THE PWLR GRAPH REPRESENTATION: A PERSISTENT WEISFEILER-LEHMAN SCHEME WITH RANDOM WALKS FOR GRAPH CLASSIFICATION

Sun Woo Park[*], **Yun Young Choi**[*], **Dosang Joe & Youngho Woo**[†]
Division of Industrial Mathematics
National Institute for Mathematical Sciences
Daejeon, Republic of Korea
{spark483, choi930121, dosjoe, youngw}@nims.re.kr

U Jin Choi
Department of Mathematical Sciences
KAIST
Daejeon, Republic of Korea
{ujchoi}@kaist.ac.kr

ABSTRACT

This paper presents the Persistent Weisfeiler-Lehman Random walk scheme (abbreviated as PWLR) for graph representations, a novel mathematical framework which produces a collection of explainable low-dimensional representations of graphs with discrete and continuous node features. The proposed scheme effectively incorporates normalized Weisfeiler-Lehman procedure, random walks on graphs, and persistent homology. We thereby integrate three distinct properties of graphs, which are local topological features, node degrees, and global topological invariants, while preserving stability from graph perturbations. This generalizes many variants of Weisfeiler-Lehman procedures, which are primarily used to embed graphs with discrete node labels. Empirical results suggest that these representations can be efficiently utilized to produce comparable results to state-of-the-art techniques in classifying graphs with discrete node labels, and enhanced performances in classifying those with continuous node features.

1 INTRODUCTION

Non-Euclidean data structures are crucial subjects of researches, ranging from interactions among protein structures to social networking systems Stokes et al. (2020); Senior et al. (2020). Graphs are commonly utilized for modeling the innate properties of a wide class of these non-Euclidean structures Baek et al. (2021). A number of strategies producing state-of-the-art results in analyzing the properties of graphs include graph neural networks (GNN) or message passing neural networks (MPNN) Kipf & Welling (2017), graph kernels Vishwanathan et al. (2010), Weisfeiler-Lehman procedures Shervashidze et al. (2011); Weisfeiler & Lehman (1968), random walks Nikolentzos & Vazirgiannis (2020); Zhang et al. (2018b); Lovasz (1993), and persistent homological techniques Rieck et al. (2019); Carriere et al. (2020); Edelsbrunner & Harer (2010). These techniques are often employed to obtain graph representations suitable for graph classifications, which aim to classify innate properties of graphs by detecting their structural differences. The varying topological structures of graphs, however, make the task of obtaining a consistent graph representation demanding.

[Related Works] The Weisfeiler-Lehman (WL) isomorphism test measures similarities among graphs with discrete labels by updating the coloring of nodes, each of which represents a depth k unfolding tree Weisfeiler & Lehman (1968). Shervashidze et al. implemented the test in the form of graph kernels, producing state-of-the-art results in classifying graph data sets Vishwanathan et al.

[*] Equal contribution as co-first authors
[†] Corresponding author. Email: youngw@nims.re.kr

(2010); Shervashidze et al. (2011). Various literature focused on generalizing the WL procedure that allows one to represent graphs with continuous node attributes Togninalli et al. (2019); Bodnar et al. (2021), and incorporate global topological invariants Morris et al. (2017); Rieck et al. (2019).

Utilizing stochastic processes on graphs, such as random walks (RW), is an alternative approach to embed graphs to real vector spaces. A random walk on a graph is a random course of travel along the nodes of G characterized by iterative processes of starting from a node and randomly choosing an adjacent node, or itself, to travel to. A number of state-of-the-art approaches include graph kernels using return probabilities of RW Zhang et al. (2018b), and comparing the number of common random walks Borgwardt & Kriegel (2005); Sugiyama & Borgwardt (2015).

Persistent homological techniques are known to be effective for computing the global topological invariants of data sets Carlsson (2009); Edelsbrunner & Harer (2010); Hofer et al. (2017). One determines height functions over nodes and edges of G to construct persistence diagrams, which capture the homological properties of G. The encapsulated properties depend on which features of graphs the predetermined height function utilizes. These features include node labels Rieck et al. (2019), and spectral decompositions of the adjacency matrix Carriere et al. (2020).

[Motivation] The motivation for this project originates from instability and non-optimal dimensionality of representations obtained from pre-existing approaches. Variants of GNNs do not necessarily guarantee representation stability with respect to graph perturbations, where a pair of graphs with similar topological structures may be embedded to a pair of vectors whose distance between them may be arbitrarily large Xu et al. (2019). The WL procedure also substantially increases the dimensionality of representations as the number of iterations of the procedure increase Shervashidze et al. (2011). While graph kernels effectively bound these dimensions, they produce representations that depend on the choice of training data sets Kashima et al. (2003); Vishwanathan et al. (2010).

To address these limitations, we propose the Persistent Weisfeiler-Lehman Random Walk embedding framework (PWLR), a novel graph embedding formalism which produces a collection of low-dimensional representations of graphs while preserving stability from graph perturbations. Inspired from WL Shervashidze et al. (2011) and PWL Rieck et al. (2019) procedures, the PWLR framework captures local topological features, node degrees, and global topological invariants of graphs by effectively incorporating the normalized WL procedure, random walks, and persistent homological approaches. Experimental results suggest that the proposed algorithm produces comparable results to state-of-the-art techniques in classifying graphs with discrete node labels, and enhanced performances in classifying those with continuous node attributes or edge weights.

[Contributions] The novelty of the PWLR embedding framework can be summarized as follows.

• A mathematical framework incorporating **three distinct topological properties** (Theorem 2.1)

• Effective **low-dimensional** representations for classifying graphs with discrete and **continuous features**. (Table 2, 3)

• **Stability** of representations with respect to graph perturbations (Theorem 2.6)

2 Persistent Weisfeiler-Lehman Random Walk Graph Representation

[Algorithm] The PWLR scheme obtains a vector representation of a finite graph G as follows: [1] [2]

$$\varphi_{\text{PWLR}}(G) := \varphi\left(\left(M_G^{k_1}X\right)^T M_G^{k_2}\right). \tag{1}$$

We note that k_1, k_2 are positive integers, X is a $|V| \times l$ matrix consisting of a concatenation of node labels of G, M_G is the normalized weighted adjacency matrix of dimension $|V| \times |V|$ given by

$$M_G := (D+I)^{-1}(A+I), \tag{2}$$

$(\cdot)^T$ is the transpose of a matrix, and φ is the Euclidean embedding obtained from persistent homological features utilizing the updated node labels $(M_G^{k_1}X)^T M_G^{k_2}$. The matrix M_G is the normalized

[1] Appendix provided at: https://arxiv.org/abs/2208.13427

[2] Github repository: https://github.com/spark483/The-PWLR-graph-representation

Algorithm 1 PWLR Embedding Framework

```
**Input:** Graph $G = (V, E)$, iterations $(k_1, k_2)$, $p = 1$, $M_G := (D + I)^{-1}(A + I)$, $X := \{L(v_i)\}_{v_i \in V}$.
Initialize $\overline{D}_e := \{\overline{d}_e := (\overline{d}_v, \overline{d}_w) \mid e = (v, w) \in E\}$.  ($\overline{d}_v$ is the unweighted degree at $v$.)
Set $\varphi_{\text{H0}}, \varphi_{\text{H1}} := \emptyset$,  $\varphi_{\text{H0,Opt}}, \varphi_{\text{H1,Opt}} := [0, \cdots, 0]$ (of length $|\overline{D}|$).
**Normalized WL:** Update $X \leftarrow M_G^{k_1} X$
**Random Walk:** Update $X \leftarrow X^T M_G^{k_2}$
**Persistent homological embedding:** Define $h_E((v_1, v_2)) := \|X(v_1) - X(v_2)\|_p$
Obtain a set $\tilde{E}$ by sorting the set $E(G)$ using $h_E$: $\tilde{E} := \{e_i \in E(G) \mid h_E(e_i) \leq h_E(e_j) \text{ if } i \leq j\}$.
**for** $i \in \{1, \cdots, |\tilde{E}|\}$ **do**
  Define $E^{[i]} := \{e_j \in \tilde{E} \mid j \leq i\}$,  Initialize $G^{[i-1]} := (V, E^{[i-1]})$, $G^{[i]} := (V, E^{[i]})$.
  Pick $e_i \in E^{[i]} \setminus E^{[i-1]}$ and compute $\overline{d}_{e_i} := (\overline{d}_{v_1^i}, \overline{d}_{v_2^i})$.
  **if** # Comp. $G^{[i]}$ − # Comp. $G^{[i-1]} > 0$ **then**
    $\varphi_{\text{H0}} \leftarrow \text{Concat}(\varphi_{\text{H0}}, [h_E(e_i) + 1])$,  $\varphi_{\text{H0,Opt}}(\overline{d}_{e_i}) \leftarrow \varphi_{\text{H0,Opt}}(\overline{d}_{e_i}) + (h_E(e_i) + 1)$
  **end if**
  **if** # Cycle $G^{[i]}$ − # Cycle $G^{[i-1]} > 0$ **then**
    $\varphi_{\text{H1}} \leftarrow \text{Concat}(\varphi_{\text{H1}}, [h_E(e_i)] + 1)$,  $\varphi_{\text{H1,Opt}}(\overline{d}_{e_i}) \leftarrow \varphi_{\text{H1,Opt}}(\overline{d}_{e_i}) + (h_E(e_i) + 1)$
  **end if**
**end for**
**Return:** $\varphi_{\text{H0}}, \varphi_{\text{H1}}, \varphi_{\text{H0,Opt}}, \varphi_{\text{H1,Opt}}$.
```

adjacency convolutional operator used in message passing neural networks (MPNN) Kipf & Welling (2017); Chen et al. (2020); Velickovic et al. (2018). Theorem 2.1 provides a list of correspondences among each component of (1) and the utilized algorithms for analyzing finite graphs, the proof of which is in Appendix E.2.

Theorem 2.1. *Given a finite undirected graph $G = (V, E)$ without self-loops, the PWLR procedure incorporates three disjoint topological properties of G.*

• *The component $M_G^{k_1} \times (\cdot)$ (**WL**) incorporates local topological properties by representing depth k_1 unfolding trees with fixed vertices.*

• *The component $(\cdot) \times M_G^{k_2}$ (**R**) incorporates node degrees with local topological properties.*

• *Lastly, the component $\varphi(\cdot)$ (**P**) incorporates homological invariants of G, i.e. the connected components and cycles.*

The parameters k_1 and k_2, each corresponding to the number of iterations of the WL or RW procedure, provide a collection of adaptable graph representations suitable for classification tasks. They determine the extent of incorporating node attributes and node degrees in constructing graph representations. The map φ incorporates global topological characteristics by constructing persistence diagrams from preset height functions, as will be shown in (13, 15).

Each property mentioned in Theorem 2.1 is known to be crucial for distinguishing isomorphism classes of graphs. The WL procedure and its variants are well-known for their state-of-the-art results in classifying graphs Shervashidze et al. (2011); Togninalli et al. (2019). Geerts, Mazowiecki, and Peréz proved that incorporating node degrees enhances conventional message passing neural networks (MPNN) in distinguishing nodes based on their attributes Geerts et al. (2021). In addition, persistent homological algorithms are known to be effective for enhancing the accuracy of graph classification algorithms Rieck et al. (2019); Carriere et al. (2020). Interested readers may refer to Appendix D for expositions on random walks and persistent homology. A pseudo-code and a diagram summarizing the PWLR embedding scheme are outlined in Algorithm 1 and Figure 1. Explicit computations on how the PWLR algorithm constructs graph representations can be found in Appendix C.

[Normalized WL procedure] We analyze the correspondence between the operator $M_G^{k_1} \times (\cdot)$ and the node labels obtained from the normalized WL procedure. We first define the normalized WL procedure, obtained from normalizing a node label updated from a WL procedure by its length.

Definition 2.2 (Normalized WL procedure)**.** Let $G := (V, E)$ be a graph with node attributes $L : V \to \mathbb{R}^l_{>0}$. Given a node $v \in V$, denote by $N(v)$ the set of nodes which are adjacent to v, including v itself. Let $\| \cdot \|_1$ be the l_1-norm of a node label, and $\tilde{a}_{v,w}$ be the weight on edges $a_{v,w}$ from node v to w, with $\tilde{a}_{v,v} = 1$. The normalized WL procedure updates the attribute of v by

$$L(v) \mapsto \frac{\sum_{w \in N(v)} \tilde{a}_{v,w} L(w)}{\sum_{w \in N(v)} \tilde{a}_{v,w} \|L(w)\|_1}. \tag{3}$$

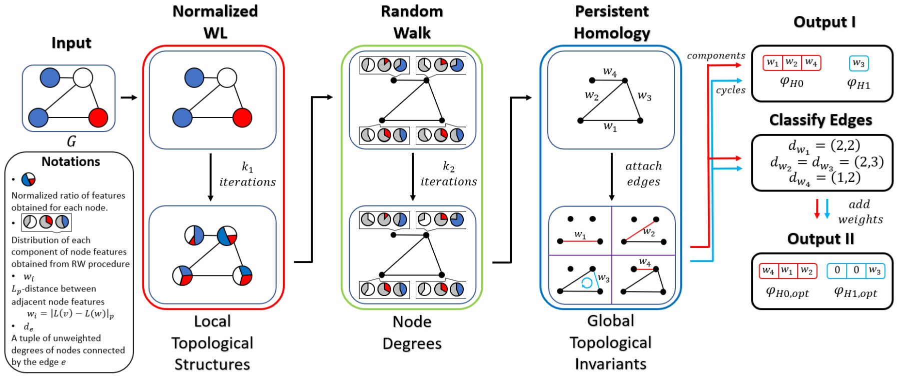

Figure 1: A diagram outlining the architecture of the PWLR embedding scheme

If we further assume that the l_1-norms of all node attributes are equal to 1, then we immediately obtain the following correspondence stated in Proposition 2.3. The proposition can be applied to graphs with discrete node labels, where each node label is embedded to real vector spaces using the one-hot encoder, see Appendix C for instance.

Proposition 2.3. *Suppose that for every node $v \in V(G)$, the l_1 norm of node labels is equal to 1, i.e. $\|L(v)\|_1 = 1$. Then the right multiplication $M_G^{k_1} \times (\cdot)$ is equivalent to k_1 iterations of normalized WL procedure applied to G.*

Using the above proposition, the over-smoothening phenomenon of WL procedure can be reinterpreted by using the eigenvectors of M_G. Because M_G is a stochastic matrix (a square matrix whose sum of entries is equal to 1 for each row), a vector ν whose entries are all equal to 1 is the right eigenvector of M_G with eigenvalue 1. Denote by π_G the left eigenvector of M_G with eigenvalue 1, whose entries will be computed in Theorem 2.5. The limiting behavior of $M_G^k X$ for sufficiently large k can be obtained from the Perron-Frobenius theorem (See Theorem Appendix D.1):

$$\lim_{k\to\infty} M_G^k X = \nu \pi_G^T X \tag{4}$$

Note that $\nu \pi_G^T X$ is a matrix whose entries for each column are all identical. We can hence reinterpret the over-smoothening phenomenon of the WL procedure as the limiting behavior of the operator M_G.

[Random Walks] Let $P \in \mathbb{R}^n$ be a probability distribution over the set of nodes of G. The random walk on G updates P by multiplying its transpose with M_G to the right, i.e. $P \mapsto P^T M_G$, where P^T is the transpose of P. We generalize the construction above by substituting the probability distribution P with the matrix of concatenated node attributes X.

Definition 2.4. Let $G := (V, E)$ be a graph with a matrix of concatenated node attributes X. Fix a positive integer k_2. We say that the node attributes are updated from k_2 iterations of random walks (RW) over G if the matrix X is updated to $X^T M_G^{k_2}$.

The core difference between WL and RW procedure lies in the difference between left and right eigenvectors of M_G. To elaborate, the left eigenspace of M_G with eigenvalue 1 is spanned by the probability distribution of node degrees.

Theorem 2.5 ((Section 1, Lovasz (1993))). *Let $G := (V, E)$ be a finite graph. The left eigenspace of M_G with eigenvalue 1 is spanned by*

$$\pi_G := \left[\frac{d_v}{\sum_{w \in V(G)} d_w} \right]_{v \in V(G)}. \tag{5}$$

Denote by ν the column vector whose entries are all equal to 1. By the Perron-Frobenius Theorem (Theorem Appendix D.1)

$$\lim_{k\to\infty} X^T M_G^k = X^T \nu \pi_G^T, \tag{6}$$

Table 1: A part of classification results obtained from data sets with discrete node labels and continuous node attributes. Cells notated as N/A indicate graph classification schemes which do not report classification results on the given graph data set or may have limitations in processing the node features and edge weights present in the given graph data set. The entries for PROTEINS, BZR, and COX-2 data sets report the highest classification results obtained from using discrete node labels, or from using both discrete and continuous attributes if possible. All classification results other than the proposed method are obtained from pre-existing publications.

	MUTAG	PTC-FR	NCI1	PROTEINS	BZR	BZR-MD	COX2	COX2-MD
WL	88.72±1.11	67.64±0.74	85.58±0.15	76.11±0.64	N/A	N/A	N/A	N/A
PWL-H0	86.10±1.37	67.30±1.50	85.34±0.14	75.31±0.73	N/A	N/A	N/A	N/A
PWL-H1	90.51±1.34	67.15±1.09	85.46±0.16	75.27±0.38	N/A	N/A	N/A	N/A
WWL	87.27±1.50	N/A	85.75±0.25	77.91±0.80	84.42±2.03	69.76±0.94	78.29±0.47	**76.33±1.02**
RetGK-I,II	90.30±1.10	67.80±1.10	84.50±0.20	**78.00±0.30**	87.10±0.70	62.77±1.69	81.40±0.60	59.47±1.66
HGK-WL	N/A	N/A	N/A	76.70±0.41	78.59±0.63	68.94±0.65	78.13±0.45	74.61±1.74
FGW	88.42±5.67	N/A	**86.42±1.63**	74.55±2.74	85.12±4.15	N/A	77.23±4.86	N/A
FC	87.31±0.66	N/A	N/A	74.54±0.48	85.61±0.59	75.61±1.13	81.01±0.88	73.41±0.79
GCKN-subtree	91.60±6.70	68.40±7.40	82.00±1.20	77.60±0.40	86.40±0.50	N/A	**81.70±0.70**	N/A
DM	N/A	68.39±3.57	83.07±1.07	76.19±2.91	N/A	73.55±5.76	N/A	72.28±9.37
Perslay	89.80±0.90	N/A	73.50±0.30	74.80±0.30	N/A	N/A	80.90±1.00	N/A
PWLR-H0	89.52±0.90	**69.12±1.16**	75.66±0.30	74.62±0.62	**89.39±0.74**	75.08±1.32	80.61±1.11	71.36±1.46
PWLR-H1	**91.97±0.92**	67.04±0.67	74.50±0.42	73.02±0.50	**89.29±0.66**	**77.75±0.98**	81.02±0.55	69.96±1.26
PWLR-H0+H1	89.47±1.51	**68.75±1.39**	77.15±0.23	73.94±0.58	**88.95±0.90**	**76.44±1.37**	80.88±0.54	71.36±1.46
PWLR-OPT-H0	89.73±1.01	64.02±0.99	75.99±0.23	74.10±0.59	**89.46±0.55**	72.47±1.44	79.94±0.58	72.10±2.13
PWLR-OPT-H1	**91.91±1.61**	66.19±1.23	72.76±0.33	73.22±0.39	**88.93±0.79**	74.18±1.18	80.22±0.84	70.51±1.86
PWLR-OPT-H0+H1	89.17±0.84	65.73±1.16	79.05±0.40	74.46±0.38	**89.32±0.83**	75.55±1.08	80.97±1.15	72.97±1.00

where each row of $X^T \nu \pi_G^T$ lies in the span of π_G^T. Thus, iterations of RW procedures incorporate information on node degrees with the given node features.

[Persistent Homology] Given k_1 iterations of WL procedure and k_2 iterations of RW procedure, we update the matrix of node labels X to $(M_G^{k_1} X)^T M_G^{k_2}$. Given a node $v \in V$, we use the abbreviation $X^{[k_1,k_2]}(v)$ to denote the features of the node v obtained from the matrix $(M_G^{k_1} X)^T M_G^{k_2}$.

In the spirit of persistent WL (PWL) procedure Rieck et al. (2019), we characterize the global topological invariants of G by constructing a sequence of nested subgraphs induced from the updated matrix $(M_G^{k_1} X)^T M_G^{k_2}$. We define the height functions $h_V : V \to \mathbb{R}$ and $h_E : E \to \mathbb{R}$ as

$$\begin{aligned} h_V(v) &= 0 \text{ for all } v \in V \\ h_E(v_1, v_2) &= \|X^{[k_1,k_2]}(v_1) - X^{[k_1,k_2]}(v_2)\|_p \end{aligned} \tag{7}$$

where $\|\cdot\|_p$ is the l_p-norm over $\mathbb{R}^l$. Let $\tilde{E}$ be the set $E(G)$ that is sorted using the function h_E:

$$\tilde{E} := \{e_i \in E(G) \mid h_E(e_i) \leq h_E(e_j) \text{ if } i \leq j\} \tag{8}$$

The set of edges $\tilde{E}$ is sorted in a manner that their heights are in an increasing order. Using the i-th edge e_i of the sorted list $\tilde{E}$, we define the subgraphs $G^{[i]}$ as

$$G^{[i]} := (V, E^{[i]} := \{e_j \in \tilde{E} \mid j \leq i\}) \tag{9}$$

We now obtain a sequence of nested subgraphs of G:

$$G^{[0]} \subset G^{[1]} \subset G^{[2]} \subset \cdots \subset G^{|E|} = G. \tag{10}$$

The nodes of all $G^{[i]}$'s are fixed, whereas the edges are added in increasing order of their weights.

∘ **Vector Representation:** For each subgraph $G^{[i]}$, we compute the number of its connected components and cycles, corresponding to its 0-th and 1-st homology groups. The ranks of these groups are known as *Betti numbers*, denoted respectively as $\beta_0(G^{[i]})$ and $\beta_1(G^{[i]})$. As more edges are added, the 0-th Betti numbers of subgraphs decrease, whereas the 1-st Betti numbers increase. Given an edge $e_i \in \tilde{E}$ that is included in $G^{[i]}$ but not in $G^{[i-1]}$, the variations in homological invariants can be computed as follows:

$$h_0^i := \beta_0(G^{[i-1]}) - \beta_0(G^{[i]}) \tag{11}$$

$$h_1^i := \beta_1(G^{[i]}) - \beta_1(G^{[i-1]}) \tag{12}$$

Table 2: A part of classification results obtained from data sets with discrete node labels and continuous node attributes. Cells notated as N/A indicate graph classification schemes which do not report classification results on the given graph data set or may have limitations in processing the node features and edge weights present in the given graph data set. The entries for PROTEINS, BZR, and COX-2 data sets report the highest classification results obtained from using discrete node labels, or from using both discrete and continuous attributes if possible. All classification results other than the proposed method are obtained from pre-existing publications.

	MUTAG	PTC-FR	NCI1	PROTEINS	BZR	BZR-MD	COX2	COX2-MD
WL	88.72±1.11	67.64±0.74	**85.58±0.15**	76.11±0.64	N/A	N/A	N/A	N/A
PWL-H0	86.10±1.37	67.30±1.50	85.34±0.14	75.31±0.73	N/A	N/A	N/A	N/A
PWL-H1	90.51±1.34	67.15±1.09	85.46±0.16	75.27±0.38	N/A	N/A	N/A	N/A
WWL	87.27±1.50	N/A	85.75±0.25	77.91±0.80	84.42±2.03	69.76±0.94	78.29±0.47	**76.33±1.02**
RETGK-I,II	90.30±1.10	67.80±1.10	84.50±0.20	**78.00±0.30**	87.10±0.70	62.77±1.69	81.40±0.60	59.47±1.66
GCKN-SUBTREE	91.60±6.70	68.40±7.40	82.00±1.20	77.60±0.40	86.40±0.50	N/A	**81.70±0.70**	N/A
PERSLAY	89.80±0.90	N/A	73.50±0.30	74.80±0.30	N/A	N/A	80.90±1.00	N/A
PWLR	**91.97±0.92**	**69.12±1.16**	77.15±0.23	74.62±0.62	**89.39±0.74**	**77.75±0.98**	81.02±0.55	71.36±1.46
PWLR-OPT	**91.91±1.61**	66.19±1.23	79.05±0.40	74.46±0.38	**89.46±0.55**	75.55±1.08	80.97±1.15	72.97±1.00

Euler's characteristic formula (Appendix D.4) implies that whenever an edge $e \in E^{[i]}$ is newly added, either β_0 decreases by 1, or β_1 increases by 1. Given a connected graph G, there are $|V| - 1$ heights on edges which record decrements of β_0, and $|E| - |V| + 1$ heights on edges which record increments of β_1. The sorted lists of such heights induce the following representations of graphs:

$$\begin{aligned} \varphi_{H_0}^{[k_1,k_2]} &:= [h_E(e_i) + \tau \mid h_0^i > 0]_{e_i \in \tilde{E}} \\ \varphi_{H_1}^{[k_1,k_2]} &:= [h_E(e_i) + \tau \mid h_1^i > 0]_{e_i \in \tilde{E}} \end{aligned} \tag{13}$$

The bias term τ (usually equal to 1) distinguishes cases where the height levels $h_E(e_i)$ are equal to 0 from those where the edges with heights 0 do not occur in G.

○ **Reduced Dimensions:** To further reduce the embedded dimensions, we may record these heights using the following procedure. Any edge $e = (v_1, v_2)$ can be represented as a tuple of unweighted degrees of two nodes $\overline{d}_e := (\overline{d}_{v_1}, \overline{d}_{v_2})$. We denote by $\overline{D}_E$ the set of tuples of unweighted degrees of two nodes connected by an edge:

$$\overline{D}_E := \{\overline{d}_e := (\overline{d}_v, \overline{d}_w) \mid e = (v, w) \in E\} \tag{14}$$

Using the set of tuples, we represent the graphs as $|\overline{D}_E|$-dimensional real vectors by taking constrained summations of sorted heights over the set of edges, based on their associated tuples of unweighted degrees $\overline{d_e}$. Given a fixed tuple $\overline{d}$ of unweighted degrees, the $\overline{d}$-components of the representations $\varphi_{H_0,Opt}^{[k_1,k_2]}$ and $\varphi_{H_1,Opt}^{[k_1,k_2]}$ are sums of heights on edges e_i such that the values h_0^i (or h_1^i, respectively) are positive, and that the tuples of unweighted degrees $\overline{d}_{e_i}$ are equal to $\overline{d}$. The explicit definition of $\varphi_{H_0,Opt}^{[k_1,k_2]}$ and $\varphi_{H_1,Opt}^{[k_1,k_2]}$ can be thus summarized as shown in the following equation:

$$\begin{aligned} \varphi_{H_0,Opt}^{[k_1,k_2]}(\overline{d}) &:= \sum_{\substack{e_i \in \tilde{E} \text{ such that} \\ h_0^i > 0,\ \overline{d}_{e_i} = \overline{d}}} (h_E(e_i) + \tau) \\ \varphi_{H_1,Opt}^{[k_1,k_2]}(\overline{d}) &:= \sum_{\substack{e_i \in \tilde{E} \text{ such that} \\ h_1^i > 0,\ \overline{d}_{e_i} = \overline{d}}} (h_E(e_i) + \tau) \end{aligned} \tag{15}$$

[Representation Stability] It is imperative to verify whether the representations from (13, 15) preserve stability with respect to graph perturbations. Carrière et al. verified that the heat kernel signature preserves representation stability with respect to graph perturbations Carriere et al. (2020); Chazal et al. (2016). As for the PWLR scheme, the incorporation of three algorithms allows us to numerically compute the upper bound of the distance between two representations of graphs. We leave the proof of the stability theorem in Appendix E.3, which uses geometric ergodicity and perturbation theory of random walks over finite graphs Schweitzer (1968); Lovasz (1993).

Theorem 2.6 (Stability for PWLR graph representations)**.** *Let G, G' be two connected graphs with the same number of nodes. Let ϵ be defined as $\epsilon := M_G - M_{G'}$. Denote by $0 < \mu_{2,G}, \mu_{2,G'} < 1$ the*

Table 3: Dimensions of some graph representations processed in the Random Forest Classifier for classifying graphs. Cells notated as ”-“ indicate graph data sets which do not have the prescribed node features. For the first four data sets, the dimensions of representations constructed from WL and PWL procedures are obtained by processing the discrete node labels of graphs, ignoring any continuous node features or attributes on edges. The variable h denotes the number of WL iteration procedures used for obtaining the representations. The asterix ”$*$“ indicates that the obtained graph representations are subject to changes based on the choice of training data sets.

DATA SETS	MUTAG	PTC-FR	NCI1	PROTEINS	BZR	BZR-MD	COX2	COX2-MD
AVERAGE # NODES	17.93	14.56	29.87	39.06	35.75	21.30	41.22	26.28
AVERAGE # EDGES	19.79	15.00	32.30	72.82	38.36	225.06	43.45	335.12
DISCRETE LABELS	7	19	22	3	10	8	8	7
CONTINUOUS FEATURES	-	-	-	29	3	-	3	-
EDGE ATTRIBUTES	-	-	-	-	-	1	-	1
# GRAPHS	188	351	4110	1113	405	306	467	303
GRAPH KERNELS (10-FOLD)*	169*	315*	3699*	996*	364*	275*	420*	272*
WL, PWL-H0 ($h = 1$)	40	148	288	299	N/A	N/A	N/A	N/A
WL, PWL-H0 ($h = 10$)	15,969	27,139	530,723	329,035	N/A	N/A	N/A	N/A
WL, PWL-H0 ($h = 20$)	41,825	63,404	$\geq 10^6$	721,222	N/A	N/A	N/A	N/A
PWLR-H0 (ANY k_1,k_2)	**28**	**64**	**111**	620	**54**	**56**	**33**	**36**
PWLR-H1 (ANY k_1,k_2)	**7**	**8**	**18**	539	**6**	**5**	**6**	**5**
PWLR-OPT (ANY k_1,k_2)	**7**	**10**	**10**	**74**	**9**	**8**	**9**	**9**

second largest eigenvalues of M_G and $M_{G'}$. Then under certain conditions (see Theorem Appendix E.4), there exists a fixed constant $C > 0$ such that

$$\|\varphi_{H_i}^{[k_1,k_2]}(G) - \varphi_{H_i}^{[k_1',k_2']}(G')\|_1 < C(\mu_{2,G}^{k_2} + \mu_{2,G'}^{k_2'} + \|\epsilon\|_1)$$

A generalization of Theorem 2.6 for both representations from (13, 15) can be found in Theorem E.4 and Corollary E.5 in Appendix E. The differences between two representations are numerically controlled by the second largest eigenvalues of M_G and $M_{G'}$ and the perturbation matrix ϵ. The eigenvalues are strictly less than 1 because both matrices are stochastic. Hence, for sufficiently large k_2, the l_1-distance between two vectors is controlled by the norm of ϵ. Theorem Appendix D.2 further shows that the updated node features converge to the probability distribution of node degrees. Hence, the PWLR scheme quantifies the extent of incorporating node degrees to graph representations.

[Time Complexity] The total time complexity for embedding graphs with l-dimensional node attributes using the PWLR procedure up to k_1-iterations of WL kernel and k_2-iterations of RW is $\mathcal{O}(k_1 \times k_2 \times m \times (l + \log m))$. We refer to Appendix E.4 for further details on computing the time complexity of the PWLR algorithm.

3 EXPERIMENTS

We implement the PWLR framework in Python and execute experiments on classifying data sets of finite graphs. Tables 2 and 3 list the classification results and dimensions of representations obtained from the PWLR procedure and contemporary graph embedding techniques.

[Data sets] We classify cheminformatics graph data sets with discrete and continuous features Kersting et al. (2016). For classifying graphs with discrete node labels, we choose MUTAG, PTC, NCI, PROTEINS, and DD data sets. All discrete node labels are one-hot encoded as real coordinate vectors. For classifying graphs with continuous attributes, we choose PROTEINS, BZR, COX2, BZR-MD, and COX2-MD data sets. A full table of classification results can be found in Appendix B.

[Procedures] For implementing the PWLR embedding scheme, we choose two numbers of iterations k_1 and k_2 from 0 to 29. We denote by “PWLR-H0” and “PWLR-H1” the vectors obtained from (13), “PWLR-OPT-H0” and “PWLR-OPT-H1” the vectors obtained from (15), and by “PWLR-H0+H1” and “PWLR-OPT-H0+H1” the vectors obtained from concatenating the two vectors “PWLR-H0” and “PWLR-H1” (“PWLR-OPT-H0” and “PWLR-OPT-H1”, respectively). We implemented 10 iterations of 10-fold cross validations for classifying graph data sets, along with inner 5-fold cross validations over the training sets for tuning the hyperparmeters using grid search. As a classifier, we use the random forest classifier Breiman (2001) to effectively assess the contributions of the architecture of the embedding framework. The optimal number of iterations k_1 and k_2 for classifying graphs are provided in Appendix B.

[Results and Highlights] To evaluate the performance of the proposed algorithm in classifying graphs, we compare the PWLR scheme with WL kernel (WL) Shervashidze et al. (2011), Persistent WL representations (PWL) Rieck et al. (2019), Wasserstein WL kernel (WWL) Togninalli et al. (2019), graph kernels based on return probabilities of random walks (RetGK) Zhang et al. (2018b), hash graph kernels (HGK) Morris et al. (2016), Fused Gromov-Wasserstein kernels (FGW) Titouan et al. (2019), filtration curves for graph representations (FC) O'Bray et al. (2021), the supervised version of graph convolutional kernel networks using subtree features (GCKN-subtree) Chen et al. (2020), Perslay Carriere et al. (2020), and DeepMap (DM) Ye et al. (2020).. The first two procedures can represent graphs with discrete node labels. All other techniques can process graphs with both discrete and continuous node features, and weights on edges. All contemporary classification results are imported from available results recorded in pre-existing publications. Comparisons in classification results obtained from other graph kernels or graph neural networks (GNN) can be found in Appendix B. Table 2 records the highest averages and standard deviations obtained from each graph classification techniques. Experimental results suggest that our PWLR embedding framework possesses two key empirical merits for representing graphs.

○ **Low-dimensional Embeddings:** The proposed scheme constructs a collection of low-dimensional representations independent from the choice of training data sets and the number of iterations k_1 and k_2. As shown in Table 3, it constructs fixed low-dimensional graph representations which produce comparable results to contemporary techniques. We especially notice that the "PWLR-OPT" representations (15) obtained for graphs in large data sets are of substantially smaller dimensions than those obtained from graph kernels or WL procedures. Representations obtained from graph kernel based methods are characterized by inner products between embeddings of graphs and those from the training data set. As such, they heavily depend on the choice of training data. In addition, subsequent iterations of WL procedure substantially increases the number of obtainable distinct node labels, thus inevitably increasing the dimensions of representations.

○ **Classification Results:** The PWLR embedding framework produces comparable results to state-of-the-art techniques in classifying graphs with discrete node labels, and enhances these techniques in classifying graphs with continuous node attributes. While the PWLR embedding framework falls short in classifying some data sets with discrete node labels, all the proposed representations are of low dimensions, a property difficult to guarantee from other graph kernel techniques. As for other graph data sets with both discrete and continuous attributes along with weights on edges, the PWLR embedding framework improves or produces comparable state-of-the-art classification results.

4 Conclusion

The problem of embedding graphs to real vector spaces requires a careful approach to how properties of graphs can be adequately incorporated to their representations. We address this question by introducing a novel mathematical framework for graph representations which guarantees stability with respect to graph perturbations as well as incorporates local topological features, node degrees, and global topological invariants. Experimental results suggest that our PWLR embedding framework provides low-dimensional representations effective for classifying graphs with both discrete and continuous node features. Meanwhile, the PWLR scheme implicitly assumes that the graphs are undirected and stochastically fixed. Future research may hence focus on extending the proposed framework to such graphs.

Acknowledgments

We sincerely thank the reviewers for providing constructive and enlightening comments during the referee process. Sun Woo Park, Yun Young Choi, and Youngho Woo were supported by the National Institute for Mathematical Sciences (NIMS) grant funded by the Korean Government (MSIT) B22920000. Dosang Joe was supported by the National Institute for Mathematical Sciences (NIMS) grant funded by the Korean Government (MSIT) B22810000.

REFERENCES

Baek, M., DiMaio, F., Anischenko, I., Dauparas, J., Ovchinnilkov, S., Lee, G. R., Wang, J., Cong, Q., Kinch, L. N., Schaeffer, R. D., and et al. Accurate prediction of protein structures and interactions using a three-track network. *Science*, 373(6557):871–876, 2021.

Bodnar, C., Frasca, F., Otter, N., Wang, Y., Liò, P., Montufar, G. F., and Bronstein, M. Weisfeiler and lehman go cellular: Cw networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 2625–2640. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/157792e4abb490f99dbd738483e0d2d4-Paper.pdf.

Borgwardt, K. M. and Kriegel, H.-P. Shortest-path kernels on graphs. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pp. 74–81, USA, 2005. IEEE Computer Society. doi: 10.1109/ICDM.2005.132. URL https://doi.org/10.1109/ICDM.2005.132.

Borgwardt, K. M., Ong, C. S., Sch´onauer, S., Vishwanathan, S., Smola, A. J., and Kriegel, H.-P. Protein function prediction via graph kernels. *Bioinformatics*, 21:i47–i56, 2005.

Breiman, L. Random forests. *Machine learning*, 45:5–32, 2001. URL https://doi.org/10.1023/A:1010933404324.

Carlsson, G. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

Carriere, M., Chazal, F., Ike, Y., Lacombe, T., Royer, M., and Umeda, Y. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, pp. 2786–2796, 2020.

Chazal, F., de Silva, V., Glisse, M., and Oudot, S. *The structure and stability of persistence modules*. Springer International Publishing, 2016.

Chen, D., Jacob, L., and Mairal, J. Convolutional kernel networks for graph-structured data. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 1576–1586. PMLR, 2020.

Debnath, A. K., Lopez de Compadre, R. L., Debnath, G., Shusterman, A. J., and HAnsch, C. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, 34(2):786–797, 1991.

Dobson, P. and Doig, A. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of Molecular Biology*, 330(4):771–783, 2003.

Du, S. S., Hou, K., Salakhutdinov, R. R., Poczos, B., Wang, R., and Xu, K. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/663fd3c5144fd10bd5ca6611a9a5b92d-Paper.pdf.

Edelsbrunner, H. and Harer, J. *Computational topology: An introduction*. American Mathematical Society, 2010.

Edelsbrunner, H., Letscher, D., and Zomorodian, A. topological persistence and simplification. *Discrete and computational geometry*, 2002.

Feragen, A., Kasenburg, N., Petersen, J., Bruijne, M. d., and Borgwardt, K. Scalable kernels for graphs with continuous attributes. *Advances in Neural Information Processing Systems*, 32, 2013. URL https://papers.nips.cc/paper/2013/hash/a2557a7b2e94197ff767970b67041697-Abstract.html.

Geerts, F., Mazowiecki, F., and Peréz, G. A. Let's agree to degree: Comparing graph convolutional networks in the message-passing framework. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 3640–3649. PMLR, 2021.

Hatcher, A. *Algebraic Topology*. Cambridge University Press, 2002.

Helma, C., King, R. D., Kramer, S., and Srinivasan, A. The predictive toxicology challenge 2000-2001. *Bioinformatics*, 17(1):107–108, 2001.

Hofer, C., Kwitt, R., Niethammer, M., and Uhl, A. Deep learning with topological signatures. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/883e881bb4d22a7add958f2d6b052c9f-Paper.pdf.

Hu, N., Rustamov, R. M., and Guibas, L. J. Stable and informative spectral signatures for graph matching. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2313–2320, 2014.

Isaacson, D. L. and Madsen, R. W. *Markov chains: theory and applications*. John wiley and Sons, 1976.

Kashima, H., Tsuda, K., and Inokuchi, A. Marginalized kernels between labeled graphs. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pp. 321–328. AAAI Press, 2003.

Kersting, K., Kriege, N. M., Morris, C., Mutzel, P., and Neumann, M. Benchmark data sets for graph kernels, 2016. URL http://graphkernels.cs.tu-dortmund.de.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Lawler, G. *Introduction to Stochastic Processes*. Chapman and Hall, 2006.

Lovasz, L. Random walks on graphs: a survey. *Combinatorics*, pp. 1–46, 1993.

Morris, C., Kriege, N. M., Kersting, K., and Mutzel, P. Faster kernel for graphs with continuous attributes via hashing. In *IEEE International Conference on Data Mining (ICDM), 2016*, pp. 1095–1100, 2016.

Morris, C., Kersting, K., and Mutzel, P. Glocalized weisfeiler-lehman graph kernels: Global-local feature maps of graphs. In *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 327–336, 2017. doi: 10.1109/ICDM.2017.42.

Niepert, M., Ahmed, M., and Kutzkov, K. Learning convolutional neural networks for graphs. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pp. 2014–2023. PMLR, 2016.

Nikolentzos, G. and Vazirgiannis, M. Random walk graph neural networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 16211–16222. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/ba95d78a7c942571185308775a97a3a0-Paper.pdf.

O'Bray, L., Rieck, B., and Borgwardt, K. Filtration curves for graph representation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '21, pp. 1267–1275. Association for Computing Machinery, 2021.

Rieck, B., Bock, C., and Borgwardt, K. A persistent Weisfeiler–Lehman procedure for graph classification. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 5448–5458, Long Beach, California, USA, 2019. PMLR. URL http://proceedings.mlr.press/v97/rieck19a.html.

Schweitzer, P. Perturbation theory and finite markov chains. *Journal of Applied Probability*, 5(3): 401–413, 1968.

Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Zidek, A., Nelson, A. W., Bridgland, and et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011.

Stokes, J. M., Yang, K., Swanson, K., Jin, W., Cubillos-Ruiz, A., Donghia, N. M., MacNair, C. R., French, S., Carfrae, L. A., Bloom-Ackerman, Z., and et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):401–413, 2020.

Sugiyama, M. and Borgwardt, K. Halting in random walk kernels. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/31b3b31a1c2f8a370206f111127c0dbd-Paper.pdf.

Sutherland, J. J., O'brien, L. A., and Weaver, D. F. Spline-fitting with a genetic algorithm: A method for developing classification structure-activity relationships. *Journal of chemical information and computer sciences*, 43(6):1906–1915, 2003.

Titouan, V., Courty, N., Tavenard, R., Laetitia, C., and Flamary, R. Optimal transport for structured data with application on graphs. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 6275–6284, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/titouan19a.html.

Togninalli, M., Ghisu, E., Llinares-López, F., Rieck, B., and Borgwardt, K. Wasserstein weisfeiler–lehman graph kernels. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pp. 6436–6446. Curran Associates, Inc., 2019.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Vishwanathan, S., Schraudolph, N. N., Kondor, R., and Borgwardt, K. M. Graph kernels. *Journal of Machine Learning Research*, 11:1201–1242, 2010.

Wale, N. and Karypic, G. Comparison of descriptor spaces for chemical compound retrieval and classification. *Proceedings of the International Conference on Data Mining*, 14:347–375, 2008. URL https://doi.org/10.1007/s10115-007-0103-5.

Weisfeiler, B. and Lehman, A. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia, Ser.2*, 9, 1968.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Ye, W., Askarisichani, O., Jones, A., and Singh, A. Learning deep graph representations via convolutional neural networks. *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020. doi: 10.1109/TKDE.2020.3014089.

Zhang, M., Cui, Z., Neumann, M., and Chen, Y. An end-to-end deep learning architecture for graph classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018a. URL https://ojs.aaai.org/index.php/AAAI/article/view/11782.

Zhang, Z., Wang, M., Xiang, Y., Huang, Y., and Nehorai, A. Retgk: Graph kernels based on return probabilities of random walks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018b. URL https://proceedings.neurips.cc/paper/2018/file/7f16109f1619fd7a733daf5a84c708c1-Paper.pdf.