T_{\mathsf{PS}} \preceq T_{\mathsf{S}}$, $T_{\mathsf{PS}} \preceq T_{\mathsf{Gu}}$, and $T_{\mathsf{PS}} \preceq T_{\mathsf{Dv}}$. Equipped with the above color transformations, Theorem 4.3 is equivalent to the following:

Theorem A.7. *For any graph matrix $\boldsymbol{M} \in \{\boldsymbol{A}, \boldsymbol{L}, \hat{\boldsymbol{L}}\}$, $T_{\mathsf{GP}} \circ T_{\mathsf{SP}} \circ T_{\mathsf{PS}}^\infty \circ T_{\mathsf{NM}} \circ T_\uparrow \preceq T_{\mathsf{GP}} \circ T_{\mathsf{EP},\boldsymbol{M}}^\infty$.*

We will decompose the proof into a set of lemmas. First, we leverage a recent breakthrough in graph theory established by Rattan & Seppelt (2023). We restate their result in our context as follows.

Definition A.8. Define a family of graph matrices $\mathfrak{E}$ as follows, called *equitable matrices*:

a) Base matrices: the identify matrix $\boldsymbol{I}$, all-one matrix $\boldsymbol{J}$, adjacency matrix $\boldsymbol{A}$, degree matrix $\boldsymbol{D}$ are all equitable;

b) Algebraic property: for any equitable matrices $\boldsymbol{M}_1, \boldsymbol{M}_2 \in \mathfrak{E}$, $\boldsymbol{M}_1 + \boldsymbol{M}_2, c\boldsymbol{M}_1, \boldsymbol{M}_1\boldsymbol{M}_2 \in \mathfrak{E}$, where $c \in \mathbb{C}$ can be any constant.

c) Spectral property: for any $\boldsymbol{M} \in \mathfrak{E}$ and $\lambda \in \mathbb{C}$, let $\boldsymbol{P}_\lambda^{\boldsymbol{M}}$ be the projection onto the eigenspace spanned by the eigenvectors of $\boldsymbol{M}$ with eigenvalue λ ($\boldsymbol{P}_\lambda^{\boldsymbol{M}} = \boldsymbol{O}$ if λ is not an eigenvalue of $\boldsymbol{M}$). Then, $\boldsymbol{P}_\lambda^{\boldsymbol{M}} \in \mathfrak{E}$.

Based on Definition A.8, we readily have the following proposition:

Proposition A.9. *The Laplacian matrix $\boldsymbol{L}$ and the normalized Laplacian matrix $\hat{\boldsymbol{L}}$ are equitable.*

Rattan & Seppelt (2023) proved the following main result:

Theorem A.10. *For any $\boldsymbol{M} \in \mathfrak{E}$, $T_{\mathsf{S}}^\infty(T_{\mathsf{NM}}(\chi^{\mathsf{C}})) \preceq \boldsymbol{M}$, where $\chi^{\mathsf{C}} \in \mathcal{M}_2$ is the constant color mapping.*

Corollary A.11. *For any symmetric equitable matrix $\boldsymbol{M} \in \mathfrak{E}$ and initial color mapping $\chi^0 \in \mathcal{M}_1$, $(T_{\mathsf{PS}}^\infty \circ T_{\mathsf{NM}} \circ T_\uparrow)(\chi^0) \preceq \mathcal{P}^{\boldsymbol{M}}$.*

Proof. For any symmetric equitable matrix $\boldsymbol{M} \in \mathfrak{E}$ and any $\lambda \in \mathbb{R}$, $\boldsymbol{P}_\lambda^{\boldsymbol{M}} \in \mathfrak{E}$ holds by Definition A.8(c). Therefore, Theorem A.10 implies that $T_{\mathsf{S}}^\infty(T_{\mathsf{NM}}(\chi^{\mathsf{C}})) \preceq \boldsymbol{P}_\lambda^{\boldsymbol{M}}$, i.e., $T_{\mathsf{PS}}^\infty(T_{\mathsf{NM}}(\chi^{\mathsf{C}})) \preceq \boldsymbol{P}_\lambda^{\boldsymbol{M}}$. Next, note that $\boldsymbol{P}_\lambda^{\boldsymbol{M}} \neq \boldsymbol{O}$ iff λ is an eigenvalue of $\boldsymbol{M}$. Therefore, the graph invariant $\boldsymbol{\Delta}$ over $\mathcal{G}_2$ defined by $\boldsymbol{\Delta}_G(u, v) = \mathbb{I}[\lambda$ is an eigenvalue of $\boldsymbol{M}]$ satisfies that $(T_{\mathsf{Gu}} \circ T_{\mathsf{Dv}} \circ T_{\mathsf{Gu}} \circ T_{\mathsf{PS}}^\infty \circ T_{\mathsf{NM}})(\chi^{\mathsf{C}})) \preceq \boldsymbol{\Delta}$. Since $T_{\mathsf{Gu}} \circ T_{\mathsf{Dv}} \circ T_{\mathsf{Gu}} \circ T_{\mathsf{PS}}^\infty \equiv T_{\mathsf{PS}}^\infty$ (based on the facts $T_{\mathsf{PS}} \preceq T_{\mathsf{Gu}}$ and $T_{\mathsf{PS}} \preceq T_{\mathsf{Dv}}$ and Corollary A.3), we have $T_{\mathsf{PS}}^\infty(T_{\mathsf{NM}}(\chi^{\mathsf{C}})) \preceq \boldsymbol{\Delta}$. By considering all $\lambda \in \mathbb{R}$, we obtain that $T_{\mathsf{PS}}^\infty(T_{\mathsf{NM}}(\chi^{\mathsf{C}})) \preceq \mathcal{P}^{\boldsymbol{M}}$. Finally, noting that $T_{\mathsf{PS}}^\infty \circ T_{\mathsf{NM}}$ is order-preserving and $T_\uparrow(\chi^0) \preceq \chi^{\mathsf{C}}$, we have $(T_{\mathsf{PS}}^\infty \circ T_{\mathsf{NM}} \circ T_\uparrow)(\chi^0) \preceq \mathcal{P}^{\boldsymbol{M}}$ for all $\chi^0 \in \mathcal{M}_1$. $\quad\square$

We are now ready to prove Theorem A.7.

Proof of Theorem A.7. Note that $T_{\mathsf{SP}} \circ T_{\mathsf{NM}} \circ T_\uparrow$ is a color refinement, and thus $T_{\mathsf{EP},\boldsymbol{M}}^\infty \circ T_{\mathsf{SP}} \circ T_{\mathsf{NM}} \circ T_\uparrow \preceq T_{\mathsf{EP},\boldsymbol{M}}^\infty$. To prove that $T_{\mathsf{GP}} \circ T_{\mathsf{SP}} \circ T_{\mathsf{PS}}^\infty \circ T_{\mathsf{NM}} \circ T_\uparrow \preceq T_{\mathsf{GP}} \circ T_{\mathsf{EP},\boldsymbol{M}}^\infty$, it suffices to prove that $T_{\mathsf{SP}} \circ T_{\mathsf{PS}}^\infty \circ T_{\mathsf{NM}} \preceq T_{\mathsf{EP},\boldsymbol{M}}^\infty \circ T_{\mathsf{SP}} \circ T_{\mathsf{NM}}$ according to Proposition A.1. Moreover, based on Proposition A.2, it suffices to prove that $T_{\mathsf{EP},\boldsymbol{M}} \circ T_{\mathsf{SP}} \circ T_{\mathsf{PS}}^\infty \circ T_{\mathsf{NM}} \equiv T_{\mathsf{SP}} \circ T_{\mathsf{PS}}^\infty \circ T_{\mathsf{NM}}$.

Let $\chi^0 \in \mathcal{M}_2$ be any initial color mapping and let $\chi = T_{\mathsf{PS}}^\infty(T_{\mathsf{NM}}(\chi_0))$. Pick any graphs G, H and vertices $u \in V_G, x \in V_H$ such that $[T_{\mathsf{SP}}(\chi)]_G(u) = [T_{\mathsf{SP}}(\chi)]_H(x)$, i.e.,

$$\{\!\{\chi_G(u, v) : v \in V_G\}\!\} = \{\!\{\chi_H(x, y) : y \in V_H\}\!\}. \tag{25}$$

Invoking Corollary A.11 obtains that

$$\{\!\{(\chi_G(u, v), \mathcal{P}_G^M(u, v)) : v \in V_G\}\!\} = \{\!\{(\chi_H(x, y), \mathcal{P}_H^M(x, y)) : y \in V_H\}\!\}. \tag{26}$$

Since $T_{\mathsf{PS}} \preceq T_{\mathsf{Dv}}$, based on Corollary A.3 we have $T_{\mathsf{Dv}} \circ T_{\mathsf{PS}}^\infty \equiv T_{\mathsf{PS}}^\infty$, i.e., $\chi \equiv T_{\mathsf{Dv}}(\chi)$. Therefore,

$$\{\!\{(\chi_G(v, v), \mathcal{P}_G^M(u, v)) : v \in V_G\}\!\} = \{\!\{(\chi_H(y, y), \mathcal{P}_H^M(x, y)) : y \in V_H\}\!\}. \tag{27}$$

Since $T_{\mathsf{PS}} \preceq T_{\mathsf{Gu}}$, we similarly have $\chi \equiv T_{\mathsf{Gu}}(\chi)$. Therefore,

$$\{\!\{(\{\!\{\chi_G(v, w) : w \in V_G\}\!\}, \mathcal{P}_G^M(u, v)) : v \in V_G\}\!\} = \{\!\{(\{\!\{\chi_H(y, z) : z \in V_H\}\!\}, \mathcal{P}_H^M(x, y)) : y \in V_H\}\!\}. \tag{28}$$

Combining with Equations (25) and (28), we have $T_{\mathsf{EP},\boldsymbol{M}}(T_{\mathsf{SP}}(\chi)) \equiv T_{\mathsf{SP}}(\chi)$. Finally, noting that χ^0 is arbitrary, we conclude that $T_{\mathsf{EP},\boldsymbol{M}} \circ T_{\mathsf{SP}} \circ T_{\mathsf{PS}}^\infty \circ T_{\mathsf{NM}} \equiv T_{\mathsf{SP}} \circ T_{\mathsf{PS}}^\infty \circ T_{\mathsf{NM}}$. $\quad\square$

A.4. Proof of Theorem 5.1

This subsection aims to prove Theorem 5.1. We will first restate Theorem 5.1 using the color refinement terminologies defined in Appendix A.1. Similarly to Appendix A.3, we define the following color transformations:

- **EPWL color refinement.** Define $T_{\mathsf{EP},\boldsymbol{M}} : \mathcal{M}_1 \to \mathcal{M}_1$ such that for any color mapping $\chi \in \mathcal{M}_1$ and rooted graph G^u,

$$[T_{\mathsf{EP},\boldsymbol{M}}(\chi)]_G(u) = \mathsf{hash}(\chi_G(u), \{\!\{(\chi_G(v), \mathcal{P}_G^{\boldsymbol{M}}(u,v)) : v \in V_G\}\!\}). \tag{29}$$

- **GD-WL color refinement.** Define $T_{\mathsf{GD},\boldsymbol{M}} : \mathcal{M}_1 \to \mathcal{M}_1$ such that for any color mapping $\chi \in \mathcal{M}_1$ and rooted graph G^u,

$$[T_{\mathsf{GD},\boldsymbol{M}}(\chi)]_G(u) = \mathsf{hash}(\chi_G(u), \{\!\{(\chi_G(v), \boldsymbol{M}_G(u,v)) : v \in V_G\}\!\}). \tag{30}$$

- **Global pooling.** Define $T_{\mathsf{GP}} : \mathcal{M}_1 \to \mathcal{M}_0$ such that for any color mapping $\chi \in \mathcal{M}_1$ and graph G,

$$[T_{\mathsf{GP}}(\chi)](G) = \mathsf{hash}(\{\!\{\chi_G(u) : u \in V_G\}\!\}). \tag{31}$$

Theorem 5.1 is equivalent to the following:

Theorem A.12. *For any graph distance matrix $\boldsymbol{M}$ listed in Section 5, $T_{\mathsf{GP}} \circ T_{\mathsf{EP},\hat{\boldsymbol{L}}}^\infty \preceq T_{\mathsf{GP}} \circ T_{\mathsf{GD},\boldsymbol{M}}^\infty$.*

In this subsection, let $\chi^{\mathsf{C}} \in \mathcal{M}_1$ be the constant color mapping and let $\chi = T_{\mathsf{EP},\hat{\boldsymbol{L}}}^\infty(\chi^{\mathsf{C}})$. Define a color mapping $\bar{\chi} \in \mathcal{M}_2$ such that $\bar{\chi}_G(u,v) = \left(\chi_G(u), \chi_G(v), \mathcal{P}_G^{\hat{\boldsymbol{L}}}(u,v)\right)$ for all graph G and vertices $u,v \in V_G$. The following proposition will play a central role in our subsequent proofs:

Proposition A.13. *For any graph matrix $\boldsymbol{M}$, if $\bar{\chi} \preceq \boldsymbol{M}$, then $T_{\mathsf{GP}} \circ T_{\mathsf{EP},\hat{\boldsymbol{L}}}^\infty \preceq T_{\mathsf{GP}} \circ T_{\mathsf{GD},\boldsymbol{M}}^\infty$.*

Proof. Based on Propositions A.1 and A.2, it suffices to prove that $T_{\mathsf{GD},\boldsymbol{M}} \circ T_{\mathsf{EP},\hat{\boldsymbol{L}}}^\infty \equiv T_{\mathsf{EP},\hat{\boldsymbol{L}}}^\infty$. Pick any $\chi^0 \in \mathcal{M}_1$ and let $\chi' = T_{\mathsf{EP},\hat{\boldsymbol{L}}}^\infty(\chi^0)$. Since $\chi' \equiv T_{\mathsf{EP},\hat{\boldsymbol{L}}}(\chi')$, we have

$$\chi'_G(u) = \chi'_H(y) \implies \chi'_G(u) = \chi'_H(y) \wedge \{\!\{(\chi'_G(v), \mathcal{P}_G^{\hat{\boldsymbol{L}}}(u,v)) : v \in V_G\}\!\} = \{\!\{(\chi'_H(y), \mathcal{P}_H^{\hat{\boldsymbol{L}}}(x,y)) : y \in V_H\}\!\}$$

$$\implies \{\!\{(\chi'_G(u), \chi'_G(v), \mathcal{P}_G^{\hat{\boldsymbol{L}}}(u,v)) : v \in V_G\}\!\} = \{\!\{(\chi'_H(x), \chi'_H(y), \mathcal{P}_H^{\hat{\boldsymbol{L}}}(x,y)) : y \in V_H\}\!\}$$

for all graphs $G, H \in \mathcal{G}$ and vertices $u \in V_G, x \in V_H$. Moreover, since $\chi' \preceq \chi$ and $\bar{\chi} \preceq \boldsymbol{M}$, we have

$$\chi'_G(u) = \chi'_H(y) \implies \{\!\{(\chi'_G(v), \boldsymbol{M}_G(u,v)) : v \in V_G\}\!\} = \{\!\{(\chi'_H(y), \boldsymbol{M}_H(x,y)) : y \in V_H\}\!\}$$

for all graphs $G, H \in \mathcal{G}$ and vertices $u \in V_G, x \in V_H$. Therefore, $\chi' \equiv T_{\mathsf{GD},\boldsymbol{M}}(\chi')$. Finally, by noting that $\chi' = T_{\mathsf{EP},\hat{\boldsymbol{L}}}^\infty(\chi^0)$ and χ^0 is arbitrary, we obtain that $T_{\mathsf{GD},\boldsymbol{M}} \circ T_{\mathsf{EP},\hat{\boldsymbol{L}}}^\infty \equiv T_{\mathsf{EP},\hat{\boldsymbol{L}}}^\infty$, as desired. $\qquad\square$

We next present several basic facts about EP-WL stable colors χ.

Proposition A.14. *For any graphs $G, H \in \mathcal{G}$ and vertices $u \in V_G, x \in V_H$, if $\chi_G(u) = \chi_H(x)$, then*

a) $\deg_G(u) = \deg_H(x)$;

b) $\mathcal{P}_G^{\hat{\boldsymbol{L}}}(u,u) = \mathcal{P}_H^{\hat{\boldsymbol{L}}}(x,x)$.

Proof. Since $T_{\mathsf{EP},\boldsymbol{M}} \preceq T_{\mathsf{WL}}$ (Lemma A.5), we have $\chi \preceq T_{\mathsf{WL}}^\infty(\chi^{\mathsf{C}})$. Item (a) follows from the fact that the 1-WL stable color can encode the vertex degree, i.e., $T_{\mathsf{WL}}^\infty(\chi^{\mathsf{C}}) \preceq \deg$. To prove item (b), note that $\chi \preceq \chi^1$, where χ^1 is defined in Proposition A.6. Therefore, item (b) readily follows from Proposition A.6. $\qquad\square$

Below, we will separately consider each of the distances listed in Section 5.

Lemma A.15. *Let $\boldsymbol{M}$ be the shortest path distance matrix. Then, $\bar{\chi} \preceq \boldsymbol{M}$.*

Proof. We will prove a stronger result that $\mathcal{P}^{\hat{L}} \preceq M$. Note that there is a relation between M and the normalized adjacency matrix $\hat{A} := D^{-1/2} A D^{-1/2} = I - \hat{L}$, which can be expressed as follows:

$$M_G(u, v) = \min\{i \in \mathbb{N} : \hat{A}_G^i(u, v) > 0\} \qquad \forall G \in \mathcal{G}, u, v \in V_G. \tag{32}$$

The above relation can be interpreted as follows: $\hat{A}_G^i(u, v) = \sum_{p_i(u,v)} \omega(p_i(u, v))$ where $p_i(u, v)$ ranges over all walks of length i from u to v, and $\omega(p_i(u, v)) := (\deg_G(u) \deg_G(v))^{-1/2} \left(\Pi_{j=1}^{i-1} \deg_G(w_i)\right)^{-1}$ for a walk $p_i(u, v) = (u, w_1, \cdots, w_{i-1}, v)$, which is always positive.

Based on the property of eigen-decomposition, we have

$$\hat{A}_G^i(u, v) = \sum_{(\lambda, P_G(u,v)) \in \mathcal{P}_G^{\hat{A}}} \lambda^i P_G(u, v) = \sum_{(\lambda, P_G(u,v)) \in \mathcal{P}_G^{\hat{L}}(u,v)} (1 - \lambda)^i P_G(u, v). \tag{33}$$

This implies that $M_G(u, v)$ can be purely determined by $\mathcal{P}_G^{\hat{L}}(u, v)$, i.e., $\mathcal{P}^{\hat{L}} \preceq M$. $\qquad\square$

Lemma A.16. *Let M be the resistance distance matrix. Then, $\bar{\chi} \preceq M$.*

Proof. We first consider the case of connected graphs. We leverage a celebrated result established in Klein & Randic (1993), which builds connections between resistance distance and graph Laplacian. Specifically, for any connected graph $G \in \mathcal{G}$ and vertices $u, v \in V_G$,

$$M_G(u, v) = L^\dagger(u, u) + L^\dagger(v, v) - 2L^\dagger(u, v), \tag{34}$$

where $\dagger$ denotes the matrix Moore–Penrose inversion. Substituting $L = D^{1/2} \hat{L} D^{1/2}$ into Equation (34), we obtain

$$M_G(u, v) = (\deg_G(u))^{-1} \hat{L}^\dagger(u, u) + (\deg_G(v))^{-1} \hat{L}^\dagger(v, v) - 2(\deg_G(u) \deg_G(v))^{-1/2} \hat{L}^\dagger(u, v), \tag{35}$$

Moreover, we have $\hat{L}^\dagger = \sum_{i:\lambda_i \neq 0} \lambda_i^{-1} P_i$ where $\sum_i \lambda_i P_i$ is the eigen-decomposition of $\hat{L}$. Combining all these relations, one can see that $M_G(u, v)$ is purely determined by the tuple $(\deg_G(u), \deg_G(v), \mathcal{P}_G^{\hat{L}}(u, u), \mathcal{P}_G^{\hat{L}}(v, v), \mathcal{P}_G^{\hat{L}}(u, v))$. Based on Proposition A.14, $\chi_G(u)$ determines $\deg_G(u)$ and $\mathcal{P}_G^{\hat{L}}(u, u)$, and $\chi_G(v)$ determines $\deg_G(v)$ and $\mathcal{P}_G^{\hat{L}}(v, v)$. Consequently, $\bar{\chi} \preceq M$.

We next consider the general case where the graph G is disconnected. In this case, L is a block diagonal matrix. It is straightforward to see that

$$M_G(u, v) = \begin{cases} L^\dagger(u, u) + L^\dagger(v, v) - 2L^\dagger(u, v) & \text{if } u \text{ and } v \text{ are in the same connected component,} \\ \infty & \text{otherwise.} \end{cases} \tag{36}$$

Since we have proved that $\mathcal{P}^M$ can encode the shortest path distance (Lemma A.15), $\mathcal{P}_G^{\hat{L}}(u, v)$ can encode whether u and v are in the same connected component. So we still have $\bar{\chi} \preceq M$. $\qquad\square$

Lemma A.17. *Let M be the hitting-time distance matrix. Then, $\bar{\chi} \preceq M$.*

Proof. Without loss of generality, we only consider connected graphs as in Lemma A.16. According to the definition of hitting-time distance, for any connected graph $G \in \mathcal{G}$ and vertices $u, v \in V_G$, we have the following recursive relation:

$$M_G(u, v) = \begin{cases} 0 & \text{if } u = v, \\ 1 + \frac{1}{\deg_G(u)} \sum_{w \in \mathcal{N}_G(u)} M_G(u, v) & \text{otherwise.} \end{cases} \tag{37}$$

Now define a new graph matrix

$$\tilde{H} = \mathbf{1}\mathbf{1}^\top + D^{-1} A(\tilde{H} - \text{diag}(\tilde{H})) \tag{38}$$

where $\text{diag}(\tilde{H})$ is the diagonal matrix obtained from $\tilde{H}$ by zeroing out all non-diagonal elements of $\tilde{H}$. It follows that $M = \tilde{H} - \text{diag}(\tilde{H})$.

We first calculate $\text{diag}(\tilde{H})$. Left-multiplying Equation (38) by $\mathbf{1}^\top A$ yields that

$$\mathbf{1}^\top A\tilde{H} = \mathbf{1}^\top A\mathbf{1}\mathbf{1}^\top + \mathbf{1}^\top A D^{-1} A(\tilde{H} - \text{diag}(\tilde{H})) = \mathbf{1}^\top A\mathbf{1}\mathbf{1}^\top + \mathbf{1}^\top A(\tilde{H} - \text{diag}(\tilde{H})),$$

where we use the fact that $\mathbf{1}^{\top}\boldsymbol{A}\boldsymbol{D}^{-1} = \mathbf{1}^{\top}$. Therefore, $\mathbf{1}^{\top}\boldsymbol{A}\mathrm{diag}(\tilde{\boldsymbol{H}}) = \mathbf{1}^{\top}\boldsymbol{A}\mathbf{1}\mathbf{1}^{\top}$, namely, $\tilde{\boldsymbol{H}}_G(u, u) = 2|E_G|(\deg_G(u))^{-1}$.

We next compute the full matrix $\boldsymbol{M}$. Based on Equation (38), we have

$$(\boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{A})\boldsymbol{M} = \mathbf{1}\mathbf{1}^{\top} - \mathrm{diag}(\tilde{\boldsymbol{H}}). \tag{39}$$

Left-multiplying Equation (39) by $\boldsymbol{D}$ leads to the fundamental equation

$$\boldsymbol{L}\boldsymbol{M} = \boldsymbol{D}\mathbf{1}\mathbf{1}^{\top} - 2|E_G|\boldsymbol{I}. \tag{40}$$

When the graph G is connected, the eigenspace of $\boldsymbol{L}$ associated with eigenvalue $\lambda = 0$ only has one dimension, and any eigenvector $\boldsymbol{b}$ satisfying $\boldsymbol{L}\boldsymbol{b} = \mathbf{0}$ has the form $\boldsymbol{b} = c\mathbf{1}$ for some c. This implies that all solutions to Equation (40) has the form

$$\boldsymbol{M} = \boldsymbol{L}^{\dagger}\boldsymbol{D}\boldsymbol{J} + \boldsymbol{J}\boldsymbol{S} - 2|E_G|\boldsymbol{L}^{\dagger} \tag{41}$$

where $\boldsymbol{S}$ is a diagonal matrix and $\boldsymbol{J} = \mathbf{1}\mathbf{1}^{\top}$. Noting that $\mathrm{diag}(\boldsymbol{M}) = \boldsymbol{O}$, we have $\boldsymbol{O} = \mathrm{diag}(\boldsymbol{L}^{\dagger}\boldsymbol{D}\boldsymbol{J}) + \mathrm{diag}(\boldsymbol{J}\boldsymbol{S}) - 2|E_G|\mathrm{diag}(\boldsymbol{L}^{\dagger})$. Since $\mathrm{diag}(\boldsymbol{J}\boldsymbol{S}) = \boldsymbol{S}$ for any diagonal matrix $\boldsymbol{S}$ and $\boldsymbol{J}\mathrm{diag}(\boldsymbol{L}^{\dagger}\boldsymbol{D}\boldsymbol{J}) = \boldsymbol{J}\boldsymbol{D}\boldsymbol{L}^{\dagger}$, we final obtain

$$\begin{aligned}\boldsymbol{M} &= \boldsymbol{L}^{\dagger}\boldsymbol{D}\boldsymbol{J} + \boldsymbol{J}\left(2|E_G|\mathrm{diag}(\boldsymbol{L}^{\dagger}) - \mathrm{diag}(\boldsymbol{L}^{\dagger}\boldsymbol{D}\boldsymbol{J})\right) - 2|E_G|\boldsymbol{L}^{\dagger}\\ &= \boldsymbol{L}^{\dagger}\boldsymbol{D}\boldsymbol{J} - \boldsymbol{J}\boldsymbol{D}\boldsymbol{L}^{\dagger} + 2|E_G|\boldsymbol{J}\mathrm{diag}(\boldsymbol{L}^{\dagger}) - 2|E_G|\boldsymbol{L}^{\dagger}.\end{aligned} \tag{42}$$

Below, we will prove that $\bar{\chi} \preceq \boldsymbol{M}$. Noting that $\boldsymbol{L}^{\dagger} = \boldsymbol{D}^{-1/2}\hat{\boldsymbol{L}}^{\dagger}\boldsymbol{D}^{-1/2}$, we have

$$\boldsymbol{M} = \boldsymbol{D}^{-1/2}\hat{\boldsymbol{L}}^{\dagger}\boldsymbol{D}^{1/2}\boldsymbol{J} - \boldsymbol{J}\boldsymbol{D}^{1/2}\hat{\boldsymbol{L}}^{\dagger}\boldsymbol{D}^{-1/2} + 2|E_G|\boldsymbol{J}\boldsymbol{D}^{-1/2}\mathrm{diag}(\hat{\boldsymbol{L}}^{\dagger})\boldsymbol{D}^{-1/2} - 2|E_G|\boldsymbol{D}^{-1/2}\hat{\boldsymbol{L}}^{\dagger}\boldsymbol{D}^{-1/2}. \tag{43}$$

Equivalently, for any graph $G \in \mathcal{G}$ and vertices $u, v \in V_G$,

$$\begin{aligned}\boldsymbol{M}_G(u, v) &= \boldsymbol{D}_G^{-1/2}(u, u)\sum_{w \in V_G}\hat{\boldsymbol{L}}_G^{\dagger}(u, w)\boldsymbol{D}_G^{1/2}(w, w) - \boldsymbol{D}_G^{-1/2}(v, v)\sum_{w \in V_G}\hat{\boldsymbol{L}}_G^{\dagger}(w, v)\boldsymbol{D}_G^{1/2}(w, w)\\ &\quad + 2|E_G|\hat{\boldsymbol{L}}_G^{\dagger}(v, v)\boldsymbol{D}_G^{-1}(v, v) - 2|E_G|\boldsymbol{D}_G^{-1/2}(u, u)\hat{\boldsymbol{L}}_G^{\dagger}(u, v)\boldsymbol{D}_G^{-1/2}(v, v).\end{aligned} \tag{44}$$

From the above equation, one can see that $\boldsymbol{M}_G(u, v)$ is fully determined by the following tuple:

$$\left(|E_G|, \deg_G(u), \deg_G(v), \hat{\boldsymbol{L}}_G^{\dagger}(v, v), \hat{\boldsymbol{L}}_G^{\dagger}(u, v), \{\!\{(\hat{\boldsymbol{L}}_G^{\dagger}(u, w), \deg_G(w)) : w \in V_G\}\!\}, \{\!\{(\hat{\boldsymbol{L}}_G^{\dagger}(w, v), \deg_G(w)) : w \in V_G\}\!\}\right).$$

According to the eigen-decomposition, $\hat{\boldsymbol{L}}^{\dagger} = \sum_{i:\lambda_i \neq 0}\lambda_i^{-1}\boldsymbol{P}_i$ where $\sum_i \lambda_i\boldsymbol{P}_i$ is the eigen-decomposition of $\hat{\boldsymbol{L}}$. Therefore, $\hat{\boldsymbol{L}}^{\dagger} \preceq \mathcal{P}^{\hat{\boldsymbol{L}}}$. Besides, based on Proposition A.14 we have that $\deg_G(u)$ is determined by $\chi_G(u)$, $\deg_G(v)$ is determined by $\chi_G(v)$, and $\mathcal{P}_G^{\hat{\boldsymbol{L}}}(v, v)$ is determined by $\chi_G(v)$. It follows that $|E_G|$ is determined by $\chi_G(u)$, because by $\chi \equiv T_{\mathsf{EP},\hat{\boldsymbol{L}}}(\chi)$ we have for all graphs $G, H \in \mathcal{G}$ and vertices $u \in V_G, x \in V_H$,

$$\begin{aligned}\chi_G(u) = \chi_H(x) &\implies \{\!\{\chi_G(v) : v \in V_G\}\!\} = \{\!\{\chi_H(y) : y \in V_H\}\!\}\\ &\implies \{\!\{\deg_G(v) : v \in V_G\}\!\} = \{\!\{\deg_H(y) : y \in V_H\}\!\}\\ &\implies |E_G| = |E_H|.\end{aligned}$$

We next prove that $\{\!\{(\hat{\boldsymbol{L}}_G^{\dagger}(u, w), \deg_G(w)) : w \in V_G\}\!\}$ is determined by $\chi_G(u)$. Again by using $\chi \equiv T_{\mathsf{EP},\hat{\boldsymbol{L}}}(\chi)$, we have for all graphs $G, H \in \mathcal{G}$ and vertices $u \in V_G, x \in V_H$,

$$\begin{aligned}\chi_G(u) = \chi_H(x) &\implies \{\!\{(\chi_G(v), \mathcal{P}_G^{\hat{\boldsymbol{L}}}(u, v)) : v \in V_G\}\!\} = \{\!\{(\chi_H(y), \mathcal{P}_H^{\hat{\boldsymbol{L}}}(x, y)) : y \in V_H\}\!\}\\ &\implies \{\!\{(\deg_G(v), \hat{\boldsymbol{L}}_G^{\dagger}(u, v)) : v \in V_G\}\!\} = \{\!\{(\deg_H(y), \hat{\boldsymbol{L}}_H^{\dagger}(x, y)) : y \in V_H\}\!\}.\end{aligned}$$

Using the same analysis and noting that $\boldsymbol{M}_G^{\dagger}(w, v) = \boldsymbol{M}_G^{\dagger}(v, w)$ for any symmetric matrix $\boldsymbol{M}$, we can prove that $\{\!\{(\hat{\boldsymbol{L}}_G^{\dagger}(w, v), \deg_G(w)) : w \in V_G\}\!\}$ is determined by $\chi_G(v)$. Combining all these relations leads to the conclusion that $\bar{\chi} \preceq \boldsymbol{M}$. $\qquad\square$

Corollary A.18. *Let M be the commute-time distance matrix. Then, $\bar{\chi} \preceq M$.*

Proof. This is a simple consequence of Lemma A.17. Denoting by H the hitting-time distance matrix, we have $M = H + H^\top$. Since $\bar{\chi} \preceq H$, $\bar{\chi} \preceq H^\top$ (because $\mathcal{P}_G^{\hat{L}}(u, v) = \mathcal{P}_G^{\hat{L}}(v, u)$ for any graph G and vertices $u, v \in V_G$). Therefore, $\bar{\chi} \preceq M$. $\qquad\square$

Lemma A.19. *Let M be the PageRank distance matrix associated with weight sequence $\gamma_0, \gamma_1, \cdots$. Then, $\bar{\chi} \preceq M$.*

Proof. By definition of PageRank distance, $M = \sum_{k=0}^{\infty} \gamma_k (D^{-1}A)^k$. Let $\sum_i \lambda_i P_i$ be the eigen-decomposition of $\hat{L}$. Then, $\hat{A} = D^{-1/2}AD^{-1/2} = I - \hat{L} = \sum_i (1 - \lambda_i)P_i$. Therefore,

$$(D^{-1}A)^k = D^{-1/2}\hat{A}^k D^{1/2} = \sum_i (1 - \lambda_i)^k D^{-1/2}P_i D^{1/2}. \tag{45}$$

Plugging the above equation into the definition of PageRank distance, we have for any graph $G \in \mathcal{G}$ and vertices $u, v \in V_G$,

$$M_G(u, v) = \sum_i \left(\sum_{k=0}^{\infty} \gamma_k (1 - \lambda_i)^k \right) P_i(u, v)(\deg_G(u))^{-1/2}(\deg_G(v))^{1/2}. \tag{46}$$

One can see that $M_G(u, v)$ is purely determined by the tuple $(\deg_G(u), \deg_G(v), \mathcal{P}_G^{\hat{L}}(u, v))$. Based on Proposition A.14, $\chi_G(u)$ determines $\deg_G(u)$ and $\chi_G(v)$ determines $\deg_G(v)$. Consequently, $\bar{\chi} \preceq M$. $\qquad\square$

We next study the (normalized) diffusion distance. Consider the continuous graph diffusion process defined as follows. Given time $t \geq 0$, let $p_G^t(u)$ be the probability "mass" of particles at position u. The particles will move following the differential equation given below:

$$\frac{\mathrm{d}}{\mathrm{d}t}p^t = Tp^t, \tag{47}$$

where T is the transition matrix. For example, when $T = AD^{-1} - I$, the differential equation essentially characterizes a random walk diffusion process. In this paper, we consider the normalized diffusion distance, which corresponds to $T = \hat{A} - I = -\hat{L}$. Given hyperparameter $\tau \geq 0$, denote by $(p|_{G^u})^\tau$ be the probability "mass" vector at time τ with the initial configuration $(p|_{G^u})^0(u) = 1$ and $(p|_{G^u})^0(v) = 0$ for all $v \neq u$. Then, the diffusion distance matrix M is defined as

$$M_G(u, v) = \|(p|_{G^u})^\tau - (p|_{G^v})^\tau\|_2. \tag{48}$$

Lemma A.20. *Let M be the normalized diffusion distance defined above. Then, $\bar{\chi} \preceq M$.*

Proof. Since Equation (47) is a linear differential equation, we can solve it and obtain $(p|_{G^u})^\tau = \exp(\tau T)(p|_{G^u})^0$. Therefore,

$$M_G(u, v) = \left\| \exp(\tau T)\left((p|_{G^u})^0 - (p|_{G^v})^0\right) \right\|_2, \tag{49}$$

where $\exp(T)$ is the matrix exponential of T. Equivalently,

$$M_G^2(u, v) = \left((p|_{G^u})^0 - (p|_{G^v})^0\right)^\top \exp(2\tau T)\left((p|_{G^u})^0 - (p|_{G^v})^0\right). \tag{50}$$

Let $\sum_i \lambda_i P_i$ be the eigen-decomposition of $\hat{L}$. Then, $\exp(2\tau T) = \exp(-2\tau \hat{L}) = \sum_i \exp(-2\tau \lambda_i)P_i$. Therefore,

$$M_G^2(u, v) = \sum_i \exp(-2\tau \lambda_i)(P_i(u, u) + P_i(v, v) - 2P_i(u, v)). \tag{51}$$

This implies that $M_G(u, v)$ is purely determined by the tuple $(\mathcal{P}_G^{\hat{L}}(u, u), \mathcal{P}_G^{\hat{L}}(v, v), \mathcal{P}_G^{\hat{L}}(u, v))$. We conclude that $\bar{\chi} \preceq M$ by using Proposition A.14. $\qquad\square$

We finally study the biharmonic distance.

Lemma A.21. *Let M be the biharmonic distance defined in Lipman et al. (2010). Then, $\bar{\chi} \preceq M$.*

Proof. Without loss of generality, we only consider connected graphs. According to Wei et al. (2021), the biharmonic distance for a connected graph can be equivalently written as

$$M_G(u, v) = (L^2)^\dagger(u, u) + (L^2)^\dagger(v, v) - 2(L^2)^\dagger(u, v). \tag{52}$$

The subsequent proof is almost the same as in Lemma A.16 and we omit it for clarity. □

A.5. Proof of Theorem 6.1

This section aims to prove Theorem 6.1. Below, we will decompose the proof into three parts. First, we will describe the color refinement algorithm corresponding to Spectral IGN, which is equivalent to Spectral IGN in terms of distinguishing non-isomorphic graphs. Then, we will prove that this color refinement algorithm is more expressive than EPWL using the color refinement terminologies defined in Appendix A.1. Finally, we will prove the other direction, i.e., EPWL is more expressive than the color refinement algorithm of Spectral IGN, thus concluding the proof of Theorem 6.1.

Color refinement algorithm for Spectral IGN. To define the algorithm, we first need to extend several concepts defined in Appendix A.1 to incorporate eigenvalues. Formally, let Λ^M be the graph spectrum invariant representing the set of eigenvalues for graph matrix M, i.e., $\Lambda^M(G) := \{\lambda : \lambda \text{ is an eigenvalue of } M_G\}$ for $G \in \mathcal{G}$. Define

$$\mathcal{G}_k^{\mathcal{P}^M} := \{(G^u, \lambda) : G^u \in \mathcal{G}_k, \lambda \in \Lambda^M(G)\}. \tag{53}$$

We can then define color mappings over $\mathcal{G}_k^{\mathcal{P}^M}$. Formally, a function χ defined over domain $\mathcal{G}_k^{\mathcal{P}^M}$ is called a color mapping if $\chi(G^u, \lambda) = \chi(H^v, \mu)$ holds for all $(G^u, \lambda), (H^v, \mu) \in \mathcal{G}_k^{\mathcal{P}^M}$ satisfying $G^u \simeq H^v$ and $\lambda = \mu$. Without ambiguity, we will use the notation $\chi_G(\lambda, u)$ to refer to $\chi(G^u, \lambda)$ for $(G^u, \lambda) \in \mathcal{G}_k^{\mathcal{P}^M}$. Define $\mathcal{M}_k^{\mathcal{P}^M}$ to be the family of all color mappings over $\mathcal{G}_k^{\mathcal{P}^M}$. We can similarly define equivalence relation "$\equiv$" and partial relation "$\preceq$" between color mappings in $\mathcal{M}_k^{\mathcal{P}^M}$. In addition, the color transformation can also be extended in a similar manner.

Throughout this section, we use the notation $\chi^{\mathcal{P}^M} \in \mathcal{M}_2^{\mathcal{P}^M}$ to represent the initial color mapping in Spectral IGN, which is defined as $\chi_G^{\mathcal{P}^M}(\lambda, u, v) = (\lambda, P_\lambda^M(u, v))$ for all $(G^{uv}, \lambda) \in \mathcal{G}_2^{\mathcal{P}^M}$, where P_λ^M is the projection matrix associated with eigenvalue λ. We then define the following color transformations:

- **2-IGN color refinement.** Define $T_{\mathsf{IGN}} : \mathcal{M}_2 \to \mathcal{M}_2$ such that for any color mapping $\chi \in \mathcal{M}_2$ and $G^{uv} \in \mathcal{G}_2$,

$$\begin{aligned}
[T_{\mathsf{IGN}}(\chi)]_G(u, v) = \mathsf{hash}(&\chi_G(u, v), \chi_G(u, u), \chi_G(v, v), \chi_G(v, u), \delta_{uv}(\chi_G(u, u)), \\
&\{\!\{\chi_G(u, w) : w \in V_G\}\!\}, \{\!\{\chi_G(w, u) : w \in V_G\}\!\}, \\
&\{\!\{\chi_G(v, w) : w \in V_G\}\!\}, \{\!\{\chi_G(w, v) : w \in V_G\}\!\}, \\
&\{\!\{\chi_G(w, w) : w \in V_G\}\!\}, \{\!\{\chi_G(w, x) : w, x \in V_G\}\!\}, \\
&\delta_{uv}(\{\!\{\chi_G(u, w) : w \in V_G\}\!\}), \delta_{uv}(\{\!\{\chi_G(w, u) : w \in V_G\}\!\}), \\
&\delta_{uv}(\{\!\{\chi_G(w, w) : w \in V_G\}\!\}), \delta_{uv}(\{\!\{\chi_G(w, x) : w, x \in V_G\}\!\})).
\end{aligned} \tag{54}$$

  Here, the function δ_{uv} satisfies that $\delta_{uv}(c) = c$ if $u = v$ and $\delta_{uv}(c) = 0$ otherwise (0 is a special element that differs from all $\chi_G(u, v)$). One can see that Equation (54) has 15 aggregations inside the hash function, which matches the number of orthogonal bases for a 2-IGN layer (Maron et al., 2019b).

- **Spectral pooling.** Define $T_{\mathsf{SP}} : \mathcal{M}_2^{\mathcal{P}^M} \to \mathcal{M}_2$ such that for any color mapping $\chi \in \mathcal{M}_2^{\mathcal{P}^M}$ and $G^{uv} \in \mathcal{G}_2$,

$$[T_{\mathsf{SP}}(\chi)]_G(u, v) = \mathsf{hash}(\{\!\{\chi_G(\lambda, u, v) : \lambda \in \Lambda^M(G)\}\!\}). \tag{55}$$

- **Spectral IGN color refinement.** Define $T_{\mathsf{SIGN}} : \mathcal{M}_2^{\mathcal{P}^M} \to \mathcal{M}_2^{\mathcal{P}^M}$ such that for any color mapping $\chi \in \mathcal{M}_2^{\mathcal{P}^M}$ and $(G^{uv}, \lambda) \in \mathcal{G}_2^{\mathcal{P}^M}$,

$$[T_{\mathsf{SIGN}}(\chi)]_G(\lambda, u, v) = \mathsf{hash}([T_{\mathsf{IGN}}(\chi(\lambda, \cdot, \cdot))]_G(u, v), [T_{\mathsf{IGN}}(T_{\mathsf{SP}}(\chi))]_G(u, v)). \tag{56}$$

- **Spectral IGN final pooling.** Define $T_{\mathsf{FP}} : \mathcal{M}_2^{\mathcal{P}^M} \to \mathcal{M}_0$ such that for any color mapping $\chi \in \mathcal{M}_2^{\mathcal{P}^M}$ and $G \in \mathcal{G}$,

$$[T_{\mathsf{FP}}(\chi)](G) = \mathsf{hash}(\{\!\{\chi_G(\lambda, u, v) : \lambda \in \Lambda^M(G), u, v \in V_G\}\!\}). \tag{57}$$

- **Joint pooling.** Define $T_{\mathsf{JP}} : \mathcal{M}_2 \to \mathcal{M}_0$ such that for any color mapping $\chi \in \mathcal{M}_2$ and $G \in \mathcal{G}$,

$$[T_{\mathsf{JP}}(\chi)](G) = \mathsf{hash}(\{\!\{\chi_G(u,v) : u,v \in V_G\}\!\}). \tag{58}$$

- **2-dimensional Pooling.** Define $T_{\mathsf{P2}} : \mathcal{M}_2 \to \mathcal{M}_1$ such that for any color mapping $\chi \in \mathcal{M}_2$ and $G \in \mathcal{G}_1$,

$$[T_{\mathsf{P2}}(\chi)]_G(u) = \mathsf{hash}(\{\!\{\chi_G(u,v) : v \in V_G\}\!\}). \tag{59}$$

- **EPWL color refinement.** Define $T_{\mathsf{EP},M} : \mathcal{M}_1 \to \mathcal{M}_1$ such that for any color mapping $\chi \in \mathcal{M}_1$ and $G^u \in \mathcal{G}_1$,

$$[T_{\mathsf{EP},M}(\chi)]_G(u) = \mathsf{hash}(\chi_G(u), \{\!\{(\chi_G(v), \mathcal{P}_G^M(u,v)) : v \in V_G\}\!\}). \tag{60}$$

- **Global pooling.** Define $T_{\mathsf{GP}} : \mathcal{M}_1 \to \mathcal{M}_0$ such that for any color mapping $\chi \in \mathcal{M}_1$ and graph G,

$$[T_{\mathsf{GP}}(\chi)](G) = \mathsf{hash}(\{\!\{\chi_G(u) : u \in V_G\}\!\}). \tag{61}$$

We now related the expressive power of Spectral IGN to the corresponding color refinement algorithm. The proof is straightforward following standard techniques, see e.g., Zhang et al. (2023a).

Proposition A.22. *The expressive power of Spectral IGN is bounded by the color mapping* $(T_{\mathsf{FP}} \circ T_{\mathsf{SIGN}}^{\infty})(\chi^{\mathcal{P}^M})$ *in distinguishing non-isomorphic graphs. Moreover, with sufficient layers and proper network parameters, Spectral IGN can be as expressive as the above color mapping in distinguishing non-isomorphic graphs.*

Equivalently, the pooling T_{FP} can be decomposed into three pooling transformations $T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_{\mathsf{SP}}$, as stated below:

Lemma A.23. $T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_{\mathsf{SP}} \circ T_{\mathsf{SIGN}}^{\infty} \equiv T_{\mathsf{FP}} \circ T_{\mathsf{SIGN}}^{\infty}$.

Proof. First, it is clear that $T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_{\mathsf{SP}} \circ T_{\mathsf{SIGN}}^{\infty} \preceq T_{\mathsf{FP}} \circ T_{\mathsf{SIGN}}^{\infty}$. Thus, it suffices to prove that $T_{\mathsf{FP}} \circ T_{\mathsf{SIGN}}^{\infty} \preceq T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_{\mathsf{SP}} \circ T_{\mathsf{SIGN}}^{\infty}$. Pick any initial color mapping $\chi^0 \in \mathcal{M}_2^{\mathcal{P}^M}$ and let $\chi = T_{\mathsf{SIGN}}^{\infty}(\chi^0)$. Note that $\chi \equiv T_{\mathsf{SIGN}}(\chi)$. We will prove that $(T_{\mathsf{FP}})(\chi) \preceq (T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_{\mathsf{SP}})(\chi)$. Pick any graphs $G, H \in \mathcal{G}$. We have

$$[T_{\mathsf{FP}}(\chi)](G) = [T_{\mathsf{FP}}(\chi)](H)$$
$$\implies \{\!\{\chi_G(\lambda, u, v) : \lambda \in \Lambda^M(G), u, v \in V_G\}\!\} = \{\!\{\chi_H(\mu, x, y) : \mu \in \Lambda^M(H), x, y \in V_H\}\!\}$$
$$\implies \{\!\{\chi_G(\lambda, u, u) : \lambda \in \Lambda^M(G), u \in V_G\}\!\} = \{\!\{\chi_H(\mu, x, x) : \mu \in \Lambda^M(H), x \in V_H\}\!\}$$
$$\implies \{\!\{\{\!\{\chi_G(\lambda, u, v) : v \in V_G\}\!\} : \lambda \in \Lambda^M(G), u \in V_G\}\!\} = \{\!\{\{\!\{\chi_H(\mu, x, y) : y \in V_H\}\!\} : \mu \in \Lambda^M(H), x \in V_H\}\!\}$$
$$\implies \{\!\{\{\!\{\{\!\{\chi_G(\lambda', u, v) : \lambda' \in \Lambda^M(G)\}\!\} : v \in V_G\}\!\} : \lambda \in \Lambda^M(G), u \in V_G\}\!\}$$
$$\quad = \{\!\{\{\!\{\{\!\{\chi_H(\mu', x, y) : \mu' \in \Lambda^M(H)\}\!\} : y \in V_H\}\!\} : \mu \in \Lambda^M(H), x \in V_H\}\!\}$$
$$\implies \{\!\{\{\!\{\{\!\{\chi_G(\lambda', u, v) : \lambda' \in \Lambda^M(G)\}\!\} : v \in V_G\}\!\} : u \in V_G\}\!\}$$
$$\quad = \{\!\{\{\!\{\{\!\{\chi_H(\mu', x, y) : \mu' \in \Lambda^M(H)\}\!\} : y \in V_H\}\!\} : x \in V_H\}\!\}$$
$$\implies [(T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_{\mathsf{SP}})(\chi)](G) = [(T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_{\mathsf{SP}})(\chi)](H),$$

where the second, third, and fourth steps in the above derivation are based on Equations (54) to (56). We have obtained the desired result. $\qquad\square$

In the subsequent proof, we will show that $(T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_{\mathsf{SP}} \circ T_{\mathsf{SIGN}}^{\infty})(\chi^{\mathcal{P}^M}) \equiv (T_{\mathsf{GP}} \circ T_{\mathsf{EP},M})(\chi^0)$, where $\chi^0 \in \mathcal{M}_1$ is the constant mapping.

Lemma A.24. *Let* $\chi^0 \in \mathcal{M}_1$ *be the constant mapping. Then,* $(T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_{\mathsf{SP}} \circ T_{\mathsf{SIGN}}^{\infty})(\chi^{\mathcal{P}^M}) \preceq (T_{\mathsf{GP}} \circ T_{\mathsf{EP},M}^{\infty})(\chi^0)$.

Proof. Define an auxiliary color transformation $T_{\times} : \mathcal{M}_1 \to \mathcal{M}_2$ such that for any color mapping $\chi \in \mathcal{M}_1$ and rooted graph $G^{uv} \in \mathcal{G}_2$,

$$[T_{\times}(\chi)]_G(u, v) = \mathsf{hash}(\chi_G(u), \chi_G(v)). \tag{62}$$

We first prove that $T_{\mathsf{GP}} \equiv T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_\times$. Pick any color mapping $\chi \in \mathcal{M}_1$ and graphs $G, H \in \mathcal{G}$,

$$[T_{\mathsf{GP}}(\chi)](G) = [T_{\mathsf{GP}}(\chi)](H)$$
$$\iff \{\!\{\chi_G(u) : u \in V_G\}\!\} = \{\!\{\chi_H(x) : x \in V_H\}\!\}$$
$$\iff \{\!\{(\chi_G(u), \{\!\{\chi_G(v) : v \in V_G\}\!\}) : u \in V_G\}\!\} = \{\!\{(\chi_H(x), \{\!\{\chi_H(y) : y \in V_H\}\!\}) : x \in V_H\}\!\}$$
$$\iff \{\!\{\{\!\{(\chi_G(u), \chi_G(v)) : v \in V_G\}\!\} : u \in V_G\}\!\} = \{\!\{\{\!\{(\chi_H(x), \chi_H(y)) : y \in V_H\}\!\} : x \in V_H\}\!\}$$
$$\iff [(T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_\times)(\chi)](G) = [(T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_\times)(\chi)](H).$$

Based on the equivalence, it suffices to prove that $(T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_{\mathsf{SP}} \circ T_{\mathsf{SIGN}}^\infty)(\chi^{\mathcal{P}^M}) \preceq (T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_\times \circ T_{\mathsf{EP},M}^\infty)(\chi^0)$. Since $T_{\mathsf{GP}} \circ T_{\mathsf{P2}}$ is order-preserving by definition, it suffices to prove that $(T_{\mathsf{SP}} \circ T_{\mathsf{SIGN}}^\infty)(\chi^{\mathcal{P}^M}) \preceq (T_\times \circ T_{\mathsf{EP},M}^\infty)(\chi^0)$.

We will prove the following stronger result: for all $t \in \mathbb{N}$, $(T_{\mathsf{SP}} \circ T_{\mathsf{SIGN}}^t)(\chi^{\mathcal{P}^M}) \preceq (T_\times \circ T_{\mathsf{EP},M}^t)(\chi^0)$. The proof is based on induction. For the base case of $t = 0$, since $(T_\times \circ T_{\mathsf{EP},M}^t)(\chi^0)$ is a constant mapping, the result clearly holds. Now assume that the result holds for $t = t'$ and consider the case of $t = t' + 1$. Denote $\chi = T_{\mathsf{SIGN}}^{t'}(\chi^{\mathcal{P}^M})$, $\hat{\chi} = T_{\mathsf{EP},M}^t(\chi^0)$, and note that $T_{\mathsf{SP}}(\chi) \preceq T_{\mathsf{SP}}(\chi^{\mathcal{P}^M}) \equiv \mathcal{P}^M$. Pick any graphs $G, H \in \mathcal{G}$ and vertices $u, v \in V_G$, $x, y \in V_H$. Based on the induction hypothesis, $[T_{\mathsf{SP}}(\chi)]_G(u, v) = [T_{\mathsf{SP}}(\chi)]_H(x, y)$ implies that $\hat{\chi}_G(u) = \hat{\chi}_H(x)$ and $\hat{\chi}_G(v) = \hat{\chi}_H(y)$. We have

$$[T_{\mathsf{SP}}(T_{\mathsf{SIGN}}(\chi))]_G(u, v) = [T_{\mathsf{SP}}(T_{\mathsf{SIGN}}(\chi))]_H(x, y)$$
$$\implies \{\!\{[T_{\mathsf{IGN}}(T_{\mathsf{SP}}(\chi))]_G(u, v) : \lambda \in \Lambda^M(G)\}\!\} = \{\!\{[T_{\mathsf{IGN}}(T_{\mathsf{SP}}(\chi))]_H(x, y) : \mu \in \Lambda^M(H)\}\!\}$$
$$\implies [T_{\mathsf{IGN}}(T_{\mathsf{SP}}(\chi))]_G(u, v) = [T_{\mathsf{IGN}}(T_{\mathsf{SP}}(\chi))]_H(x, y)$$
$$\implies [T_{\mathsf{SP}}(\chi)]_G(u, v) = [T_{\mathsf{SP}}(\chi)]_H(x, y) \wedge \{\!\{[T_{\mathsf{SP}}(\chi)]_G(u, w) : w \in V_G\}\!\} = \{\!\{[T_{\mathsf{SP}}(\chi)]_H(x, z) : z \in V_H\}\!\}$$
$$\wedge \{\!\{[T_{\mathsf{SP}}(\chi)]_G(v, w) : w \in V_G\}\!\} = \{\!\{[T_{\mathsf{SP}}(\chi)]_H(y, z) : z \in V_H\}\!\}$$
$$\implies \hat{\chi}_G(u) = \hat{\chi}_H(x) \wedge \{\!\{(\hat{\chi}_G(w), \mathcal{P}_G^M(u, w)) : w \in V_G\}\!\} = \{\!\{(\hat{\chi}_H(z), \mathcal{P}_H^M(x, z)) : z \in V_H\}\!\}$$
$$\wedge \hat{\chi}_G(v) = \hat{\chi}_H(y) \wedge \{\!\{(\hat{\chi}_G(w), \mathcal{P}_G^M(v, w)) : w \in V_G\}\!\} = \{\!\{(\hat{\chi}_H(z), \mathcal{P}_H^M(y, z)) : z \in V_H\}\!\}$$
$$\implies [T_{\mathsf{EP},M}(\hat{\chi})]_G(u) = [T_{\mathsf{EP},M}(\hat{\chi})]_H(x) \wedge [T_{\mathsf{EP},M}(\hat{\chi})]_G(v) = [T_{\mathsf{EP},M}(\hat{\chi})]_H(y)$$
$$\implies [(T_\times \circ T_{\mathsf{EP},M})(\hat{\chi})]_G(u, v) = [(T_\times \circ T_{\mathsf{EP},M})(\hat{\chi})]_H(x, y)$$

where in the first step we use the definition of Spectral IGN (Equation (56)) and spectral pooling (Equation (55)); in the third step we use the definition of 2-IGN (Equation (54)); in the fourth step we use the induction hypothesis and the fact that $T_{\mathsf{SP}}(\chi) \preceq \mathcal{P}^M$. This concludes the induction step. $\qquad\square$

Lemma A.25. *Define an auxiliary color transformation* $T_{\times, \mathcal{P}^M} : \mathcal{M}_1 \to \mathcal{M}_2$ *such that for any color mapping* $\chi \in \mathcal{M}_1$ *and rooted graph* $G^{uv} \in \mathcal{G}_2$,

$$[T_{\times, \mathcal{P}^M}(\chi)]_G(u, v) = \mathsf{hash}(\chi_G(u), \chi_G(v), \mathcal{P}_G^M(u, v)). \tag{63}$$

Then, $(T_{\times, \mathcal{P}^M} \circ T_{\mathsf{EP},M}^\infty)(\chi^0) \preceq (T_{\mathsf{SP}} \circ T_{\mathsf{SIGN}}^\infty)(\chi^{\mathcal{P}^M})$.

Proof. We will prove the following stronger result: for any $t \geq 0$, $(T_{\times, \mathcal{P}^M} \circ T_{\mathsf{EP},M}^{2t})(\chi^0) \preceq (T_{\mathsf{SP}} \circ T_{\mathsf{SIGN}}^t)(\chi^{\mathcal{P}^M})$. The proof is based on induction. For the base case of $t = 0$, since $(T_{\times, \mathcal{P}^M} \circ T_{\mathsf{EP},M}^0)(\chi^0) = T_{\times, \mathcal{P}^M}(\chi^0) \equiv T_{\mathsf{SP}} \circ \chi^{\mathcal{P}^M}$, the result clearly holds. Now assume that the result holds for $t = t'$ and consider the case of $t = t' + 1$. Denote $\chi = T_{\mathsf{EP},M}^{2t'}(\chi^0)$ and $\hat{\chi} = T_{\mathsf{SIGN}}^{t'}(\chi^{\mathcal{P}^M})$. Pick any graphs $G, H \in \mathcal{G}$ and vertices $u, v \in V_G$, $x, y \in V_H$. Based on the induction hypothesis, $[T_{\times, \mathcal{P}^M}(\chi)]_G(u, v) = [T_{\times, \mathcal{P}^M}(\chi)]_H(x, y)$ implies that $[T_{\mathsf{SP}}(\hat{\chi})]_G(u, v) = [T_{\mathsf{SP}}(\hat{\chi})]_H(x, y)$. We have

$$[(T_{\times, \mathcal{P}^M} \circ T_{\mathsf{EP},M}^2)(\chi)]_G(u, v) = [(T_{\times, \mathcal{P}^M} \circ T_{\mathsf{EP},M}^2)(\chi)]_H(x, y)$$
$$\implies [T_{\mathsf{EP},M}^2(\chi)]_G(u) = [T_{\mathsf{EP},M}^2(\chi)]_H(x) \wedge [T_{\mathsf{EP},M}^2(\chi)]_G(v) = [T_{\mathsf{EP},M}^2(\chi)]_H(y) \wedge \mathcal{P}_G^M(u, v) = \mathcal{P}_H^M(x, y)$$
$$\implies \chi_G(u) = \chi_H(x) \wedge \chi_G(v) = \chi_H(y) \wedge \mathcal{P}^M{}_G(u, v) = \mathcal{P}^M{}_H(x, y)$$
$$\wedge \mathcal{P}_G^M(u, u) = \mathcal{P}_H^M(x, x) \wedge \mathcal{P}_G^M(v, v) = \mathcal{P}_H^M(y, y) \wedge \mathbb{I}[u = v] = \mathbb{I}[x = y]$$
$$\wedge \{\!\{([T_{\mathsf{EP},M}(\chi)]_G(w), \mathcal{P}_G^M(u, w)) : w \in V_G\}\!\} = \{\!\{([T_{\mathsf{EP},M}(\chi)]_H(z), \mathcal{P}_H^M(x, z)) : z \in V_G\}\!\}$$

$$\wedge \{\!\{([T_{\mathsf{EP},M}(\chi)]_G(w), \mathcal{P}_G^M(v,w)) : w \in V_G\}\!\} = \{\!\{([T_{\mathsf{EP},M}(\chi)]_H(z), \mathcal{P}_H^M(y,z)) : z \in V_G\}\!\}$$

$$\implies \chi_G(u) = \chi_H(x) \wedge \chi_G(v) = \chi_H(y) \wedge \mathcal{P}_G^M(u,v) = \mathcal{P}_H^M(x,y)$$

$$\wedge \mathcal{P}_G^M(u,u) = \mathcal{P}_H^M(x,x) \wedge \mathcal{P}_G^M(v,v) = \mathcal{P}_H^M(y,y) \wedge \mathbb{I}[u=v] = \mathbb{I}[x=y]$$

$$\wedge \{\!\{(\chi_G(u), \chi_G(w), \mathcal{P}_G^M(u,w)) : w \in V_G\}\!\} = \{\!\{(\chi_H(x), \chi_H(z), \mathcal{P}_H^M(x,z)) : z \in V_G\}\!\}$$

$$\wedge \{\!\{(\chi_G(v), \chi_G(w), \mathcal{P}_G^M(v,w)) : w \in V_G\}\!\} = \{\!\{(\chi_H(y), \chi_H(z), \mathcal{P}_H^M(y,z)) : z \in V_G\}\!\}$$

$$\wedge \{\!\{(\chi_G(w), \mathcal{P}_G^M(w,w)) : w \in V_G\}\!\} = \{\!\{(\chi_H(z), \mathcal{P}_H^M(z,z)) : z \in V_G\}\!\}$$

$$\wedge \{\!\{(\chi_G(w), \chi_G(w'), \mathcal{P}_G^M(w,w')) : w,w' \in V_G\}\!\} = \{\!\{(\chi_H(z), \chi_H(z'), \mathcal{P}_H^M(z,z')) : z,z' \in V_G\}\!\}$$

$$\implies [T_{\mathsf{SP}}(\hat\chi)]_G(u,v) = [T_{\mathsf{SP}}(\hat\chi)]_H(x,y) \wedge [T_{\mathsf{SP}}(\hat\chi)]_G(u,u) = [T_{\mathsf{SP}}(\hat\chi)]_H(x,x)$$

$$\wedge [T_{\mathsf{SP}}(\hat\chi)]_G(v,v) = [T_{\mathsf{SP}}(\hat\chi)]_H(y,y) \wedge [T_{\mathsf{SP}}(\hat\chi)]_G(v,u) = [T_{\mathsf{SP}}(\hat\chi)]_H(y,x) \wedge \mathbb{I}[u=v] = \mathbb{I}[x=y]$$

$$\wedge \{\!\{[T_{\mathsf{SP}}(\hat\chi)]_G(u,w) : w \in V_G\}\!\} = \{\!\{[T_{\mathsf{SP}}(\hat\chi)]_H(x,z) : z \in V_H\}\!\}$$

$$\wedge \{\!\{[T_{\mathsf{SP}}(\hat\chi)]_G(v,w) : w \in V_G\}\!\} = \{\!\{[T_{\mathsf{SP}}(\hat\chi)]_H(y,z) : z \in V_H\}\!\}$$

$$\wedge \{\!\{[T_{\mathsf{SP}}(\hat\chi)]_G(w,u) : w \in V_G\}\!\} = \{\!\{[T_{\mathsf{SP}}(\hat\chi)]_H(z,x) : z \in V_H\}\!\}$$

$$\wedge \{\!\{[T_{\mathsf{SP}}(\hat\chi)]_G(w,v) : w \in V_G\}\!\} = \{\!\{[T_{\mathsf{SP}}(\hat\chi)]_H(z,y) : z \in V_H\}\!\}$$

$$\wedge \{\!\{[T_{\mathsf{SP}}(\hat\chi)]_G(w,w) : w \in V_G\}\!\} = \{\!\{[T_{\mathsf{SP}}(\hat\chi)]_H(z,z) : z \in V_H\}\!\}$$

$$\wedge \{\!\{[T_{\mathsf{SP}}(\hat\chi)]_G(w,w') : w,w' \in V_G\}\!\} = \{\!\{[T_{\mathsf{SP}}(\hat\chi)]_H(z,z') : z,z' \in V_H\}\!\}$$

$$\implies [(T_{\mathsf{IGN}} \circ T_{\mathsf{SP}})(\hat\chi)]_G(u,v) = [(T_{\mathsf{IGN}} \circ T_{\mathsf{SP}})(\hat\chi)]_H(x,y)$$

where in the second step we use the definition of $T_{\mathsf{EP},M}$ and also Lemma A.5 and Proposition A.6; in the third step we use the definition of $T_{\mathsf{EP},M}$ again; in the fourth step we use the induction hypothesis and the fact that $\mathcal{P}_G^M(u,v) = \mathcal{P}_G^M(v,u)$ for all $G \in \mathcal{G}$ and $u,v \in V_G$; in the last step we use the definition of 2-IGN (Equation (54)).

We next prove that $[(T_{\mathsf{IGN}} \circ T_{\mathsf{SP}})(\hat\chi)]_G(u,v) = [(T_{\mathsf{IGN}} \circ T_{\mathsf{SP}})(\hat\chi)]_H(x,y) \implies [(T_{\mathsf{SP}} \circ T_{\mathsf{SIGN}})(\hat\chi)]_G(u,v) = [(T_{\mathsf{SP}} \circ T_{\mathsf{SIGN}})(\hat\chi)]_H(x,y)$. This is because

$$[(T_{\mathsf{IGN}} \circ T_{\mathsf{SP}})(\hat\chi)]_G(u,v) = [(T_{\mathsf{IGN}} \circ T_{\mathsf{SP}})(\hat\chi)]_H(x,y)$$

$$\implies \Lambda^M(G) = \Lambda^M(H) \wedge [T_{\mathsf{IGN}}(\hat\chi(\lambda,\cdot,\cdot))]_G(u,v) = [T_{\mathsf{IGN}}(\hat\chi(\lambda,\cdot,\cdot))]_H(x,y) \quad \forall \lambda \in \Lambda^M(G)$$

$$\implies [(T_{\mathsf{SP}} \circ T_{\mathsf{SIGN}})(\hat\chi)]_G(u,v) = [(T_{\mathsf{SP}} \circ T_{\mathsf{SIGN}})(\hat\chi)]_H(x,y),$$

where in the first step we use the following observations: (i) $\hat\chi \preceq \chi^{\mathcal{P}^M}$, which implies that $\hat\chi_G(\lambda,u,v) = \hat\chi_H(\mu,x,y) \implies \lambda = \mu$; (ii) $(T_{\mathsf{IGN}} \circ T_{\mathsf{SP}})(\hat\chi) \preceq T_{\mathsf{SP}} \circ \chi^{\mathcal{P}^M} \equiv \mathcal{P}^M$ and thus $\Lambda^M(G) = \Lambda^M(H)$. We thus conclude the induction step. $\square$

Lemma A.26. *Let $\chi^0 \in \mathcal{M}_1$ be the constant mapping. Then, $(T_{\mathsf{GP}} \circ T_{\mathsf{EP},M}^\infty)(\chi^0) \preceq (T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_{\mathsf{SP}} \circ T_{\mathsf{SIGN}}^\infty)(\chi^{\mathcal{P}^M})$.*

Proof. Based on Lemma A.25, it suffices to prove that $(T_{\mathsf{GP}} \circ T_{\mathsf{EP},M}^\infty)(\chi^0) \equiv (T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_{\times,\mathcal{P}^M} \circ T_{\mathsf{EP},M}^\infty)(\chi^0)$, where $T_{\times,\mathcal{P}^M}$ is defined in Lemma A.25. Denote $\chi = T_{\mathsf{EP},M}^\infty(\chi^0)$. Let $G,H \in \mathcal{G}$ be any graphs such that $[T_{\mathsf{GP}}(\chi)](G) = [T_{\mathsf{GP}}(\chi)](H)$. By definition of T_{GP},

$$\{\!\{\chi_G(u) : u \in V_G\}\!\} = \{\!\{\chi_H(x) : x \in V_H\}\!\}. \tag{64}$$

Since $\chi \equiv T_{\mathsf{EP},M} \circ \chi$,

$$\{\!\{(\chi_G(u), \{\!\{(\chi_G(v), \mathcal{P}_G^M(u,v)) : v \in V_G\}\!\}) : u \in V_G\}\!\} = \{\!\{(\chi_H(x), \{\!\{(\chi_H(y), \mathcal{P}_H^M(x,y)) : y \in V_H\}\!\}) : x \in V_H\}\!\}. \tag{65}$$

Equivalently,

$$\{\!\{\{\!\{(\chi_G(u), \chi_G(v), \mathcal{P}_G^M(u,v)) : v \in V_G\}\!\} : u \in V_G\}\!\} = \{\!\{\{\!\{(\chi_H(x), \chi_H(y), \mathcal{P}_H^M(x,y)) : y \in V_H\}\!\} : x \in V_H\}\!\}. \tag{66}$$

This implies that $[(T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_{\times,\mathcal{P}^M})(\chi)](G) = [(T_{\mathsf{GP}} \circ T_{\mathsf{P2}} \circ T_{\times,\mathcal{P}^M})(\chi)](H)$, concluding the proof. $\square$

Combining Proposition A.22 and Lemmas A.23, A.24 and A.26, we conclude the proof of Theorem 6.1.

A.6. Proof of Propositions 6.3 and 6.5

This sections aims to prove Propositions 6.3 and 6.5. We will first give a brief introduction of BasisNet. The BasisNet architecture is composed of two parts: an eigenspace encoder $\Phi : \mathbb{R}^{m \times n \times n} \to \mathbb{R}^{n \times d}$ and a top graph encoder $\rho : \mathbb{R}^{n \times d} \to \mathbb{R}^{d'}$. Here, we consider the standard setting where ρ is a messgage-passing GNN that takes node features as inputs and outputs a graph representation invariant to node permutation. We assume that the expressive power of ρ is bounded by the classic 1-WL test.

We next describe the design of the eigenspace encoder Φ. Given graph matrix $\boldsymbol{M}$ and graph G, let $\boldsymbol{M}_G = \sum_{i=1}^m \lambda_i \boldsymbol{P}_i$ be the eigen-decomposition of $\boldsymbol{M}_G$, where $\lambda_1 < \cdots < \lambda_m$ are eigenvalues of $\boldsymbol{M}_G$. For each eigenspace, BasisNet processes the projection matrix $\boldsymbol{P}_i$ using a 2-IGN $\mathsf{IGN}^{(d_i)} : \mathbb{R}^{n \times n} \to \mathbb{R}^n$, where d_i is the multiplicity of eigenvalue λ_i. The output of Φ is then defined as

$$\Phi(G) = [\mathsf{IGN}^{(d_1)}(\boldsymbol{P}_1), \cdots, \mathsf{IGN}^{(d_m)}(\boldsymbol{P}_m), \boldsymbol{0}, \cdots, \boldsymbol{0}] \in \mathbb{R}^{n \times d}, \tag{67}$$

where $[\,]$ denotes the concatenation. When the number of eigenspaces is less than the output dimension d, zero-padding is applied. Note that BasisNet processes different projections using different IGNs if their multiplicities differ.

Color refinement algorithms for Siamese IGN and BasisNet. Similar to Spectral IGN, we can write the corresponding color refinement algorithms for the two GNN architectures. We first define the initial color mapping $\chi^{\mathsf{Basis}, \mathcal{P}^M} \in \mathcal{M}_2^{\mathcal{P}^M}$ as follows: $\chi_G^{\mathsf{Basis}, \mathcal{P}^M}(\lambda, u, v) = (d_\lambda, \boldsymbol{P}_\lambda^M(u, v))$ for all $(G^{uv}, \lambda) \in \mathcal{G}_2^{\mathcal{P}^M}$, where $\boldsymbol{P}_\lambda^M$ is the projection matrix associated with eigenvalue λ and d_λ is the multiplicity of eigenvalue λ. Here, we encode the multiplicity in $\chi^{\mathsf{Basis}, \mathcal{P}^M}$ because BasisNet uses different IGNs for different eigenvalue multiplicities. We then define several color transformations:

- **Siamese IGN color refinement.** Define $T_{\mathsf{Siam}} : \mathcal{M}_2^{\mathcal{P}^M} \to \mathcal{M}_2^{\mathcal{P}^M}$ such that for any color mapping $\chi \in \mathcal{M}_2^{\mathcal{P}^M}$ and $(G^{uv}, \lambda) \in \mathcal{G}_2^{\mathcal{P}^M}$,

$$[T_{\mathsf{Siam}}(\chi)]_G(\lambda, u, v) = [T_{\mathsf{IGN}}(\chi(\lambda, \cdot, \cdot))]_G(u, v), \tag{68}$$

  where T_{IGN} is defined in Equation (54).

- **BasisNet pooling.** Define $T_{\mathsf{BP}} : \mathcal{M}_2^{\mathcal{P}^M} \to \mathcal{M}_1^{\mathcal{P}^M}$ such that for any color mapping $\chi \in \mathcal{M}_2^{\mathcal{P}^M}$ and $(G^u, \lambda) \in \mathcal{G}_1^{\mathcal{P}^M}$,

$$\begin{aligned}[T_{\mathsf{BP}}(\chi)]_G(\lambda, u) = \mathsf{hash}(\chi_G(\lambda, u, u), \{\!\{\chi_G(\lambda, u, v) : v \in V_G\}\!\}, \{\!\{\chi_G(\lambda, v, u) : v \in V_G\}\!\}, \\ \{\!\{\chi_G(\lambda, v, v) : v \in V_G\}\!\}, \{\!\{\chi_G(\lambda, v, w) : v, w \in V_G\}\!\}).\end{aligned} \tag{69}$$

  One can see that Equation (69) has 5 aggregations inside the hash function, which matches the number of orthogonal bases in Maron et al. (2019b).

- **Spectral pooling.** Define $T_{\mathsf{SP1}} : \mathcal{M}_1^{\mathcal{P}^M} \to \mathcal{M}_1$ such that for any color mapping $\chi \in \mathcal{M}_1^{\mathcal{P}^M}$ and $G^u \in \mathcal{G}_1$,

$$[T_{\mathsf{SP1}}(\chi)]_G(u) = \mathsf{hash}(\{\!\{\chi_G(\lambda, u) : \lambda \in \Lambda^M(G)\}\!\}). \tag{70}$$

  Similarly, define $T_{\mathsf{SP2}} : \mathcal{M}_2^{\mathcal{P}^M} \to \mathcal{M}_2$ such that for any color mapping $\chi \in \mathcal{M}_2^{\mathcal{P}^M}$ and $G^{uv} \in \mathcal{G}_2$,

$$[T_{\mathsf{SP2}}(\chi)]_G(u, v) = \mathsf{hash}(\{\!\{\chi_G(\lambda, u, v) : \lambda \in \Lambda^M(G)\}\!\}). \tag{71}$$

- **Joint pooling.** This has been defined in Equation (58).

- **Diagonal pooling.** Define $T_{\mathsf{D}} : \mathcal{M}_2 \to \mathcal{M}_1$ such that for any color mapping $\chi \in \mathcal{M}_2$ and rooted graph G^u,

$$[T_{\mathsf{D}}(\chi)]_G(u) = \chi_G(u, u). \tag{72}$$

- **1-WL refinement.** This has been defined in Equation (11).

- **Global pooling.** This has been defined in Equation (61).

We are ready to define the color mappings corresponding to the whole algorithms:

- **Weak Spectral IGN**: the color mapping is defined as $(T_{\mathsf{JP}} \circ T_{\mathsf{SP2}} \circ T_{\mathsf{Siam}}^\infty)(\chi^{\mathcal{P}^M})$.

- **BasisNet**: the color mapping is defined as $(T_{\mathsf{GP}} \circ T_{\mathsf{WL}} \circ T_{\mathsf{SP1}} \circ T_{\mathsf{BP}} \circ T_{\mathsf{Siam}}^\infty)(\chi^{\mathsf{Basis}, \mathcal{P}^M})$.

Similar to the previous analysis, we can prove that the above two color mappings upper bound the expressive power of the corresponding GNN models. Below, it suffices to prove the following key lemma:

Lemma A.27. *For any graph matrix M, $(T_{\text{JP}} \circ T_{\text{SP2}} \circ T_{\text{Siam}}^\infty)(\chi^{\mathcal{P}^M}) \preceq (T_{\text{GP}} \circ T_{\text{WL}} \circ T_{\text{SP1}} \circ T_{\text{BP}} \circ T_{\text{Siam}}^\infty)(\chi^{\text{Basis},\mathcal{P}^M})$.*

Proof. The proof will be decomposed into a series of steps. We first prove that $T_{\text{Siam}}^\infty(\chi^{\mathcal{P}^M}) \preceq T_{\text{Siam}}^\infty(\chi^{\text{Basis},\mathcal{P}^M})$. It suffices to prove that $T_{\text{Siam}}(\chi^{\mathcal{P}^M}) \preceq \chi^{\text{Basis},\mathcal{P}^M}$. Pick any graphs $G, H \in \mathcal{G}$, eigenvalues $\lambda \in \Lambda^M(G), \mu \in \Lambda^M(H)$, and vertices $u, v \in V_G, x, y \in V_H$. Then, by definition of T_{Siam}, $[T_{\text{Siam}}(\chi^{\mathcal{P}^M})]_G(\lambda, u, v) = [T_{\text{Siam}}(\chi^{\mathcal{P}^M})]_H(\mu, x, y)$ implies that

$$\chi_G^{\mathcal{P}^M}(\lambda, u, v) = \chi_H^{\mathcal{P}^M}(\mu, x, y) \wedge \{\!\!\{\chi_G^{\mathcal{P}^M}(\lambda, w, w) : w \in V_G\}\!\!\} = \{\!\!\{\chi_H^{\mathcal{P}^M}(\mu, z, z) : z \in V_H\}\!\!\}. \tag{73}$$

Therefore, $\text{tr}([P_\lambda^M]_G) = \text{tr}([P_\mu^M]_H)$, where $\text{tr}(\cdot)$ denotes the matrix trace. Noting that $\text{tr}([P_\lambda^M]_G)$ is exactly the multiplicity of eigenvalue λ for graph matrix M_G, we have $\chi_G^{\text{Basis},\mathcal{P}^M}(\lambda, u, v) = \chi_H^{\text{Basis},\mathcal{P}^M}(\mu, x, y)$.

We then prove that $T_{\text{JP}} \circ T_{\text{SP2}} \circ T_{\text{Siam}}^\infty \preceq T_{\text{GP}} \circ T_{\text{WL}} \circ T_{\text{D}} \circ T_{\text{SP2}} \circ T_{\text{Siam}}^\infty$. Pick any initial color mapping $\chi^0 \in \mathcal{M}_2^{\mathcal{P}^M}$ and let $\chi = T_{\text{Siam}}^\infty(\chi^0)$. Note that $\chi \equiv T_{\text{Siam}}(\chi)$. We will prove that $(T_{\text{JP}} \circ T_{\text{SP2}})(\chi) \preceq (T_{\text{GP}} \circ T_{\text{WL}} \circ T_{\text{D}} \circ T_{\text{SP2}})(\chi)$. Pick any graphs $G, H \in \mathcal{G}$. We have

$$\begin{aligned}
&[(T_{\text{JP}} \circ T_{\text{SP2}})(\chi)](G) = [(T_{\text{JP}} \circ T_{\text{SP2}})(\chi)](H) \\
\implies & \{\!\!\{\{\!\!\{\chi_G(\lambda, u, v) : \lambda \in \Lambda^M(G)\}\!\!\} : u, v \in V_G\}\!\!\} = \{\!\!\{\{\!\!\{\chi_H(\mu, x, y) : \mu \in \Lambda^M(H)\}\!\!\} : x, y \in V_H\}\!\!\} \\
\implies & \{\!\!\{(\{\!\!\{\chi_G(\lambda, u, v) : \lambda \in \Lambda^M(G)\}\!\!\}, \text{atp}_G(u, v)) : u, v \in V_G\}\!\!\} \\
& = \{\!\!\{(\{\!\!\{\chi_H(\mu, x, y) : \mu \in \Lambda^M(H)\}\!\!\}, \text{atp}_H(x, y)) : x, y \in V_H\}\!\!\} \\
\implies & \{\!\!\{(\{\!\!\{\chi_G(\lambda, v, v) : \lambda \in \Lambda^M(G)\}\!\!\}, \text{atp}_G(u, v)) : u, v \in V_G\}\!\!\} \\
& = \{\!\!\{(\{\!\!\{\chi_H(\mu, y, y) : \mu \in \Lambda^M(H)\}\!\!\}, \text{atp}_H(x, y)) : x, y \in V_H\}\!\!\} \\
\implies & [(T_{\text{GP}} \circ T_{\text{WL}} \circ T_{\text{D}} \circ T_{\text{SP2}})(\chi)](G) = [(T_{\text{GP}} \circ T_{\text{WL}} \circ T_{\text{D}} \circ T_{\text{SP2}})(\chi)](H),
\end{aligned}$$

where the second step is based on Lemma A.5, and the third step is based on the definition of T_{Siam}. This proves that $T_{\text{JP}} \circ T_{\text{SP2}} \circ T_{\text{Siam}}^\infty \preceq T_{\text{GP}} \circ T_{\text{WL}} \circ T_{\text{D}} \circ T_{\text{SP2}} \circ T_{\text{Siam}}^\infty$.

We next prove that $T_{\text{D}} \circ T_{\text{SP2}} \circ T_{\text{Siam}}^\infty \preceq T_{\text{SP1}} \circ T_{\text{BP}} \circ T_{\text{Siam}}^\infty$. Pick any graphs $G, H \in \mathcal{G}$ and vertices $u \in V_G, x \in V_H$. We have

$$\begin{aligned}
&[(T_{\text{D}} \circ T_{\text{SP2}})(\chi)](G) = [(T_{\text{D}} \circ T_{\text{SP2}})(\chi)](H) \\
\implies & \{\!\!\{\chi_G(\lambda, u, u) : \lambda \in \Lambda^M(G)\}\!\!\} = \{\!\!\{\chi_H(\mu, x, x) : \mu \in \Lambda^M(H)\}\!\!\} \\
\implies & \{\!\!\{\{\!\!\{\chi_G(\lambda, u, v) : v \in V_G\}\!\!\} : \lambda \in \Lambda^M(G)\}\!\!\} = \{\!\!\{\{\!\!\{\chi_H(\mu, x, y) : y \in V_H\}\!\!\} : \mu \in \Lambda^M(H)\}\!\!\} \\
\implies & [(T_{\text{SP1}} \circ T_{\text{BP}})(\chi)](G) = [(T_{\text{SP1}} \circ T_{\text{BP}})(\chi)](H),
\end{aligned}$$

where the second step is based on the definition of T_{Siam}. This proves that $T_{\text{D}} \circ T_{\text{SP2}} \circ T_{\text{Siam}}^\infty \preceq T_{\text{SP1}} \circ T_{\text{BP}} \circ T_{\text{Siam}}^\infty$.

We conclude the proof by combining the above relations with Propositions A.1 and A.2. $\qquad\square$

We next turn to the proof of Proposition 6.5, which is almost the same as the case of BasisNet. Below, we will define the equivalent color refinement algorithm for SPE. The initial color mapping associated with SPE is simply $\mathcal{P}^M$. Then, the architecture refines $\mathcal{P}^M$ by using color transformation T_{IGN} defined in Equation (54). The remaining procedure is the same as BasisNet. Combined these together, the color mapping corresponding to the whole algorithm can be written as $(T_{\text{GP}} \circ T_{\text{WL}}^\infty \circ T_{\text{P2}} \circ T_{\text{IGN}}^\infty)(\mathcal{P}^M)$. Then, it suffices to prove the following two equivalence relations: (i) $T_{\text{IGN}}^\infty(\mathcal{P}^M) \equiv (T_{\text{SP2}} \circ T_{\text{SIGN}}^\infty)(\chi^{\mathcal{P}^M})$; (ii) $T_{\text{WL}} \circ T_{\text{P2}} \circ T_{\text{SP2}} \circ T_{\text{SIGN}}^\infty \equiv T_{\text{P2}} \circ T_{\text{SP2}} \circ T_{\text{SIGN}}^\infty$. The proof procedure is alomst the same as in Appendices A.5 and A.6 and we omit it here.

A.7. Discussions with other architectures

Graphormer (Ying et al., 2021), Graphormer-GD (Zhang et al., 2023b), and GraphiT (Mialon et al., 2021). Zhang et al. (2023b) has shown that the expressive power of these architectures is inherently bounded by GD-WL with different

distances. Here, Graphormer uses SPD, Graphormer-GD uses both SPD and RD, while the distance used in GraphiT has the form $d_G(u, v) = \sum_{i=1}^{m} \phi(\lambda_i) \boldsymbol{P}_i(u, v)$, where $\boldsymbol{P}_i$ is the projection matrix associated with eigenvalue λ for graph matrix $\hat{\boldsymbol{L}}$, and $\phi : \mathbb{R} \to \mathbb{R}$ is a general function. Therefore, based on Theorem 5.1 and the proof, it is straightforward to see that all these architectures are bounded by EPWL.

PEG (Wang et al., 2022). Given graph G, PEG maintains a feature vector $\boldsymbol{h}^{(l)}(u) \in \mathbb{R}^d$ for each node $u \in V_G$ in each layer l, and the feature is updated by the following formula:

$$\boldsymbol{h}^{(l+1)}(u) = \psi \left(\sum_{v \in N_G(u)} \phi(\|\boldsymbol{z}_G(u) - \boldsymbol{z}_G(v)\|) \boldsymbol{W} \boldsymbol{h}^{(l)}(v) \right), \tag{74}$$

where $\phi : \mathbb{R} \to \mathbb{R}$ and $\psi : \mathbb{R}^d \to \mathbb{R}^d$ are arbitrary functions, $\boldsymbol{W} \in \mathbb{R}^{d \times d}$ is a parameterized weight matrix, and $\boldsymbol{z}_G(u) \in \mathbb{R}^k$ is the positional encoding corresponding to the top k eigenvectors at node u. Here, we assume that the number k is chosen such that $\|\boldsymbol{z}_G(u) - \boldsymbol{z}_G(v)\|$ is unique for all graphs G of interest (i.e., no ambiguity problem).

We will show that the expressive power of PEG is bounded by EPWL. To obtain this result, note that $\|\boldsymbol{z}_G(u) - \boldsymbol{z}_G(v)\|^2 = \sum_{i=1}^{k} (\boldsymbol{z}_{G,i}(u))^2 + (\boldsymbol{z}_{G,i}(v))^2 - 2\boldsymbol{z}_{G,i}(u)\boldsymbol{z}_{G,i}(v)$. Since $\|\boldsymbol{z}_G(u) - \boldsymbol{z}_G(v)\|$ is unique, the span of top k eigenvectors must be equivalent to the direct sum of the eigenspaces corresponding to top k' eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_{k'}$ for some $k' \leq k$. It follows that $\|\boldsymbol{z}_G(u) - \boldsymbol{z}_G(v)\|^2 = \sum_{i=1}^{k'} \boldsymbol{P}_i(u, u) + \boldsymbol{P}_i(v, v) - 2\boldsymbol{P}_i(u, v)$ where $\boldsymbol{P}_i$ is the projection onto the eigenspace corresponding to eigenvalue λ_i. Therefore, the expressive power of PEG is bounded by the color refinement algorithm $(T_{\mathsf{GP}} \circ T_{\mathsf{PEG}}^{\infty})(\chi^0)$ with χ^0 the initial color mapping and T_{PEG} the color transformation defined below:

$$[T_{\mathsf{PEG}}(\chi)]_G(u) = \mathsf{hash} \left(\{\!\{ (\chi_G(v), \mathcal{P}_G^M(u, u), \mathcal{P}_G^M(v, v), \mathcal{P}_G^M(u, v)) : v \in V_G \}\!\} \right). \tag{75}$$

We will prove that $(T_{\mathsf{GP}} \circ T_{\mathsf{EP},M}^{\infty})(\chi^0) \preceq (T_{\mathsf{GP}} \circ T_{\mathsf{PEG}}^{\infty})(\chi^0)$. Based on Propositions A.1 and A.2, it suffices to prove that $T_{\mathsf{EP},M}^{\infty} \preceq T_{\mathsf{PEG}} \circ T_{\mathsf{EP},M}^{\infty}$. Denote $\chi = T_{\mathsf{EP},M}^{\infty}(\chi^0)$ and note that $\chi \equiv T_{\mathsf{EP},M}^{\infty}(\chi)$. Pick any graphs $G, H \in \mathcal{G}$ and vertices $u \in V_G$, $x \in V_H$. We have

$$\chi_G(u) = \chi_H(x)$$
$$\implies \mathcal{P}_G^M(u, u) = \mathcal{P}_H^M(x, x) \wedge \{\!\{ (\chi_G(v), \mathcal{P}_G^M(u, v)) : v \in V_G \}\!\} = \{\!\{ (\chi_H(y), \mathcal{P}_H^M(x, y)) : y \in V_H \}\!\}$$
$$\implies \{\!\{ (\chi_G(v), \mathcal{P}_G^M(u, u), \mathcal{P}_G^M(v, v), \mathcal{P}_G^M(u, v)) : v \in V_G \}\!\} = \{\!\{ (\chi_H(y), \mathcal{P}_H^M(x, x), \mathcal{P}_H^M(y, y), \mathcal{P}_H^M(x, y)) : y \in V_H \}\!\}$$
$$\implies [T_{\mathsf{PEG}}(\chi)]_G(u) = [T_{\mathsf{PEG}}(\chi)]_H(x),$$

where in the first and second steps we use Proposition A.14. This concludes the proof that the expressive power of PEG is bounded by EPWL.

GIRT (Ma et al., 2023b). Given graph G, GIRT maintains a feature vector for both vertices and vertex pairs. Denote by $\boldsymbol{h}^{(l)}(u) \in \mathbb{R}^d$ the feature of node $u \in V_G$ in layer l, and denote by $\boldsymbol{h}^{(l)}(u, v) \in \mathbb{R}^{d'}$ the feature of node pair $(u, v) \in V_G^2$ in layer l. The features are updated by the following formula:

$$\boldsymbol{h}_G^{(l+1)}(u, v) = \sigma \left(\rho \left((\boldsymbol{W}_{\mathsf{Q}} \boldsymbol{h}_G(u) + \boldsymbol{W}_{\mathsf{K}} \boldsymbol{h}_G(v)) \odot \boldsymbol{W}_{\mathsf{Ew}} \boldsymbol{h}_G^{(l)}(u, v) \right) + \boldsymbol{W}_{\mathsf{Eb}} \boldsymbol{h}_G^{(l)}(u, v) \right), \tag{76}$$

$$\alpha_G^{(l+1)}(u, v) = \mathsf{Softmax}_{j \in V_G}(\boldsymbol{W}_{\mathsf{A}} \boldsymbol{h}_G^{(l+1)}(u, v)), \tag{77}$$

$$\boldsymbol{h}_G^{(l+1)}(u) = \sum_{v \in V_G} \alpha_G^{(l+1)}(u, v) \cdot (\boldsymbol{W}_{\mathsf{V}} \boldsymbol{h}_G^{(l)}(v) + \boldsymbol{W}_{\mathsf{Ev}} \boldsymbol{h}_G^{(l+1)}(u, v)), \tag{78}$$

where the initial feature is defined as

$$\boldsymbol{h}_G^{(0)}(u, v) = [(\boldsymbol{D}^{-1}\boldsymbol{A})_G^0(u, v), (\boldsymbol{D}^{-1}\boldsymbol{A})_G^1(u, v), \cdots, (\boldsymbol{D}^{-1}\boldsymbol{A})_G^K(u, v)],$$
$$\boldsymbol{h}_G^{(0)}(u) = \boldsymbol{h}_G^{(0)}(u, u).$$

One can easily write the corresponding color refinement algorithm that upper bounds of the expressive power of GIRT. Formally, it can be expressed as $(T_{\mathsf{GP}} \circ T_{\mathsf{D}} \circ T_{\mathsf{GIRT}}^{\infty})(\chi^{\mathsf{GIRT}})$, where T_{D} is defined in Equation (72), the initial color mapping χ^{GIRT} is simply the multi-dimensional PageRank distance, and $T_{\mathsf{GIRT}} : \mathcal{M}_2 \to \mathcal{M}_2$ is the color refinement defined below:

$$[T_{\mathsf{GIRT}}(\chi)]_G(u, v) = \begin{cases} \mathsf{hash}(\chi_G(u, v), \chi_G(u, u), \chi_G(v, v)) & \text{if } u \neq v, \\ \mathsf{hash}(\chi_G(u, u), \{\!\{ (\chi_G(u, v), \chi_G(v, v)) : v \in V_G \}\!\}) & \text{if } u = v. \end{cases} \tag{79}$$

We will prove that $(T_{\mathsf{GP}} \circ T_{\mathsf{EP},M}^{\infty})(\chi^0) \preceq (T_{\mathsf{GP}} \circ T_{\mathsf{D}} \circ T_{\mathsf{GIRT}}^{\infty})(\chi^{\mathsf{GIRT}})$. Define color transformation $T_{\times,\mathcal{P}M} : \mathcal{M}_1 \to \mathcal{M}_2$ such that for any color mapping $\chi \in \mathcal{M}_1$ and rooted graph $G^{uv} \in \mathcal{G}_2$,

$$[T_{\times,\mathcal{P}M}(\chi)]_G(u,v) = \mathsf{hash}(\chi_G(u), \chi_G(v), \mathcal{P}_G^M(u,v), \mathcal{P}_G^M(u,u), \mathcal{P}_G^M(v,v)). \tag{80}$$

Note that $T_{\mathsf{GP}} \circ T_{\mathsf{EP},M}^{\infty} \equiv T_{\mathsf{GP}} \circ T_{\mathsf{D}} \circ T_{\times,\mathcal{P}M} \circ T_{\mathsf{EP},M}^{\infty}$ by Proposition A.14. Also, $T_{\times,\mathcal{P}M}(\chi^0) \preceq \chi^{\mathsf{GIRT}}$ due to Lemma A.16. Therefore, it suffices to prove that $T_{\times,\mathcal{P}M} \circ T_{\mathsf{EP},M}^{\infty} \preceq T_{\mathsf{GIRT}}^{\infty} \circ T_{\times,\mathcal{P}M}$. Based on Proposition A.2, it suffices to prove that $T_{\mathsf{GIRT}} \circ T_{\times,\mathcal{P}M} \circ T_{\mathsf{EP},M}^{\infty} \equiv T_{\times,\mathcal{P}M} \circ T_{\mathsf{EP},M}^{\infty}$. Denote $\chi = T_{\mathsf{EP},M}^{\infty}(\chi^0)$, where $\chi^0 \in \mathcal{M}_1$ is any initial color mapping. Pick any graphs $G, H \in \mathcal{G}$ and vertices $u, v \in V_G, x, y \in V_H$ such that $[T_{\times,\mathcal{P}M}(\chi)]_G(u,v) = [T_{\times,\mathcal{P}M}(\chi)]_H(x,y)$. We have:

- If $u = v$, then $x = y$.

$$[T_{\times,\mathcal{P}M}(\chi)]_G(u,u) = [T_{\times,\mathcal{P}M}(\chi)]_H(x,x)$$
$$\implies \chi_G(u) = \chi_H(x), \mathcal{P}_G^M(u,u) = \mathcal{P}_H^M(x,x)$$
$$\implies \{\!\{(\chi_G(u), \chi_G(v), \mathcal{P}_G^M(u,v)) : v \in V_G\}\!\} = \{\!\{(\chi_H(x), \chi_H(y), \mathcal{P}_H^M(x,y)) : y \in V_H\}\!\}$$
$$\implies \{\!\{(\chi_G(u), \mathcal{P}_G^M(u,u), \chi_G(v), \mathcal{P}_G^M(v,v), \mathcal{P}_G^M(u,v)) : v \in V_G\}\!\}$$
$$= \{\!\{(\chi_H(x), \mathcal{P}_H^M(x,x), \chi_H(y), \mathcal{P}_H^M(y,y), \mathcal{P}_H^M(x,y)) : y \in V_H\}\!\}$$
$$\implies \{\!\{([T_{\times,\mathcal{P}M}(\chi)]_G(u,v), [T_{\times,\mathcal{P}M}(\chi)]_G(v,v)) : v \in V_G\}\!\}$$
$$= \{\!\{([T_{\times,\mathcal{P}M}(\chi)]_H(x,y), [T_{\times,\mathcal{P}M}(\chi)]_H(y,y)) : y \in V_H\}\!\}$$
$$\implies [(T_{\mathsf{GIRT}} \circ T_{\times,\mathcal{P}M})(\chi)]_G(u,u) = [(T_{\mathsf{GIRT}} \circ T_{\times,\mathcal{P}M})(\chi)]_G(x,x).$$

- If $u \neq v$, then $x \neq y$.

$$[T_{\times,\mathcal{P}M}(\chi)]_G(u,v) = [T_{\times,\mathcal{P}M}(\chi)]_H(x,y)$$
$$\implies \chi_G(u) = \chi_H(x), \chi_G(v) = \chi_H(y), \mathcal{P}_G^M(u,v) = \mathcal{P}_H^M(x,y), \mathcal{P}_G^M(u,u) = \mathcal{P}_H^M(x,x), \mathcal{P}_G^M(v,v) = \mathcal{P}_H^M(y,y)$$
$$\implies [(T_{\mathsf{GIRT}} \circ T_{\times,\mathcal{P}M})(\chi)]_G(u,v) = [(T_{\mathsf{GIRT}} \circ T_{\times,\mathcal{P}M})(\chi)]_G(x,y).$$

Here, in the above derivations we use Proposition A.14. We have concluded the proof that the expressive power of GIRT is bounded by EPWL.

Spectral PPGN and Spectral k-IGN. Based on Maron et al. (2019a), PPGN can mimic the 2-FWL test (Cai et al., 1992), and Spectral k-IGN can mimic the k-WL test (Grohe, 2017). Let $T_{\mathsf{WL}(k)} : \mathcal{M}_k \to \mathcal{M}_k$ and $T_{\mathsf{FWL}(k)} : \mathcal{M}_k \to \mathcal{M}_k$ be the color refinements associated with k-WL and k-FWL, respectively, and let $\chi_k^0 \in \mathcal{M}_k$ be the initial color mapping in k-WL and k-FWL. The color refinement algorithms corresponding to k-WL and k-FWL can then be described as $(T_{\mathsf{JP}(k)} \circ T_{\mathsf{WL}(k)}^{\infty})(\chi_k^0)$ and $(T_{\mathsf{JP}(k)} \circ T_{\mathsf{FWL}(k)}^{\infty})(\chi_k^0)$, respectively, where $T_{\mathsf{JP}(k)}$ is defined in Equation (84). Maron et al. (2019a) proved that with sufficiently layers, the features of vertex k-tuples computed by k-IGN is finer than $T_{\mathsf{WL}(k)}^{\infty}(\chi_k^0)$, and the features of vertex pairs computed by PPGN is finer than $T_{\mathsf{FWL}(2)}^{\infty}(\chi_2^0)$. Later, Azizian et al. (2021) proved that the features of vertex pairs computed by PPGN is also bounded by (and thus as fine as) $T_{\mathsf{FWL}(2)}^{\infty}(\chi_2^0)$ (see Lemma 12 in their paper). Finally, Geerts & Reutter (2022) proved that the features of vertex k-tuples computed by k-IGN is bounded by (and thus as fine as) $T_{\mathsf{WL}(k)}^{\infty}(\chi_k^0)$ (see Lemma E.1 in their paper).

We now define the color refinement algorithms for Spectral PPGN and Spectral k-IGN, which are as expressive as the corresponding GNN architectures based on the results of Maron et al. (2019a); Azizian et al. (2021); Geerts & Reutter (2022). First define the following color transformations:

- **Spectral k-IGN color refinement.** Define $T_{\mathsf{SIGN}(k)} : \mathcal{M}_k^{\mathcal{P}M} \to \mathcal{M}_k^{\mathcal{P}M}$ such that for any color mapping $\chi \in \mathcal{M}_k^{\mathcal{P}M}$ and $(G^{\boldsymbol{u}}, \lambda) \in \mathcal{G}_k^{\mathcal{P}M}$,

$$[T_{\mathsf{SIGN}(k)}(\chi)]_G(\lambda, \boldsymbol{u}) = \mathsf{hash}([T_{\mathsf{WL}(k)}(\chi(\lambda, \cdots))]_G(\boldsymbol{u}), [T_{\mathsf{WL}(k)}(T_{\mathsf{SP}(k)}(\chi))]_G(\boldsymbol{u})). \tag{81}$$

- **Spectral PPGN color refinement.** Define $T_{\mathsf{PPGN}} : \mathcal{M}_2^{\mathcal{P}M} \to \mathcal{M}_2^{\mathcal{P}M}$ such that for any color mapping $\chi \in \mathcal{M}_2^{\mathcal{P}M}$ and $(G^{\boldsymbol{u}}, \lambda) \in \mathcal{G}_2^{\mathcal{P}M}$,

$$[T_{\mathsf{SPPGN}}(\chi)]_G(\lambda, \boldsymbol{u}) = \mathsf{hash}([T_{\mathsf{FWL}(2)}(\chi(\lambda, \cdots))]_G(\boldsymbol{u}), [T_{\mathsf{FWL}(2)}(T_{\mathsf{SP}(2)}(\chi))]_G(\boldsymbol{u})). \tag{82}$$

- **Spectral pooling.** Define $T_{\mathsf{SP}(k)} : \mathcal{M}_k^{\mathcal{P}^M} \to \mathcal{M}_k$ such that for any color mapping $\chi \in \mathcal{M}_k^{\mathcal{P}^M}$ and $G^{\boldsymbol{u}} \in \mathcal{G}_k$,

$$[T_{\mathsf{SP}(k)}(\chi)]_G(\boldsymbol{u}) = \mathsf{hash}(\{\!\{\chi_G(\lambda, \boldsymbol{u}) : \lambda \in \Lambda^M(G)\}\!\}). \tag{83}$$

- **Joint pooling.** Define $T_{\mathsf{JP}(k)} : \mathcal{M}_k \to \mathcal{M}_0$ such that for any color mapping $\chi \in \mathcal{M}_k$ and $G \in \mathcal{G}$,

$$[T_{\mathsf{JP}(k)}(\chi)](G) = \mathsf{hash}(\{\!\{\chi_G(\boldsymbol{u}) : \boldsymbol{u} \in V_G^k\}\!\}). \tag{84}$$

Define the initial color mapping $\chi_k^{\mathcal{P}^M} \in \mathcal{M}_k^{\mathcal{P}^M}$ such that for any graph G, vertices $\boldsymbol{u} \in V_G^k$, and $\lambda \in \Lambda^M(G)$,

$$[\chi_k^{\mathcal{P}^M}]_G(\lambda, \boldsymbol{u}) = \mathsf{hash}(\lambda, [\boldsymbol{P}_\lambda^M]_G(u_1, u_1), \cdots, [\boldsymbol{P}_\lambda^M]_G(u_1, u_k), \cdots, [\boldsymbol{P}_\lambda^M]_G(u_k, u_1), \cdots, [\boldsymbol{P}_\lambda^M]_G(u_k, u_k)), \tag{85}$$

where $\boldsymbol{P}_\lambda^M$ is the projection onto eigenspace associated with eigenvalue λ for graph matrix $\boldsymbol{M}$. The color refinement algorithm for Spectral PPGN is then defined as $(T_{\mathsf{JP}(2)} \circ T_{\mathsf{SP}(2)} \circ T_{\mathsf{SPPGN}}^\infty)(\chi_2^{\mathcal{P}^M})$. Similarly, the color refinement algorithm for Spectral k-IGN is then defined as $(T_{\mathsf{JP}(k)} \circ T_{\mathsf{SP}(k)} \circ T_{\mathsf{SIGN}(k)}^\infty)(\chi_2^{\mathcal{P}^M})$. We aim to prove that following two results:

Proposition A.28. $(T_{\mathsf{JP}(2)} \circ T_{\mathsf{SP}(2)} \circ T_{\mathsf{SPPGN}}^\infty)(\chi_2^{\mathcal{P}^M}) \equiv (T_{\mathsf{JP}(2)} \circ T_{\mathsf{FWL}(2)}^\infty)(\chi_2^0)$.

Proposition A.29. $(T_{\mathsf{JP}(k)} \circ T_{\mathsf{SP}(k)} \circ T_{\mathsf{SIGN}(k)}^\infty)(\chi_k^{\mathcal{P}^M}) \equiv (T_{\mathsf{JP}(k)} \circ T_{\mathsf{WL}(k)}^\infty)(\chi_k^0)$.

We will only prove Proposition A.28, as the proof of Proposition A.29 is almost the same.

Proof of Proposition A.28. We first prove that $(T_{\mathsf{JP}(2)} \circ T_{\mathsf{SP}(2)} \circ T_{\mathsf{SPPGN}}^\infty)(\chi_2^{\mathcal{P}^M}) \preceq (T_{\mathsf{JP}(2)} \circ T_{\mathsf{FWL}(2)}^\infty)(\chi_2^0)$. Since $T_{\mathsf{SP}(2)}(\chi_2^{\mathcal{P}^M}) \preceq \chi_2^0$ (Lemma A.5), it suffices to prove that $T_{\mathsf{SP}(2)} \circ T_{\mathsf{SPPGN}}^\infty \preceq T_{\mathsf{FWL}(2)}^\infty \circ T_{\mathsf{SP}(2)}$. Based on Proposition A.2, it suffices to prove that $T_{\mathsf{FWL}(2)} \circ T_{\mathsf{SP}(2)} \circ T_{\mathsf{SPPGN}}^\infty \equiv T_{\mathsf{SP}(2)} \circ T_{\mathsf{SPPGN}}^\infty$. Denote $\chi = T_{\mathsf{SPPGN}}^\infty(\hat{\chi}^0)$, where $\hat{\chi}^0 \in \mathcal{M}_2^{\mathcal{P}^M}$ is any initial color mapping. Pick any graphs $G, H \in \mathcal{G}$ and vertices $u, v \in V_G$, $x, y \in V_H$ such that $[T_{\mathsf{SP}(2)}(\chi)]_G(u, v) = [T_{\mathsf{SP}(2)}(\chi)]_H(x, y)$. We have:

$$[T_{\mathsf{SP}(2)}(\chi)]_G(u, v) = [T_{\mathsf{SP}(2)}(\chi)]_H(x, y)$$
$$\implies \{\!\{\chi_G(\lambda, u, v) : \lambda \in \Lambda^M(G)\}\!\} = \{\!\{\chi_H(\mu, x, y) : \mu \in \Lambda^M(H)\}\!\}$$
$$\implies \{\!\{[(T_{\mathsf{FWL}(2)} \circ T_{\mathsf{SP}(2)})(\chi)]_G(u, v) : \lambda \in \Lambda^M(G)\}\!\} = \{\!\{[(T_{\mathsf{FWL}(2)} \circ T_{\mathsf{SP}(2)})(\chi)]_H(x, y) : \mu \in \Lambda^M(H)\}\!\}$$
$$\implies [(T_{\mathsf{FWL}(2)} \circ T_{\mathsf{SP}(2)})(\chi)]_G(u, v) = [(T_{\mathsf{FWL}(2)} \circ T_{\mathsf{SP}(2)})(\chi)]_H(x, y).$$

where in the second step we use the definition of T_{SPPGN} and the fact that $\chi \equiv T_{\mathsf{SPPGN}}(\chi)$.

We next prove that $(T_{\mathsf{JP}(2)} \circ T_{\mathsf{FWL}(2)}^\infty)(\chi_2^0) \preceq (T_{\mathsf{JP}(2)} \circ T_{\mathsf{SP}(2)} \circ T_{\mathsf{SPPGN}}^\infty)(\chi_2^{\mathcal{P}^M})$. Based on Zhang et al. (2023a), we have $T_{\mathsf{FWL}(2)}^\infty \preceq T_{\mathsf{PS}}^\infty \circ T_{\mathsf{NM}}$ and $T_{\mathsf{PS}}^\infty \equiv T_{\mathsf{Du}} \circ T_{\mathsf{Dv}} \circ T_{\mathsf{PS}}^\infty$, where $T_{\mathsf{PS}}, T_{\mathsf{NM}}, T_{\mathsf{Du}}, T_{\mathsf{Dv}}$ are defined in Appendix A.3. On the other hand, we have proved in Corollary A.11 that $(T_{\mathsf{PS}}^\infty \circ T_{\mathsf{NM}})(\chi_2^0) \preceq \mathcal{P}^M$. Therefore, $(T_{\mathsf{PS}}^\infty \circ T_{\mathsf{NM}})(\chi_2^0) \preceq (T_{\mathsf{Du}} \circ T_{\mathsf{Dv}})(\mathcal{P}^M) \preceq T_{\mathsf{SP}(2)}(\chi_2^{\mathcal{P}^M})$. This finally implies that $T_{\mathsf{FWL}(2)}^\infty(\chi_2^0) \preceq T_{\mathsf{SP}(2)}(\chi_2^{\mathcal{P}^M})$.

It thus suffices to prove that $(T_{\mathsf{FWL}(2)}^\infty \circ T_{\mathsf{SP}(2)})(\chi_2^{\mathcal{P}^M}) \preceq (T_{\mathsf{SP}(2)} \circ T_{\mathsf{SPPGN}}^\infty)(\chi_2^{\mathcal{P}^M})$. We will prove the following stronger result: for any $t \geq 0$, $(T_{\mathsf{FWL}(2)}^\infty \circ T_{\mathsf{SP}(2)})(\chi_2^{\mathcal{P}^M}) \preceq (T_{\mathsf{SP}(2)} \circ T_{\mathsf{SPPGN}}^t)(\chi_2^{\mathcal{P}^M})$. The proof is based on induction. For the base case of $t = 0$, the result clearly holds. Now assume that the result holds for $t = t'$ and consider the case of $t = t' + 1$. Denote $\chi = T_{\mathsf{SPPGN}}^t(\chi_2^{\mathcal{P}^M})$. Pick any graphs $G, H \in \mathcal{G}$ and vertices $u, v \in V_G$, $x, y \in V_H$. We have

$$[(T_{\mathsf{FWL}(2)}^\infty \circ T_{\mathsf{SP}(2)})(\chi_2^{\mathcal{P}^M}))]_G(u, v) = [(T_{\mathsf{FWL}(2)}^\infty \circ T_{\mathsf{SP}(2)})(\chi_2^{\mathcal{P}^M}))]_H(x, y)$$
$$\implies [(T_{\mathsf{FWL}(2)} \circ T_{\mathsf{FWL}(2)}^\infty \circ T_{\mathsf{SP}(2)})(\chi_2^{\mathcal{P}^M}))]_G(u, v) = [(T_{\mathsf{FWL}(2)} \circ T_{\mathsf{FWL}(2)}^\infty \circ T_{\mathsf{SP}(2)})(\chi_2^{\mathcal{P}^M}))]_H(x, y)$$
$$\implies [(T_{\mathsf{FWL}(2)} \circ T_{\mathsf{SP}(2)})(\chi)]_G(u, v) = [(T_{\mathsf{FWL}(2)} \circ T_{\mathsf{SP}(2)})(\chi)]_H(x, y)$$
$$\implies \Lambda^M(G) = \Lambda^M(H) \wedge [(T_{\mathsf{FWL}(2)}(\chi(\lambda, \cdots))]_G(u, v) = [(T_{\mathsf{FWL}(2)}(\chi(\lambda, \cdots))]_H(x, y) \quad \forall \lambda \in \Lambda^M(G).$$

Combining the last two steps implies that $[(T_{\mathsf{SP}(2)} \circ T_{\mathsf{SPPGN}})(\chi)]_G(u, v) = [(T_{\mathsf{SP}(2)} \circ T_{\mathsf{SPPGN}})(\chi)]_H(x, y)$. We thus conclude the induction step. $\qquad\square$

A.8. Counterexamples

In this subsection, we aim to reveal the expressivity gaps between different GNN models. This is achieved by constructing a pair of counterexample graphs $G, H \in \mathcal{G}$ such that one GNN can distinguish while the other cannot. Here, we will leverage an important theoretical tool to construct counterexamples, known as Fürer graphs Fürer (2001). An in-depth introduction of Fürer graphs can be found in Zhang et al. (2023a).

Definition A.30 (Fürer graphs). Given any connected graph $F = (V_F, E_F)$, the Fürer graph $G(F) = (V_{G(F)}, E_{G(F)})$ is constructed as follows:

$$V_{G(F)} = \{(x, X) : x \in V_F, X \subset N_F(x), |X| \bmod 2 = 0\},$$
$$E_{G(F)} = \{\{(x, X), (y, Y)\} \subset V_G : \{x, y\} \in E_F, (x \in Y \leftrightarrow y \in X)\}.$$

Here, $x \in Y \leftrightarrow y \in X$ holds when either ($x \in Y$ and $y \in X$) or ($x \notin Y$ and $y \notin X$) holds. For each $x \in V_F$, denote the set

$$\mathsf{Meta}_F(x) := \{(x, X) : X \subset N_F(x), |X| \bmod 2 = 0\}, \tag{86}$$

which is called the meta vertices of $G(F)$ associated to x. Note that $V_{G(F)} = \bigcup_{x \in V_F} \mathsf{Meta}_F(x)$.

We next define an operation called "twist":

Definition A.31 (Twisted Fürer graphs). Let $G(F) = (V_{G(F)}, E_{G(F)})$ be the Fürer graph of $F = (V_F, E_F)$, and let $\{x, y\} \in E_F$ be an edge of F. The *twisted* Fürer graph of $G(F)$ for edge $\{x, y\}$, is constructed as follows: $\mathsf{twist}(G(F), \{x, y\}) := (V_{G(F)}, E_{\mathsf{twist}(G(F), \{x, y\})})$, where

$$E_{\mathsf{twist}(G(F), \{x, y\})} := E_{G(F)} \triangle \{\{\xi, \eta\} : \xi \in \mathsf{Meta}_F(x), \eta \in \mathsf{Meta}_F(y)\},$$

and $\triangle$ is the symmetric difference operator, i.e., $A \triangle B = (A \backslash B) \cup (B \backslash A)$. For an edge set $S = \{e_1, \cdots, e_k\} \subset E_F$, we further define

$$\mathsf{twist}(G(F), S) := \mathsf{twist}(\cdots \mathsf{twist}(G(F), e_1) \cdots, e_k). \tag{87}$$

Note that Equation (87) is well-defined as the resulting graph does not depend on the order of edges $e_1, \cdots, e_k$ for twisting.

The following result is well-known (see e.g., Zhang et al., 2023a, Corollary I.5 and Lemma I.7)):

Theorem A.32. *For any graph F and any set $S_1, S_2 \subset E_F$, $\mathsf{twist}(G(F), S_1) \simeq \mathsf{twist}(G(F), S_2)$ iff $|S_1| \bmod 2 = |S_2| \bmod 2$.*

Below, we will prove Proposition A.33 using Fürer graphs. Note that it can be easily checked via a computer program whether a pair of graphs can be distinguished by a given color refinement algorithm. However, an in-depth understanding of why a given color refinement algorithm can/cannot distinguish these (twisted) Fürer graphs is beyond the scope of this paper and is left for future work.

Proposition A.33. *The following hold:*

a) *There exists a pair of graphs G, H such that SWL cannot distinguish them but Siamese IGN, Spectral IGN, and EPWL with any graph matrix $M \in \{A, L, \hat{L}\}$ can distinguish them;*

b) *For any $M \in \{A, L, \hat{L}\}$, there exists a pair of graphs G, H such that Weak Spectral IGN cannot distinguish them but Spectral IGN can distinguish them;*

c) *There exists a pair of graphs G, H such that Weak Spectral IGN cannot distinguish them with any $M \in \{A, L, \hat{L}\}$, but GD-WL with any distance listed in Section 5 can distinguish them;*

d) *There exists a pair of graphs G, H such that Spectral IGN and EPWL with any graph matrix $M \in \{A, L, \hat{L}\}$ cannot distinguish them, but SWL can distinguish them.*

Proof. For Proposition A.33(a,b,c,d), the counterexample graphs are the Fürer graph and twisted Fürer graph for base graph F defined in Figure 3(a,b,c,d), respectively. □
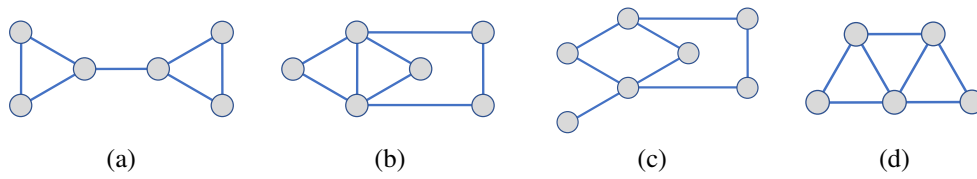
Figure 3. Illustrations of base graphs used to construct Fürer graph and twisted Fürer graph for proving Proposition A.33.