Experimental Investigation of Neural and Weisfeiler-Lehman-Kernel Graph Representations for Downstream SVM-Based Classification

Sergej Borisov[1], Marek Dědič[2,3], and Martin Holeňa[4]

[1] Faculty of Mathematics and Physics, Charles University, Prague, borisov@cs.cas.cz
[2] Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University, Prague, marek@dedic.eu
[3] Cisco Cognitive Intelligence, Cisco Systems, Inc., Prague
[4] Institute of Computer Science, Academy of Sciences, Prague, martin@cs.cas.cz

Abstract: Graphs are one of the most ubiquitous kinds of data. However, data analysis methods have been developed primarily for numerical data, and to make use of them, graphs need to be represented as elements of some Euclidean space. An increasingly popular way of representing them in this way are graph neural networks (GNNs). Because data analysis applications typically require identical results for isomorphic graphs, the representations learned by GNNs also need to be invariant with respect to graph isomorphism. That motivated recent research into the possibilities of recognizing non-isomorphic pairs of graphs by GNNs, primarily based on the Weisfeiler-Lehman (WL) isomorphism test. This paper reports the results of a first experimental comparison of four variants of two important GNNs based on the WL test from the point of view of graph representation for downstream classification by means of a support vector machins (SVM). Those methods are compared not only with each other, but also with a recent generalization of the WL subtree kernel. For all GNN variants, two different representations are included in the comparison. The comparison revealed that the four considered representations of the same kind of GNN never significantly differ. On the other hand, there was always a statistically significant difference between representations originating from different kinds of GNNs, as well as between any representation originating from any of the considered GNNs and the representation originating from the generalized WL kernel.

Keywords: graph representation learning, graph neural networks, message-passing networks, Weisfeiler-Lehman isomorphism test, Weisfeiler-Lehman subtree kernel

1 Introduction

In the present time, graph-structured data are one of the most ubiquitous kinds of data. However, from the point of view of data analysis and knowledge discovery in data, they received attention only during the last decades. Therefore, data analysis methods for common tasks such as classification, regression and clustering have been developed primarily for numerical data, and if they are used for graph-structured data, graphs needs to be represented as vectors in some Euclidean space. This requirement is not specific for graphs, but can also be encountered with other kinds of non-numerical data, the best-known example probably being textual data. The seminal papers about graph representation [20] and [9], which introduced, respectively, the neighbourhood sampling strategies DeepWalk and node2vec, were strongly influenced by the Skip-gram model for text, implemented by the word2vec algorithm [16, 17].

An increasingly popular way of representing graphs by vectors is using *graph neural networks* [10]. Typically, downstream applications are desired to provide identical results for isomorphic graphs. Consequently, the representations learned by GNNs need to be *invariant with respect to isomorphism*. That motivated recent research into the possibilities of recognizing non-isomorphic pairs of graphs by GNNs [1, 3, 6, 11, 15, 18, 29], primarily based on the classical WL isomorphism test, capable of revealing such pairs in many situations [27].

The WL-test iteratively constructs neighbourhood subtrees rooted in graph vertices. Such a construction can also be used for graph kernels [22]. In particular [24] introduced the *WL graph kernels*, of which most relevant to our work is the WL subtree kernel. That kernel was recently generalized to the *relaxed WL kernel* [23]. Whereas the kernel itself evaluates the graph with a scalar value, the rooted subtrees used in a kernel definition can be easily represented with vectors of non-negative numbers. Therefore, the WL kernels can be viewed as a kernel counterpart of representation learning with WL-test-based GNNs.

This work-in-progress paper reports a first experimental comparison of two important WL-test-based GNNs with the general relaxed WL kernel from the point of view of graph representation for downstream classification by means of an SVM. Of each GNN, four variants were available, and for all employed GNN variants, two different representations were included into the comparison. The comparison was performed on 20 graphsets created from benchmark datasets and its results were tested for statistical significance.

The next section gives an overview of message-passing neural networks, which are the most common kind of GNNs, and at the same time the kind to which the known WL-test-based GNNs belong. It also recalls the principles of the relaxed WL kernel. The key part of the paper is Section 3, in which the performed experimental comparison is described and its first results are presented. Finally, Sec-

tion 4 concludes the paper and indicates possible furhter research.

2 Methodology

2.1 Preliminaries

In this comparison, we consider *undirected graphs with labelled and colored vertices*. More precisely, graphs $G = (V,E,\lambda)$ such that

- $E \subseteq \{X \subseteq V : \#X = 2\}$, where V is the set of vertices, E is the set of edges, and $\#X$ denotes the cardinality of X;
- $\lambda : V \to \mathbb{R}^{d_\lambda}$ denotes a general labelling;
- $\exists c : V \to \{0,1\}^{d_c}$, called colouring, where $d_c \leq d_\lambda$, $\forall v \in V : (c(v))_1 + \cdots + (c(v))_{d_c} = 1$, and c fulfils $c = (\lambda_1, \ldots, \lambda_{d_c})$, i.e. the coloring is part of the general labelling, which may in addition contain also other components.

Using this notation and the symbol $N(v)$ for the neighbours of vertex v in the graph G, the WL isomorphism test [27] can be described. The test consists in an iterative algorithm applied to pairs G, G' of graphs with equal numbers of vertices and such that if G and G' are isomorphic, then the isomporphism preserves their colorings. In each iteration, the colorings are updated, taking into account the neighbours of all vertices, in such a way that an isomorphism preserves also the updated colorings. Hence, if after a finite number of iterations, some value of the updated coloring occurs in both graphs with different frequency, then the isomorphism of G and G' can be rejected. However, it can never be definitely confirmed. The pseudocode of the WL test is in Algorithm 1.

Denote A_G and A'_G the adjacency matrices of G and G', respectively, and $1_{\#V}$ the vector $(1,\ldots,1)$ of length $\#V$. Then the situation that the WL isomorphism test cannot reject isomorphism of G and G' can be characterized by the following three equivalent conditions [5]:

(i) the WL test does not reject the isomorphism of G and G' within $\#V$ iterations;

(ii) for each tree T, the number of homomorphisms from T to G equals the number of homomorphisms from T to G';

(iii) there exists a matrix $X \in 0,1^{\#V \times \#V}$ solving the following system of linear equations:

$$\begin{aligned} A_G X &= X A'_G, \\ X 1_{\#V} &= 1_{\#V}, \\ 1_{\#V}^\top X &= 1_{\#V}^\top. \end{aligned} \tag{4}$$

Observe that if G and G' are really isomorphic, then the equations (4) are solved by the permutation matrix X fulfilling $A'_G = X^\top A_G X$, which transforms A_G into A'_G.

Algorithm 1 Weisfeiler-Lehman isomorphism test

Require: Graphs G, G' with sets of vertices V, V', sets of edges E, E', and colourings c, c' such that $V \cap V' = \emptyset, \#V = \#V'$, maximal number of iterations $h \leq \#V$

Define the colouring $c_0 : V \cup V' \to \{0,1\}^{d_c} \cup \{0,1\}^{d'_c}$ by

$$c_0(v) = \begin{cases} c(v) & \text{if } v \in V \\ c'(v) & \text{if } v \in V' \end{cases} \tag{1}$$

Set $\Sigma_0 = c_0(V \cup V')$

Order Σ_0 as $\sigma_0^1, \ldots, \sigma_0^{\#\Sigma_0}$

if exists $j = 1, \ldots, \#\Sigma_0$ such that $\#\{v \in V : c_0(v) = \sigma_0^j\} \neq \#\{v \in V' : c_0(v) = \sigma_0^j\}$ **then**

    Return the fact that G and G' are not isomorphic

else

    Set $i = 1$

    Define the colouring $c_1 : V \cup V' \to \Sigma_0 \times \mathbb{N}_0^{\#\Sigma_0}$ by

$$\begin{aligned} &c_1(v) = (c_0(v), (s_1, \ldots, s_{\#\Sigma_0})), \text{ where} \\ &s_j = \#\{u \in N(v) : c_0(u) = \sigma_0^j\} \text{ for } j = 1, \ldots, \#\Sigma_0 \end{aligned} \tag{2}$$

    Set $\Sigma_1 = c_1(V \cup V')$

    Order Σ_1 as $\sigma_1^1, \ldots, \sigma_1^{\#\Sigma_1}$

end if

while $i < h$ and $\#\{v \in V : c_i(v) = \sigma_i^j\} = \#\{v \in V' : c_i(v) = \sigma_i^j\}$ for $j = 1, \ldots, \#\Sigma_i$ **do**

    Increment $i = i + 1$

    Define the colouring $c_i : V \cup V' \to \Sigma_{i-1} \times \mathbb{N}_0^{\#\Sigma_{i-1}}$ by

$$\begin{aligned} &c_i(v) = (c_{i-1}(v), (s_1, \ldots, s_{\#\Sigma_{i-1}})), \text{ where} \\ &s_j = \#\{u \in N(v) : c_0(u) = \sigma_0^j\} \text{ for } j = 1, \ldots, \#\Sigma_i \end{aligned} \tag{3}$$

    Set $\Sigma_i = c_i(V \cup V')$

    Order Σ_i as $\sigma_i^1, \ldots, \sigma_i^{\#\Sigma_i}$

end while

if exists $j = 1, \ldots, \#\Sigma_i$ such that $\#\{v \in V : c_i(v) = \sigma_0^j\} \neq \#\{v \in V' : c_i(v) = \sigma_0^j\}$ **then**

    Return the information that G and G' are not isomorphic

else

    Return the information that the WL test did not reject the isomorphism of G and G' within h iterations

end if

Finally, a *feedforward artificial neural network* NN will for the purpose of this paper be a stationary connectionist feedforward structure

$$\mathrm{NN} = (I, O, H, C, \mathscr{F}, (\Phi_e)_{e\in C}, (\Psi_v)_{v\in H\cup O}) \text{ where} \quad (5)$$

- I, O and H are mutually disjoint sets of input, output and hidden neurons;
- $C \subseteq I \times H \cup H \times H \cup H \times O$ is a set of connections;
- $\forall v \in I : \mathrm{out}(v) = \{\xi \in H \cup O : (v,\xi) \in C\} \neq \emptyset$, $\mathrm{out}(v)$ is the output set of the neuron v;
- $\forall v \in O : \mathrm{inp}(v) = \{\xi \in I \cup H : (\xi, v) \in C\} \neq \emptyset$, $\mathrm{inp}(v)$ is the input set of the neuron v;
- $\forall v \in H : \mathrm{inp}(v) \neq \emptyset, \mathrm{out}(v) \neq \emptyset$;
- $\exists \mathrm{Dom} \subseteq \mathbb{R}^{\#I} : \mathscr{F} \subseteq \{F : \mathrm{Dom} \to \mathbb{R}^{\#O}\}$, $\mathscr{F}$ is the set of mappings computable by the NN;
- $\forall v \in H \cup O \; \exists \mathrm{Dom}_v \subseteq \mathbb{R}^{\#\mathrm{inp}(v)} : \Psi_v \subseteq \{\psi : \mathrm{Dom}_v \to \mathbb{R}\}$, the elements of Ψ_v are called *somatic operators* available for the neuron v;
- $\forall e \in C \; \exists \mathrm{Dom}_e \subseteq \mathbb{R} : \Phi_e \subseteq \{\varphi : \mathrm{Dom}_e \to \mathbb{R}\}$, the elements of Φ_e are called *synaptic operators* available for the connection e;
- all mappings computable by the NN fulfil the following condition of resursive composability: $\forall F \in \mathscr{F} \; \forall v \in H \cup O \; \forall \xi \in \mathrm{inp}(v) \; \exists \psi_v \in \Psi_v \; \exists \varphi_{(\xi,v)} \in \Phi_{(\xi,v)} \forall X \in \mathrm{Dom} \; \exists A_X : I \cup H \cup O \to \mathbb{R}$ – the activation state of NN corresponding to the input X, fulfilling

$$A_X|I = X, A_X|O = F(X), \quad (6)$$

$$\forall v \in H \cup O : A_X(v) = \psi_v((\varphi_{(\xi,v)}(A_X(\xi)))_{\xi\in\mathrm{inp}(v)}). \quad (7)$$

The above definition covers in particular multilayer perceptrons and radial basis functions, for which a plethora of theoretical results is available, such as the classical universal approximation results concerning density of $\mathscr{F}$ in general function spaces [4, 12–14, 19], and results concerning the applicability of the laws of large numbers and of the central limit theorem [28]. As usually, the set $\mathscr{F}$ will be assumed parametrizable with a finite-dimensional set W of parameters. Hence,

$$\exists q \in \mathbb{N} \; \exists \mathscr{W} \subseteq \mathbb{R}^q \exists \omega : W \to \{F : \mathrm{Dom} \to \mathbb{R}^{\#O}\} \text{ such that } \mathscr{F} = \omega(\mathscr{W}). \quad (8)$$

Needless to say, the parametrizability of $\mathscr{F}$ also induces the parametrizability of the sets of somatic operators $\Psi_v, v \in H \cup O$ and synaptic operators $\Phi_e, e \in C$.

2.2 Message-passing Neural Networks

Message-passing neural networks (MPNN) are probably the kind of neural networks most often used for graph data. They are characterized by the following two properties [10]:

1. Their input and hidden neurons are structured into a sequence of layers $H_0 = I, H_1, \ldots, H_L$, like in the case of multilayer perceptrons (MLPs) and radial basis functions:

$$k \neq \ell \Rightarrow H_k \cap H_\ell = \emptyset, \quad C \cap (H \times H) \subseteq \bigcup_{1\leq\ell\leq L} H_{\ell-1} \times H_\ell. \quad (9)$$

2. The structure of their connections as well as their somatic and synaptic operators attempts to reflect the structure and properties of the graphs to which the network is applied. The approach that is most typical, and at the same time perfectly suitable for the kind of graphs considered in this paper, consists in:
   - network connections attempt to reflect the structure of graph edges, in particular input and output sets of neurons attempt to reflect the neighbourhoods of graph vertices;
   - in each layer $\ell = 0, \ldots, L$, a group of neurons $v_{v,1}, \ldots, v_{v,d_\ell} \in H_\ell$ can be assigned to a vertex $v \in V$, with $d_0 = d_\lambda$, and the vector $\psi_v^{(\ell)} = (\psi_{v_{v,1}}, \ldots, \psi_{v_{v,d_\ell}})$ of their somatic operators for a layer $\ell < L$ fulfils

$$\psi_v^{(\ell+1)} = f_{\mathrm{up}}^{(\ell)}(\psi_v^{(\ell)}, f_{\mathrm{ag}}^{(\ell)}(\psi_{u_1}^{(\ell)}, \ldots, \psi_{u_{\#N(v)}}^{(\ell)})), \quad (10)$$

     where $f_{\mathrm{up}}^{(\ell)} : \mathbb{R}^{d_\ell} \times \mathbb{R}^{d_{\mathrm{ag}}} \to \mathbb{R}^{d_{\ell+1}}$ is called *update function*, $\{u_1, \ldots, u_{\#N(v)}\} = N(v)$, $f_{\mathrm{ag}}^{(\ell)} : \mathbb{R}^{d_\ell \times \#N(v)} \to \mathbb{R}^{d_{\mathrm{ag}}}$ is called *aggregation function*, and $d_{\mathrm{ag}} \in \mathbb{N}$ with a usual choice $d_{\mathrm{ag}} = d_\ell$ or $d_{\mathrm{ag}} = d_{\ell+1}$;
   - synaptic operators are not used.
3. The resulting mapping F composed according to (6) is required to be either invariant or equivariant with respect to permutations of the d_0 columns of its input matrix. Needless to say, equivariance – preserving in the output the permutation of the input columns, is possible only if $|O| = d_0$.

An important concept relevant to MPNN as well as to other kinds of GNNs is *consistency with graph colouring*. An MPNN is consistent with the colouring c of a graph G if it fulfils

$$\psi_u^{(0)} = \psi_v^{(0)} \Leftrightarrow c(u) = c(v). \quad (11)$$

Because $(\boldsymbol{\psi}_v^{(\ell)})_{v\in V}$ is for each $\ell = 1,\dots,L$ a mapping into the Euclidean space $\mathbb{R}^{d_\ell \times \#V}$, or equivalently, into the Euclidean space $\mathbb{R}^{d_\ell \#V}$, it can be used for graph representation. Usually, however, only $(\boldsymbol{\psi}_v^{(L)})_{v\in V}$ is used to this end. In our experiments reported in Section 3, both possibilities were used.

Network 1-GNN was proposed in [18] as part of a study of the relationships between GNNs and the WL test. That study actually considered more general GNNs, called k-GNNs, with a general $k \in \mathbb{N}$, and investigated their relationships to k-WL tests, which are generalizations of the WL test from vertices to k-tuples of vertices. The 1-GNN is not only the simplest network of this kind, but at the same time also a specific kind of MPNN, for which the functions $f_{\text{up}}^{(\ell)}$ and $f_{\text{ag}}^{(\ell)}$ in (10) are defined as

$$\begin{aligned} f_{\text{up}}^{(\ell)}(\boldsymbol{\psi}_v^{(\ell)}, f_{\text{ag}}^{(\ell)}(\boldsymbol{\psi}_{u_1}^{(\ell)},\dots,\boldsymbol{\psi}_{u_{\#N(v)}}^{(\ell)})) = \\ = \boldsymbol{\sigma}(W_1^{(\ell)}\boldsymbol{\psi}_v^{(\ell)} + b^{(\ell)} + f_{\text{ag}}^{(\ell)}(\boldsymbol{\psi}_{u_1}^{(\ell)},\dots,\boldsymbol{\psi}_{u_{\#N(v)}}^{(\ell)})), \\ f_{\text{ag}}^{(\ell)}(\boldsymbol{\psi}_{u_1}^{(\ell)},\dots,\boldsymbol{\psi}_{u_{\#N(v)}}^{(\ell)}) = \sum_{j=1}^{\#N(v)} W_2^{(\ell)}\boldsymbol{\psi}_{u_j}^{(\ell)}, \end{aligned} \quad (12)$$

where $W_1^{(\ell)}, W_2^{(\ell)} \in \mathbb{R}^{d_{\ell+1}} \times \mathbb{R}^{d_\ell}, b^{(\ell)} \in \mathbb{R}^{d_{\ell+1}}$, and $\boldsymbol{\sigma}$ is a component-wise non-linear activation function, e.g. a sigmoid or ReLU.

From the point of view of a relationship between this kind of networks and the WL test, it is important that for a 1-GNN consistent with the graph colouring, and for the colouring c_i constructed in the i-th iteration of Algorithm 1, $i \leq \min(h,L)$, it was proven in [18] that:

(i) $\forall u,v \in V : c_i(u) = c_i(v) \Rightarrow \boldsymbol{\psi}_u^{(i)} = \boldsymbol{\psi}_v^{(i)}$;

(ii) there exists a sequence of weight matrices and bias vectors $(W_1^{(0)}, W_2^{(0)}, b^{(0)}), \dots, (W_1^{(i-1)}, W_2^{(i-1)}, b^{(i-1)})$ such that if the functions $f_{\text{up}}^{(0)}, f_{\text{ag}}^{(0)}, \dots, f_{\text{up}}^{(i-1)}, f_{\text{ag}}^{(i-1)}$ are defined using this sequence, then for any $i' = 1,\dots,i, u,v \in V : \boldsymbol{\psi}_u^{(i')} = \boldsymbol{\psi}_v^{(i')} \Rightarrow c_{(i')}(u) = c_{(i')}(v)$.

Graph Isomorphism Network (GIN) was proposed in [29] and defines the functions $f_{\text{up}}^{(\ell)}$ and $f_{\text{ag}}^{(\ell)}$ in (10) as

$$\begin{aligned} f_{\text{up}}^{(\ell)}(\boldsymbol{\psi}_v^{(\ell)}, f_{\text{ag}}^{(\ell)}(\boldsymbol{\psi}_{u_1}^{(\ell)},\dots,\boldsymbol{\psi}_{u_{\#N(v)}}^{(\ell)})) = \\ = f_{\text{mlp}}^{(\ell)}((1+\varepsilon_\ell))\boldsymbol{\psi}_v^{(\ell)} + f_{\text{ag}}^{(\ell)}(\boldsymbol{\psi}_{u_1}^{(\ell)},\dots,\boldsymbol{\psi}_{u_{\#N(v)}}^{(\ell)})), \\ f_{\text{ag}}^{(\ell)}(\boldsymbol{\psi}_{u_1}^{(\ell)},\dots,\boldsymbol{\psi}_{u_{\#N(v)}}^{(\ell)}) = \sum_{j=1}^{\#N(v)} \boldsymbol{\psi}_{u_j}^{(\ell)}, \end{aligned} \quad (13)$$

where $\varepsilon_\ell > 0$ and $f_{\text{mlp}}^{(\ell)} : \mathbb{R}^{d_\ell} \to \mathbb{R}^{d_{\ell+1}}$ is an MLP producing the representation of the graph in the $(\ell+1)$st layer.

A relationship between GIN and the WL test is based on applying the universal approximation resultsfor MLPs [4, 12–14] to $f_{\text{mlp}}^{(\ell)}$, in combination with a result proven in [29]as Corollary 6:

Let $\mathscr{X} \subseteq \mathbb{R}^{d_\ell}$ be countable and bounded, $p_{\max} \in \mathbb{N}$. Then there exists a function $f : \mathscr{X} \to \mathbb{R}^{d_\ell+1}$ such that for infinitely many choices of ε, including all irrational numbers, the function $h : \mathscr{X} \times \bigcup_{p=1}^{p_{\max}} \mathscr{X}^p \to \mathscr{X}$ defined

$$\forall c \in \mathscr{X}\ \forall p = 1,\dots,p_{\max}\ \forall X = (x_1,\dots,x_p) \in \mathscr{X}^p : \\ h(c,X) = (1+\varepsilon)f(c) + \sum_{j=1}^{p} f(x_j) \quad (14)$$

is unique with respect to multisets, which means that

$$\begin{aligned} \forall c \in \mathscr{X}\ \forall p = 1,\dots,p_{\max} \forall X,X' \in \mathscr{X}^p : \\ [\forall x \in \mathscr{X} : \#\{j = 1,\dots,p : x_j = x\} = \\ = \#\{j = 1,\dots,p : x'_j = x\}] \Rightarrow h(c,X) = h(c,X'). \end{aligned} \quad (15)$$

2.3 Relaxed Weisfeiler-Lehman Kernel

The relaxed WL kernel [23] is a recent relaxation of the *WL subtree kernel*, which was proposed in [24] and is based on the WL isomorphism test described in Algorithm 1. Taking into account the number of iterations of the WL test, which is in the context of the WL subtree kernel called depth, the value of this kernel for a pair of graphs $G = (V,E,\lambda), G' = (V',E',\lambda')$ is defined as

$$k_{\text{WLsubtree}}^{(h)}(G,G') = \sum_{i=0}^{h} \sum_{v\in V} \sum_{v'\in V'} \delta(c_i(v), c_i(v')), \quad (16)$$

where δ denotes the Kronecker delta, also known as Dirac kernel. Consequently, the WL subtree kernel reflects only exact match of the colouring produced for both graphs in every iteration of the WL test, although in many real-world problems, more important than exact match is a similarity of those colourings.

To overcome this drawback, the exact match is in [23] for $i = 0,\dots,h$ weakened to the equivalence with respect to the clusters $C_1^\rho,\dots,C_{k_\rho}^\rho$ produced by a partitioning ρ of Σ_i. Hence,

$$\begin{aligned} \forall i,j = 1,\dots,k_\rho : i \neq j \Rightarrow C_i^\rho \cap C_j^\rho = \emptyset, \\ C_1^\rho \cup \dots \cup C_{k_\rho}^\rho = \Sigma_i,\ \rho : \Sigma_i \to \{1,\dots,k_\rho\}, \\ \forall k = 1,\dots,k_\rho,\ \forall \sigma \in C_k^\rho : \rho(\sigma) = k. \end{aligned} \quad (17)$$

Moreover, for each Σ_i, not only one such partitioning ρ is used, but a finite set $\Theta_i = \{\rho_1^i,\dots,\rho_{\#\Theta_i}^i\}$ of them. This turns (16) finally into the definition of a relaxed WL kernel:

$$k_{\text{R-WL}}^{(h)}(G,G') = \sum_{i=0}^{h} \sum_{\rho\in\Theta_i} \sum_{v\in V} \sum_{v'\in V'} \delta(\rho(c_i(v)), \rho(c_i(v'))). \quad (18)$$

Its name originates from the fact that $k_{\text{R-WL}}^{(h)}$ is more general than $k_{\text{WLsubtree}}^{(h)}$, to which it turns if $\Theta_i = \{\rho_i\}$ with $\rho_i(\sigma) = \{\sigma\}$ for $i = 1,\dots,h, \sigma \in \Sigma_i$.

The construction of the partitionings $\rho \in \Theta_i, i = 1,\ldots,h$, in (18) is based on replacing each $\sigma \in \Sigma_i$ for $i = 1,\ldots,h$ with a set of isomorphic unfolding trees. An unfolding tree $T^i(G,v)$ of depth i in a graph G rooted in a vertex $v \in V$ will be defined in accordance with [5], as a rooted tree $T = (V(T),E(T))$ with root r such that:

- $\exists f$ – homomorphism from T to G;
- $f(r) = v$;
- for each non-leaf $t \in V(T)$, f induces a bijection between the set of children of t in T and $N(v)$ in G.

For unfolding trees, a natural distance is a tree-edit distance. The definition of the tree-edit distance employed in the context of the relaxed WL kernel can be found in [23].

From the point of view of graph representation, it is important that (16) and (18) are actually scalar products of vectors, which suggests to use those vectors as Euclidean-space representations of the graphs G and G' in downstream data analysis applications. In particular for the WL subtree kernel, define a vector $\phi^{\mathrm{WL}}(G) \in \mathbb{R}^{n_{\mathrm{WL}}}$ with $n_{\mathrm{WL}} = \sum_{i=0}^{h} \#\Sigma_i$ as

$$\phi^{\mathrm{WL}}(G) = (\phi^{\mathrm{WL}}_{0,1},\ldots,\phi^{\mathrm{WL}}_{0,\#\Sigma_0},\phi^{\mathrm{WL}}_{1,1},\ldots,\phi^{\mathrm{WL}}_{h,\#\Sigma_h}), \quad (19)$$

where for $i = 0,\ldots,h$, $j = 1,\ldots,\#\Sigma_i$,

$$\phi^{\mathrm{WL}}_{i,j} = \#\{v \in V : c_i(v) = \sigma^i_j\}. \quad (20)$$

Then (16) can be indeed rewritten as the scalar product

$$k^{(h)}_{\mathrm{WLsubtree}}(G,G') = \phi^{\mathrm{WL}}(G)^{\top}\phi^{\mathrm{WL}}(G'). \quad (21)$$

Similarly for the relaxed WL kernel, define a vector $\phi^{\text{R-WL}}(G) \in \mathbb{R}^{n_{\text{R-WL}}}$ with $n_{\text{R-WL}} = \sum_{i=0}^{h}\sum_{j=1}^{\#\Theta_i} k_{\rho^i_j}$ as

$$\phi^{\text{R-WL}}(G) = (\phi^{\text{R-WL}}_{0,1},\ldots,\phi^{\text{R-WL}}_{0,\#\Sigma_0},\phi^{\text{R-WL}}_{1,1,1},\ldots,\phi^{\text{R-WL}}_{1,1,k_{\rho^1_1}},$$
$$\phi^{\text{R-WL}}_{1,2,1},\ldots,\phi^{\text{R-WL}}_{1,\#\Theta_1,k_{\rho^1_{\#\Theta_1}}},\phi^{\text{R-WL}}_{2,1,1},\ldots,\phi^{\text{R-WL}}_{h,\#\Theta_h,k_{\rho^h_{\#\Theta_h}}}), \quad (22)$$

where for $i = 0,\ldots,h$, $j = 1,\ldots,\#G_i$, $k = 1,\ldots,k_{\rho^i_j}$,

$$\phi^{\text{R-WL}}_{i,j,k} = \#\{v \in V : \rho^i_j(c_i(v)) = k\}. \quad (23)$$

Then (18) can be rewritten as the scalar product

$$k^{(h)}_{\text{R-WL}}(G,G') = \phi^{\text{R-WL}}(G)^{\top}\phi^{\text{R-WL}}(G'). \quad (24)$$

It is useful to realize that the label sets Σ_i for $i = 0,\ldots,h$ on which the definitions of ϕ^{WL} and $\phi^{\text{R-WL}}$ rely, were defined only for a pair of graphs (G,G'), as $\Sigma_i = c_i(V \cup V')$, cf. Line 15 od Algorithm 1. For dealing with a whole set $\mathscr{G}$ of graphs such that each $G \in \mathscr{G}$ has its own sets of vertices V^G and edges E^G as well as its own labelling λ^G and colouring c^G, then it is convenient to generalize their definition to

$$\Sigma_i = \bigcup_{G\in\mathscr{G}} c^G_i(v^G), i = 0,\ldots,h. \quad (25)$$

It is this definition that we used in our experiments.

3 Experimental Comparison

For the experimental comparison, we implemented the 1-GNN and GIN networks using layers and neural networks learning methods available in the PyTorch Geometric library [21]. We also made use of the `GenWL` implementation [8] of the relaxed WL kernel by the authors of the paper [23]. Both network implementations were used in their decay variants, and the `GenWL` implementation was used in the R-WL* variant, in accordance with the experiments reported in [23].

3.1 Employed Graphsets

To be able to asses the statistical significance of the comparison results, we performed the comparison on 20 mutually disjoint graphsets GS1–GS20, adopting the usual assumption that for the employed data, disjointness is a sufficient condition for statistical independence. Those graphsets were obtained from real-world benchmark sets of graph data. Due to our objective of investigating the suitability of the compared graph representation methods for downstream classification, we used datasets from binary graph classification tasks to this end.

We have chosen four benchmark sets of graph data, which are available both in the PyTorch Geometric library [21], and in the `GenWL` implementation of the relaxed WL kernel [8].

1. *BZR* is a set of 405 ligands for the benzodiazepine receptor, classified with respect to their activity in benzodiazepine binding [25].
2. *COX-2* is a set of 467 cyclooxygenase-2 inhibitors, classified with respect to their activity against human recombination enzyme [25].
3. *DHFR* is a set of 756 inhibitors of dihydrofolate reductase, classified with respect to their activity in the inhibition of the enzymatic reduction that converts dihydrofolate to tetrahydrofolate [25].
4. *NC1* is a set of 4110 compounds evaluated in bioassays of the National Cancer Institute on non-small cells of lung tumour, classified with respect to growth inhibition of this kind of human tumour [26].

The precise origin of the graphsets GS1–GS20 in those four benchmark sets is listed in Table 1.

3.2 Experimental Setup

Before starting the experiments, we need two make two decisions concerning the compared GNNs: First, what network topology to use, and how to obtain a graph representation from a trained network. Second, a decision concerning the comparison as a whole, including the relaxed WL kernel – how to evaluate the suitability of each representation for downstream classification. We now address each of those design decisions in some detail.

Table 1: Origin of the employed graphsets.

Graphset	Original benchmark data
GS1	BZR, graphs with nr. 1-200
GS2	BZR, graphs with nr. 201-405
GS3	COX2, graphs with nr. 1-250
GS4	COX2, graphs with nr. 251-467
GS5	DHFR, graphs with nr. 1-250
GS6	DHFR, graphs with nr. 251-500
GS7	DHFR, graphs with nr. 501-756
GS8	NCI1, graphs with nr. mod 13 = 0
GS9	NCI1, graphs with nr. mod 13 = 1
GS10	NCI1, graphs with nr. mod 13 = 2
GS11	NCI1, graphs with nr. mod 13 = 3
GS12	NCI1, graphs with nr. mod 13 = 4
GS13	NCI1, graphs with nr. mod 13 = 5
GS14	NCI1, graphs with nr. mod 13 = 6
GS15	NCI1, graphs with nr. mod 13 = 7
GS16	NCI1, graphs with nr. mod 13 = 8
GS17	NCI1, graphs with nr. mod 13 = 9
GS18	NCI1, graphs with nr. mod 13 = 10
GS19	NCI1, graphs with nr. mod 13 = 11
GS20	NCI1, graphs with nr. mod 13 = 12

Topology of the Compared GNNs . Due to the fact that the GNNs were trained on classification data, and also due to the default settings of the employed `GenWL` implementation of the relaxed WL kernel, both of them contained the following layers:

(i) an input layer, which receives the colourings of vertices, i.e. the components $(\lambda_1,\ldots,\lambda_{d_c})$ of the label λ, no matter whether the label has possibly still other components, i.e. whether $d_c = d_\lambda$ or $d_c < d_\lambda$;

(ii) 5 MPNN layers of the same size, specific for 1-GNN and for GIN;

(iii) an average-pooling layer averaging over vertices of the graph;

(iv) a crossentropy-classification layer.

As to the MPNN layers, we investigated 2 variants of them, differing in size:

(iia) The size of all 5 layers equals the average number of vertices among the graphs in the graphset.

(iib) The size of all 5 layers equals the maximal number of vertices among the graphs in the graphset.

GNN values used for graph representation were obtained as the activities, i.e. the results of somatic operators, in neurons of the MPNN layers. In both compared GNNs, the activities are obtained separately for each vertex of each graph. Therefore, activities for each graph are first averaged over all of its vertices, in accordance with using an average pooling layer during their training. According to the neurons that were actually used to this end, two kinds of graph representation were considered:

(i) representation restricted to activities of neurons of the last MPNN layer, which is a restriction commonly encountered in representation learning by artificial neural networks;

(ii) representation with activities of neurons of all MPNN layers, which is more similar to the representation (24) based on the relaxed WL kernel.

Apart from the GNN topology, the hyperparameter values of all compared methods were set to their defaults in the employed implementation, and if no default was available, to values obtained through slight tuning.

Evaluation of graph representations with respect to their suitability for downstream classification was by means of accuracy on test data obtained in classification using as input each of the representations. To this end, a linear support vector machine (SVM) classifier was employed, in accordance with the default setting in the `GenWL` implementation of the relaxed WL kernel [8]. To increase the reliability of the accuracy assessment, a 10-fold cross-validation was used.

3.3 First Results

The results for both kinds of GNN-based representations for the 4 considered networks, i.e. 1-GNN and GIN with both variants (iia) and (iib) of the MPNN layers, as well as the results for the R-WL*-based representation are presented in Table 2. For each of those representations, the mean and standard deviation of the accuracies on the validation folds are reported, from 10-fold cross-validation on each of the 20 graphsets.

According to Table 2, the highest accuracy has been most frequently, namely 8 times among the 20 employed graphsets, achieved with the representation by activities of the last layer of the the variant (iia) of MPNN layers of a 1-GNN, as well as with the representation by activities of all layers of the the variant (iib) of MPNN layers of a 1-GNN. However, the differences between accuracies in Table 2 are not only due to essential differences between the considered representations, but also due to random influences. To separate the former from the latter requires to assess statistical significance of those differences.

3.4 Assessment of Statistical Significance

To assess the statistical significance of the obtained results, we first tested the basic null hypotheses that the mean classification accuracy for all 9 representations coincides. To this end, we applied the Friedman test with 10 replicates to the results for all 200 validation folds from the cross-validation of SVM-classification on the 20 graphsets. This basic null hypothesis was strongly rejected, with the achieved significance $p = 4 * 10^{-92}$. For the post-hoc

Table 2: Mean and standard deviation of the validation-fold accuracy from a 10-fold cross-validation of classification by an SVM, using as input the representations of graphs obtained from the 1-GNN and GIN networks, as well as from the R-WL* variant of the relaxed WL kernel. MPNN layers (iia) refer to 5 message passing layers of length equal to the average number of vertices among the graphs in the graphset on which the network is being trained, MPNN layers (iib) refer to 5 message passing layers of length equal to the maximal number of vertices among the graphs in that graphset. For each graphset, the representation yielding the highest accuracy is in bold.

Representation	Activities of the last MPNN layer				Activities of all MPNN layers				Relaxed
	MPNN layers (iia)		MPNN layers (iib)		MPNN layers (iia)		MPNN layers (iib)		WL
Graphset	1-GNN	GIN	1-GNN	GIN	1-GNN	GIN	1-GNN	GIN	kernel
	Accuracy [%]: mean ± standard deviation								
GS1	96 ± 4	94 ± 4	$\mathbf{98 \pm 4}$	94 ± 4	96 ± 5	94 ± 3	$\mathbf{98 \pm 3}$	94 ± 5	90 ± 9
GS2	86 ± 8	82 ± 8	88 ± 7	83 ± 9	85 ± 7	86 ± 6	$\mathbf{89 \pm 5}$	84 ± 9	81 ± 9
GS3	82 ± 7	81 ± 3	84 ± 4	82 ± 4	84 ± 5	81 ± 4	$\mathbf{86 \pm 6}$	85 ± 6	82 ± 4
GS4	88 ± 6	81 ± 8	89 ± 6	82 ± 7	89 ± 5	83 ± 3	$\mathbf{90 \pm 8}$	82 ± 6	76 ± 7
GS5	81 ± 9	79 ± 8	$\mathbf{84 \pm 6}$	74 ± 8	80 ± 5	76 ± 7	$\mathbf{84 \pm 5}$	76 ± 8	66 ± 5
GS6	$\mathbf{96 \pm 5}$	85 ± 7	92 ± 5	84 ± 6	$\mathbf{96 \pm 4}$	84 ± 11	94 ± 3	84 ± 5	83 ± 8
GS7	$\mathbf{98 \pm 3}$	93 ± 5	92 ± 3	91 ± 5	$\mathbf{98 \pm 3}$	92 ± 4	92 ± 5	92 ± 5	89 ± 5
GS8	$\mathbf{83 \pm 6}$	79 ± 6	79 ± 7	80 ± 7	$\mathbf{83 \pm 4}$	78 ± 6	79 ± 9	79 ± 6	65 ± 8
GS9	80 ± 6	75 ± 8	93 ± 6	75 ± 7	79 ± 10	76 ± 5	$\mathbf{94 \pm 5}$	73 ± 7	75 ± 7
GS10	$\mathbf{83 \pm 8}$	81 ± 5	66 ± 8	76 ± 7	$\mathbf{83 \pm 6}$	78 ± 6	69 ± 6	77 ± 7	74 ± 7
GS11	73 ± 11	$\mathbf{79 \pm 7}$	75 ± 4	71 ± 8	75 ± 6	78 ± 8	77 ± 7	71 ± 6	71 ± 8
GS12	$\mathbf{93 \pm 4}$	80 ± 6	90 ± 5	80 ± 5	91 ± 4	84 ± 8	90 ± 7	79 ± 7	64 ± 6
GS13	85 ± 4	75 ± 10	85 ± 4	78 ± 8	$\mathbf{86 \pm 7}$	71 ± 9	85 ± 8	74 ± 10	71 ± 8
GS14	$\mathbf{85 \pm 7}$	73 ± 9	81 ± 5	72 ± 9	$\mathbf{85 \pm 6}$	76 ± 9	82 ± 3	71 ± 9	69 ± 7
GS15	$\mathbf{91 \pm 6}$	66 ± 11	79 ± 7	71 ± 6	$\mathbf{91 \pm 4}$	67 ± 7	78 ± 7	67 ± 12	64 ± 6
GS16	81 ± 8	$\mathbf{86 \pm 5}$	75 ± 6	80 ± 7	81 ± 7	85 ± 4	77 ± 7	79 ± 7	69 ± 6
GS17	87 ± 6	78 ± 7	$\mathbf{91 \pm 6}$	74 ± 8	88 ± 6	78 ± 8	$\mathbf{91 \pm 5}$	72 ± 5	68 ± 10
GS18	92 ± 4	78 ± 8	94 ± 4	75 ± 10	93 ± 3	75 ± 7	$\mathbf{95 \pm 3}$	75 ± 8	63 ± 8
GS19	$\mathbf{88 \pm 7}$	72 ± 7	82 ± 6	72 ± 7	87 ± 5	75 ± 6	81 ± 8	73 ± 9	69 ± 6
GS20	85 ± 4	73 ± 7	$\mathbf{90 \pm 5}$	71 ± 9	82 ± 6	75 ± 8	89 ± 9	70 ± 6	66 ± 7

analysis, we employed the Wilcoxon signed rank test with two-sided alternative for all 36 pairs of the investigated representations, because of the inconsistence of the more commonly used mean ranks post-hoc test, to which recently Benavoli et al. pointed out [2]. For correction to multiple hypotheses testing, we used the Holm method [7].

The results of the pairwise comparisons of the 9 investigated representations and of their significance testing are presented in Table 3. A number $n_{a,b}$ in a row of the table corresponding to a representation a and a column corresponding to a representation b states in how many among the graphsets GS1–GS20, the representation a lead to a higher mean classification accuracy than the representation b. If $n_{a,b} > n_{b,a}$ and the difference between the representations a and b is according to the Wilcoxon signed rank test significant at the familywise level 5%, after the Holm correction, then $n_{a,b}$ is in bold.

The boldfaced significant differences in Table 3 reveal that representations originating from the same kind of GNNs never lead to significantly different accuracy, neither those originating from a 1-GNN, nor those originating from a GIN. On the other hand, if one of the representations originates from a 1-GNN and the other from a GIN, then the difference between the accuracies is significant, and similarly if one of them is GNN-based and the other originates from the relaxed WL kernel. Combined with the results in Table 2, this means that any of the four representations originating from the 1-GNN yields most frequently the highest accuracy, whereas the differences between those four representations are not significant. The fact that the accuracies of the four representations originating from the same kind of GNNs were neither for 1-GNN nor for GIN significantly different, suggests to test the stronger hypothesis that the mean classification accuracy of all those four representations coincides. For 1-GNN, the Friedman test indeed did not reject that hypothesis, with quite high achieved significance $p = 0.19$. For GIN, it rejected the hypothesis on the usual significance level 5%, but even on the significance level 4% it did not reject it any more (achieved significance $p = 0.043$).

Table 3: Comparison of the investigated representations. A number $n_{a,b}$ in a row of the table corresponding to a representation a and a column corresponding to a representation b states in how many among the graphsets GS1–GS20, the representation a lead to a higher mean classification accuracy than the representation b. That number is in bold if $n_{a,b} > n_{b,a}$ and the difference between the representations a and b is according to the Wilcoxon singed rank test significant at the familywise level 5%, after the Holm correction. MPNN layers (iia) refer to 5 MPNN layers of length equal to the average number of vertices among the graphs in the graphset on which the network is being trained, MPNN layers (iib) refer to 5 MPNN layers of length equal to the maximal number of vertices among the graphs in that graphset.

Representation			Activities of the last MPNN layer				Activities of all MPNN layers				Relaxed WL kernel
			MPNN layers (iia)		MPNN layers (iib)		MPNN layers (iia)		MPNN layers (iib)		
			1-GNN	GIN	1-GNN	GIN	1-GNN	GIN	1-GNN	GIN	
Activities of the last MPNN layer	MPNN layers (iia)	1-GNN	★	**18**	9	**19**	6	**17**	9	**19**	**19**
		GIN	2	★	4	10	2	9	4	13	**17**
	MPNN layers (iib)	1-GNN	10	**15**	★	**17**	7	**16**	3	**15**	**19**
		GIN	0	6	3	★	0	4	3	9	**17**
Activities of all MPNN layers	MPNN layers (iia)	1-GNN	6	**18**	10	**20**	★	**17**	10	**19**	**20**
		GIN	2	8	3	13	3	★	3	11	**18**
	MPNN layers (iib)	1-GNN	10	**15**	10	**17**	10	**16**	★	**16**	**19**
		GIN	1	5	3	6	1	3	2	★	**18**
Relaxed WL kernel			0	1	1	0	0	1	1	1	★

4 Conclusion and Further Research

This work-in-progress paper experimentally compared 8 variants of graph representations based on two recently proposed graph neural networks inspired by the WL isomorphism test, as well as a representation based on a recent generalization of the WL subtree kernel. They were compared not with respect to their distance in the embeding space, but with respect to downstream classification by means of an SVM. The results of that comparison indicate that the highest classification accuracy is achieved with representations originating from 1-GNN networks. At the same time, the comparison revealed that any two representations originating from different kinds of GNNs always lead to significantly different accuracies, and also the accuracies of a GNN-based representation and of the representation originating from the relaxed WL kernel differ significantly. On the other hand, the four accuracies of representations originating from the same kind of GNNs are never significantly different. Moreover, the data even don't contradict a stronger hypothesis that the mean classification accuracy of all those representations coincides.

The result that the representation based on the relaxed WL kernel was inferior to the GNN-based representations is explainable by the fact that the kernel was designed for dense and structurally more diverse graphs, whereas the reported investigation was performed on simple molecular graphs.

However, the paper presents really only first results that, in our opinion, indicate usefulness of a possible further research in this direction, which we consider interesting due to the crucial role nowadays played by representation learning. Needless to say, such a research would have to be more comprehensive with respect to the employed graph data, as well as with respect to the involved representation methods.

As to the employed data, we used 20 comparatively small disjoint graphsets from only four different benchmark datasets, to be able to assess the statistical significance of the differences between the compared representations. To use instead a similar number of separate benchmarks, would lead to much larger graphsets and would allow to cover a broader spectrum of applications, and to drop the assumption that for the employed data, disjointness is a sufficient condition for statistical independence. The benchmarks for future investigation should include also dense and structurally diverse graphs, i.e. the kind of graphs for which the relaxed WL kernel is intended. In addition, it would be interesting to know how the GNN-based representations change if instead of the colouring components $(\lambda_1, \ldots, \lambda_{d_c})$ of a labelling λ, the complete labeling is used.

As to the involved representation methods, it would be interesting to compare the relaxed WL kernel with methods based on other GNNs inspired by the WL test. Indeed, other variants of the WL subtree kernel have been sufficiently compared with the relaxed WL kernel in [23] and shown to be inferior to it, and several of them have also already been compared with GNN-based methods inspired by the WL test [1, 11, 15]. On the other hand, comparisons among different methods are sporadic [3, 11, 15], and a comparison of some of them with the relaxed WL kernel is, to the best of our knowledge, for the first time reported in this paper.

Acknowledgement

The authors are deeply indebted to Till Hendrik Schulz for repeated discussions and for his help with the relaxed WL kernel and with its implementation GenWL. The research reported in this paper has been supported by the Charles University SVV project 260575 and partially supported by the Czech Science Foundation (GAČR) grant 18-18080S.

References

[1] D. Al-Rfou, R. adn Zelle and B. Perozzi. DDGK: Learning graph representations for deep divergence graph kernels. In *International Wolrd Wide Web Conference*, pages 37–48, 2019.

[2] A. Benavoli, G. Corani, and F. Mangili. Should we really use post-hoc tests based on mean-ranks? *Journal of Machine Learning Research*, 17:1–10, 2016.

[3] Z. Chen, S. Villar, L. Chen, and J. Bruna. On the equivalence between graph isomorphism testing and function approximation with GNNs. In *33rd Conference on Neural Information Processing Systems*, pages 1–19, 2019.

[4] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989.

[5] H. Dell, M. Grohe, and G. Rattan. Lovász meets weisfeiler and leman. In *45th International Colloquium on Automata, Languages, and Programming*, page article no. 40, 2018.

[6] F. Errica, D. Bacciu, and A. Micheli. Theoretically expressive and edge-aware graph learning. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 175–180, 2020.

[7] S. Garcia and F. Herrera. An extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.

[8] *GenWL. A Generalized Weisfeiler-Lehman Graph Kernel*, 2021. https://github.com/mlai-bonn/GenWL.

[9] A. Grover and J. Leskovec. Node2vec: Scalable feature learning for networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–86, 2016.

[10] W.L. Hamilton. *Graph Repesentation Learning*. Morgan & Claypool Publishers, San Rafael, 2020.

[11] N.T. Hoang and T. Maehara. Graph homomorphism convolution. In *37th International Conference on Machine Learning*, pages 1–15, 2020.

[12] K. Hornik. Approximation capabilities of multilayer neural networks. *Neural Networks*, 4:251–257, 1991.

[13] K. Hornik, M. Stichcombe, and H. White. Multilayer feeforward networks are univesal approximators. *Neural Networks*, 2:359–366, 1989.

[14] V. Kůrková. Kolmogorov's theorem and multilayer neural networks. *Neural Networks*, 5:501–506, 1992.

[15] H. Maron, H. Ben-Hamu, H. Serviansky, and Y. Lipman. Provably powerful graph networks. In *33rd Conference on Neural Information Processing Systems*, pages 1–12, 2019.

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. arXiv preprint 1301.3781, 2013.

[17] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *27th Conference on Neural Information Processing Systems*, pages 3111–3119, 2013.

[18] C. Morris, M. Ritzert, M. Fey, W.L. Hamilton, J.E. Lenssen, et al. Weisfeiler and Leman go neural: Higher-order graph neural networks. In *33rd AAAI Conference on Artificial Intelligence*, pages 4602–4609, 2019.

[19] J. Park and I.W. Sandberg. Approximation and radial-basis-function networks. *Neural Computation*, 5:305–316, 1993.

[20] B. Perozzi, R. Al-Rfou, and S. Skiena. DeepWalk| online learning of social representations. In *ACM KDDM*, pages 701–710, 2014.

[21] *PyTorch Geometric Documentation*, 2021. https://pytorch-geometric.readthedocs.io/.

[22] B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, 2002.

[23] T.H. Schulz, T. Horváth, P. Welke, and S. Wrobel. A generalized weisfeiler-lehman graph kernel. ArXiv preprint arXiv:2101.08104.v1, 2021.

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

[25] J.J. Sutherland, L.A. O'Brien, and D.F. Weaver. Spline-fitting with a genetic algorithm: A method for developing classification structure-activity relationships. *Journal of Chemical Information and Computer Science*, 43:1906–1915, 2003.

[26] N. Wale, I.A. Watson, and G. Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14:347–375, 2008.

[27] B. Weisfeiler and A.A. Lehman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, 2:12–16, 1968.

[28] H. White. *Artificial Neural Networks: Approximation and Learning Theory*. Blackwell Publishers, Cambridge, 1992.

[29] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, pages 1–17, 2019.