Weisfeiler–Leman at the margin: When more expressivity matters

Billy J. Franks [*1] **Christopher Morris** [*2] **Ameya Velingker** [3] **Floris Geerts** [4]

Abstract

The Weisfeiler–Leman algorithm (1-WL) is a well-studied heuristic for the graph isomorphism problem. Recently, the algorithm has played a prominent role in understanding the expressive power of message-passing graph neural networks (MPNNs) and being effective as a graph kernel. Despite its success, 1-WL faces challenges in distinguishing non-isomorphic graphs, leading to the development of more expressive MPNN and kernel architectures. However, the relationship between enhanced expressivity and improved generalization performance remains unclear. Here, we show that an architecture's expressivity offers limited insights into its generalization performance when viewed through graph isomorphism. Moreover, we focus on augmenting 1-WL and MPNNs with subgraph information and employ classical margin theory to investigate the conditions under which an architecture's increased expressivity aligns with improved generalization performance. In addition, we show that gradient flow pushes the MPNN's weights toward the maximum margin solution. Further, we introduce variations of expressive 1-WL-based kernel and MPNN architectures with provable generalization properties. Our empirical study confirms the validity of our theoretical findings.

1. Introduction

Graph-structured data are prevalent in application domains ranging from chemo- and bioinformatics (Jumper et al., 2021; Stokes et al., 2020; Wong et al., 2023), combinatorial optimization (Cappart et al., 2021), to image (Simonovsky & Komodakis, 2017) and social-network analysis (Easley & Kleinberg, 2010), underlying the importance of machine learning methods for graphs. Nowadays, there are numerous approaches for machine learning on graphs, most notably those based on *graph kernels* (Borgwardt et al., 2020; Kriege et al., 2020) or *message-passing graph neural networks* (MPNNs) (Gilmer et al., 2017; Scarselli et al., 2009). Here, graph kernels (Shervashidze et al., 2011) based on the 1-*dimensional Weisfeiler–Leman algorithm* (1-WL) (Weisfeiler & Leman, 1968), a well-studied heuristic for the graph isomorphism problem, and corresponding MPNNs (Morris et al., 2019; Xu et al., 2019), have recently advanced the state-of-the-art in supervised vertex- and graph-level learning (Morris et al., 2021).

However, due to 1-WL's limitations in distinguishing non-isomorphic graphs (Arvind et al., 2015; Cai et al., 1992), numerous recent works proposed more expressive extensions of the 1-WL and corresponding MPNNs (Morris et al., 2021). For example, Bouritsas et al. (2020) introduced an approach to enhance the 1-WL and MPNNs by incorporating subgraph information, achieved by labeling vertices based on their structural roles regarding a set of predefined (sub)graphs. Through the careful selection of such graphs, Bouritsas et al. (2020) demonstrated that these enhanced 1-WL and MPNNs variants can effectively discriminate between pairs of non-isomorphic graphs, which the 1-WL and k-WL (Cai et al., 1992), 1-WL's more expressive generalization, cannot. *Furthermore, empirical results (Bouritsas et al., 2020) indicate that this added expressive power often results in improved predictive performance. However, the exact mechanisms underlying this performance improvement remain unclear.*

Although recent work (Morris et al., 2023) has used 1-WL to establish upper and lower bounds on the Vapnik–Chervonenkis (VC) dimension of MPNNs, these findings do not explain the above empirical observations. Specifically, they do not explain the empirical trend that increased expressive power corresponds to enhanced generalization performance while keeping the size of the training set fixed. Concretely, (Morris et al., 2023) demonstrated a strong correlation between the VC dimension of MPNNs and the number of non-isomorphic graphs that 1-WL can differentiate. Consequently, increasing 1-WL's expressive capabilities increases the VC dimension, worsening generalization performance. A parallel argument can be made regarding 1-WL-based kernels.

[*]Equal contribution [1]Department of Computer Science, University of Kaiserslautern-Landau [2]Department of Computer Science, RWTH Aachen University [3]Google Research [4]Department of Computer Science, University of Antwerp. Correspondence to: Billy J. Franks <billy.franks@rptu.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Present work Here, we investigate to what extent the 1-WL and more expressive extensions can be used as a proxy for an architecture's predictive performance. First, we show that data distributions exist such that the 1-WL and corresponding MPNNs distinguish every pair of non-isomorphic graphs with different class labels while no (linear) classifier can do better than random outside of the training set. Hence, we show that the graph isomorphism perspective is too limited to understand MPNNs' generalization properties. Secondly, based on Alon et al. (2021)'s theory of partial concepts, we derive *tight* upper and lower bounds for the VC dimension of the 1-WL-based kernels and corresponding MPNNs parameterized by the *margin* separating the data. In addition, building on Ji & Telgarsky (2019), we show that gradient flow pushes the MPNN's weights toward the maximum margin solution. Thirdly, we show when 1-WL variants using subgraph information can make the data linearly separable, leading to a positive margin. Building on this, we derive conditions under which more expressive 1-WL variants lead to better generalization performance and derive 1-WL variants with favorable generalization properties. Our empirical study confirms the validity of our theoretical findings.

Our theory establishes the first link between increased expressive power and improved generalization performance. Moreover, our results provide the first margin-based lower bounds for MPNNs' VC dimension. *Overall, our results provide new insights into when more expressive power translates into better generalization performance, leading to a more fine-grained understanding of designing expressive MPNNs.*

1.1. Related work

In the following, we discuss relevant related work.

Graph kernels based on the 1-WL Shervashidze et al. (2011) were the first to utilize the 1-WL as a graph kernel. Later, Morris et al. (2017; 2020b; 2022) generalized this to variants of the k-WL. Moreover, Kriege et al. (2016) derived the *Weisfeiler-Leman optimal assignment kernel*, using the 1-WL to compute optimal assignments between vertices of two given graphs; see Appendix A for an extended discussion.

MPNNs Recently, MPNNs (Gilmer et al., 2017; Scarselli et al., 2009) emerged as the most prominent graph representation learning architecture. Notable instances of this architecture include, e.g., Duvenaud et al. (2015); Hamilton et al. (2017), and Veličković et al. (2018), which can be subsumed under the message-passing framework introduced in Gilmer et al. (2017). Recently, connections between MPNNs and Weisfeiler–Leman-type algorithms have been shown (Barceló et al., 2020; Geerts et al., 2021; Morris et al., 2019; Xu et al., 2019). Specifically, Morris et al. (2019) and Xu et al. (2019) showed that the 1-WL limits the expressive power of any possible MPNN architecture in distinguishing non-isomorphic graphs. (Bouritsas et al., 2020) showed how to make MPNNs more expressive by incorporating subgraph information; see Appendix A for an extended discussion on MPNNs and more expressive MPNNs.

Generalization abilities of graph kernels and MPNNs Scarselli et al. (2018) used classical techniques from learning theory (Karpinski & Macintyre, 1997) to show that MPNNs' VC dimension (Vapnik, 1995) with piece-wise polynomial activation functions on a *fixed* graph, under various assumptions, is in $\mathcal{O}(P^2 n \log n)$, where P is the number of parameters and n is the order of the input graph; see also Hammer (2001). Garg et al. (2020) showed that the empirical Rademacher complexity (see, e.g., Mohri et al. (2012)) of a specific, simple MPNN architecture, using sum aggregation, is bounded in the maximum degree, the number of layers, Lipschitz constants of activation functions, and parameter matrices' norms. Most recently, Morris et al. (2023) made progress connecting MPNNs' expressive power and generalization ability via the Weisfeiler–Leman hierarchy. They derived that MPNNs' VC dimension depends tightly on the number of equivalence classes computed by the 1-WL over a given set of graphs. Moreover, they showed that MPNNs' VC dimension depends logarithmically on the number of colors computed by the 1-WL and polynomially on the number of parameters; see Appendix A for an extended discussion on MPNNs' generalization properties.

Margin theory and VC dimension Using the margin as a regularization mechanism dates back to Vapnik & Chervonenkis (1964). Later, the concept of margin was successfully applied to *support vector machines* (SVMs) (Cortes & Vapnik, 1995; Vapnik, 1998) and connected to VC dimension theory; see Mohri et al. (2012) for an overview. Grønlund et al. (2020) derived the so-far tightest generalization bounds for SVMs. Alon et al. (2021) introduced the theory of VC dimension of *partial concepts*, i.e., the hypothesis set allows partial functions and showed, analogous to the standard case, that finite VC dimension implies learnability and vice versa.

2. Background

Let $\mathbb{N} \coloneqq \{1, 2, 3, \dots\}$. For $n \geq 1$, let $[n] \coloneqq \{1, \dots, n\} \subset \mathbb{N}$. We use $\{\!\{\dots\}\!\}$ to denote multisets, i.e., the generalization of sets allowing for multiple instances for each of its elements. For two sets X and Y, let X^Y denote the set of functions mapping from Y to X. Let $S \subset \mathbb{R}^d$, then the *convex hull* $\mathsf{conv}(S)$ is the minimal convex set containing the set S. For $\boldsymbol{p} \in \mathbb{R}^d, d > 0$, and $\varepsilon > 0$, the *ball* $B(\boldsymbol{p}, \varepsilon, d) \coloneqq \{\boldsymbol{s} \in \mathbb{R}^d \mid \|\boldsymbol{p} - \boldsymbol{s}\| \leq \varepsilon\}$. Here, and in the remainder of the paper, $\|\cdot\|$ refers to the 2-*norm* $\|\boldsymbol{x}\| \coloneqq \sqrt{x_1^2 + \cdots + x_d^2}$, for $\boldsymbol{x} \in \mathbb{R}^d$.

Graphs An *(undirected) graph* G is a pair $(V(G), E(G))$ with *finite* sets of *vertices* or *nodes* $V(G)$ and *edges* $E(G) \subseteq \{\{u, v\} \subseteq V(G) \mid u \neq v\}$. For ease of notation, we denote an edge $\{u, v\}$ in $E(G)$ by (u, v) or (v, u). The *order* of a graph G is its number $|V(G)|$ of vertices. If not stated otherwise, we set $n \coloneqq |V(G)|$ and call G an n-order graph. We denote the set of all n-order graphs by $\mathcal{G}_n$. For a graph $G \in \mathcal{G}_n$, we denote its *adjacency matrix* by $\boldsymbol{A}(G) \in \{0, 1\}^{n \times n}$, where $A(G)_{vw} = 1$ if, and only, if $(v, w) \in E(G)$. For a set of nodes $S \subseteq V(G)$, we denote the induced subgraph of G as $G[S] \coloneqq (V(G) \cap S, E(G) \cap S^2)$. We use standard notation and terminology; see Appendix B for details.

Kernels A *kernel* on a non-empty set $\mathcal{X}$ is a symmetric, positive semidefinite function $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Equivalently, a function $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel if there is a *feature map* $\phi \colon \mathcal{X} \to \mathcal{H}$ to a Hilbert space $\mathcal{H}$ with inner product $\langle \cdot, \cdot \rangle$ such that $k(x, y) = \langle \phi(x), \phi(y) \rangle$ for all x and $y \in \mathcal{X}$. We also call $\phi(x) \in \mathcal{H}$ a *feature vector*. A *graph kernel* is a kernel on the set $\mathcal{G}$ of all graphs. In the context of graph kernels, we also refer to a feature vector as a *graph embedding*.

VC Dimension of partial concepts Let $\mathcal{X}$ be a non-empty set. As outlined in Alon et al. (2021), we consider *partial concepts* $\mathbb{H} \subseteq \{0, 1, \star\}^{\mathcal{X}}$, where each concept $c \in \mathbb{H}$ is a *partial* function. That is, if $x \in \mathcal{X}$ such that $c(x) = \star$, then c is *undefined* at x. The *support* of a partial concept $h \in \mathbb{H}$ is the set $\mathsf{supp}(h) \coloneqq \{x \in \mathcal{X} \mid h(x) \neq \star\}$. The VC dimension of (total) concepts (Vapnik, 1995) straightforwardly generalizes to partial concepts. That is, the *VC dimension* of a partial concept class $\mathbb{H}$, denoted $\mathsf{VC}(\mathbb{H})$, is the maximum cardinality of a shattered set $U \coloneqq \{x_1, \ldots, x_m\} \subseteq \mathcal{X}$. Here, the set U is *shattered* if for any $\boldsymbol{\tau} \in \{0, 1\}^m$ there exists $c \in \mathcal{H}$ such that $c(x_i) = \tau_i$, for all $i \in [m]$. In essence, Alon et al. (2021) showed that the standard definition of PAC learnability extends to partial concepts, recovering the equivalence of finite VC dimension and PAC learnability.

Geometric margin classifiers Classifiers with a geometric margin, e.g., *support vector machines* (Cortes & Vapnik, 1995), are a cornerstone of machine learning. A *sample* $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_s, y_s) \in \mathbb{R}^d \times \{0, 1\}$, for $d > 0$, is (r, λ)-*separable* if (1) there exists $\boldsymbol{p} \in \mathbb{R}^d$ and $r > 0$ and a ball $B(\boldsymbol{p}, r, d)$ such that $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_s \in B(\boldsymbol{p}, r, d)$ and (2) the Euclidean distance between $\mathsf{conv}(\{\boldsymbol{x}_i \mid y_i = 0\})$ and $\mathsf{conv}(\{\boldsymbol{x}_i \mid y_i = 1\})$ is at least 2λ. Then, the sample is *linearly separable* with *margin* λ. We define the set of concepts

$$\mathbb{H}_{r,\lambda}(\mathbb{R}^d) \coloneqq \Big\{ h \in \{0, 1, \star\}^{\mathbb{R}^d} \;\Big|\; \forall\, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_s \in \mathsf{supp}(h) \colon (\boldsymbol{x}_1, h(\boldsymbol{x}_1)), \ldots, (\boldsymbol{x}_s, h(\boldsymbol{x}_s)) \text{ is } (r, \lambda)\text{-separable} \Big\}.$$

Alon et al. (2021) showed that the VC dimension of the concept class $\mathbb{H}_{r,\lambda}(\mathbb{R}^d)$ is asymptotically lower- and upper-bounded by r^2/λ^2. Importantly, the above bounds are independent of the dimension d, while standard VC dimension bounds scale linearly with d (Anthony & Bartlett, 2002).

The 1-dimensional Weisfeiler–Leman algorithm The 1-WL or *color refinement* is a well-studied heuristic for the graph isomorphism problem, originally proposed by Weisfeiler & Leman (1968). Intuitively, the algorithm determines if two graphs are non-isomorphic by iteratively coloring or labeling vertices. Formally, let $G = (V(G), E(G), \ell)$ be a labeled graph. In each iteration, $t > 0$, the 1-WL computes a vertex coloring $C^1_t \colon V(G) \to \mathbb{N}$, depending on the coloring of the neighbors. That is, in iteration $t > 0$, we set $C^1_t(v) \coloneqq$

$$\mathsf{RELABEL}\Big(\big(C^1_{t-1}(v), \{\!\{C^1_{t-1}(u) \mid u \in N(v)\}\!\}\big)\Big),$$

for all vertices $v \in V(G)$, where RELABEL injectively maps the above pair to a unique natural number, which has not been used in previous iterations. In iteration 0, the coloring $C^1_0 \coloneqq \ell$ is used. To test whether two graphs G and H are non-isomorphic, we run the above algorithm in "parallel" on both graphs. If the two graphs have a different number of vertices colored $c \in \mathbb{N}$ at some iteration, the 1-WL *distinguishes* the graphs as non-isomorphic. Moreover, if the number of colors between two iterations, t and $(t + 1)$, does not change, i.e., the cardinalities of the images of C^1_t and C^1_{i+t} are equal, the algorithm terminates. For such t, we define the *stable coloring* $C^1_\infty(v) = C^1_t(v)$, for $v \in V(G \dot\cup H)$.

Graph kernels based on the 1-WL Let G be a graph, following Shervashidze et al. (2009), the idea for a kernel based on the 1-WL is to run the 1-WL for $T \geq 0$ iterations, resulting in a coloring function $C^1_t \colon V(G) \to \mathbb{N}$ for each iteration $t \leq T$. Let Σ_t denote the *range* of C^1_t, i.e., $\Sigma_t \coloneqq \{c \mid \exists\, v \in V(G) \colon C^1_t(v) = c\}$. We assume Σ_t to be ordered by the natural order of $\mathbb{N}$, i.e., we assume that Σ_t consists of $c_1 < \cdots < c_{|\Sigma_t|}$. After each iteration, we compute a *feature vector* $\phi_t(G) \in \mathbb{R}^{|\Sigma_t|}$ for each graph G. Each component $\phi_t(G)_i$ counts the number of occurrences of vertices of G labeled by $c_i \in \Sigma_t$. The overall feature vector $\phi_{\mathsf{WL}}(G)$ is defined as the concatenation of the feature vectors of all T iterations, i.e., $\phi^{(T)}_{\mathsf{WL}}(G) \coloneqq \big[\phi_0(G), \ldots, \phi_T(G)\big]$, where $[\ldots]$ denote column-wise vector concatenation. This results in the kernel $k^{(T)}_{\mathsf{WL}}(G, H) \coloneqq \langle \phi^{(T)}_{\mathsf{WL}}(G), \phi^{(T)}_{\mathsf{WL}}(H) \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the standard inner product. We further define the *normalized* 1-WL *feature vector* $\overline{\phi^{(T)}_{\mathsf{WL}}}(G) \coloneqq \phi^{(T)}_{\mathsf{WL}}(G) / \|\phi^{(T)}_{\mathsf{WL}}(G)\|$ obtained by normalizing the 1-WL feature vector to unit length.

Weisfeiler–Leman optimal assignment kernel Based on the 1-WL, Kriege et al. (2016) defined the *Weisfeiler–Leman*

optimal assignment kernel (1-WLOA), which computes an optimal assignment between the colors computed by the 1-WL for all iterations; see Kriege et al. (2016) for details. Given two graphs G and H and let $T \geq 0$, the 1-WLOA computes

$$k_{\mathsf{WLOA}}(G, H) \coloneqq \sum_{t \in [T] \cup \{0\}} \sum_{c \in \Sigma_t} \min(\phi_t(G)_c, \phi_t(H)_c).$$

Observe that for a fixed but arbitrary n, we can compute a corresponding finite-dimensional feature map $\phi^{(T)}_{\mathsf{WLOA}}$ for the set of n-order graphs. From the theory developed in Kriege et al. (2016), it follows that the 1-WLOA kernel has the same expressive power as the 1-WL in distinguishing non-isomorphic graphs.

More expressive variants of the 1-WL It is easy to see that the 1-WL cannot distinguish all pairs of non-isomorphic graphs (Arvind et al., 2015; Cai et al., 1992). However, there exists a large set of more expressive extensions of the 1-WL, which have been successfully leveraged as kernel or neural architectures (Morris et al., 2021). Moreover, empirical results suggest that such added expressive power often translates into increased predictive performance. Nonetheless, the precise mechanisms underlying this performance boost remain unclear.

In the following, we define a simple, more expressive modification of the 1-WL, the 1-WL$_{\mathcal{F}}$. It is a simplified variant of the algorithms defined in Bouritsas et al. (2020), which does not account for orbit information. Let G be a graph and $\mathcal{F}$ be a finite set of graphs. For $F \in \mathcal{F}$, we define a vertex labeling $\ell_F \colon V(G) \to \mathbb{N}$ such that $\ell_F(v) = \ell_F(w)$ if, and only, if there exists $X_v \subseteq V(G)$ with $v \in X_v$ and $X_w \subseteq V(G)$ with $w \in X_w$ such that $G[X_v] \simeq F$ and $G[X_w] \simeq F$. In other words, ℓ_F encodes the presence of subgraphs $G[X_v]$ in G, isomorphic to F and containing vertex v. Furthermore, we define the vertex labeling $\ell_{\mathcal{F}} \colon V(G) \to \mathbb{N}$, where $\ell_{\mathcal{F}}(v) = \ell_{\mathcal{F}}(w)$ if, and only, if, for all $F \in \mathcal{F}$, $\ell_F(v) = \ell_F(w)$. Finally, for $t \geq 0$, we define the vertex coloring $C^{1,\mathcal{F}}_t \colon V(G) \to \mathbb{N}$, where $C^{1,\mathcal{F}}_0(v) \coloneqq \ell_{\mathcal{F}}(v)$ and $C^{1,\mathcal{F}}_t(v) \coloneqq$

$$\mathsf{RELABEL}\Big(\big(C^{1,\mathcal{F}}_{t-1}(v), \{\!\!\{C^{1,\mathcal{F}}_{t-1}(u) \mid u \in N(v)\}\!\!\}\big)\Big),$$

for $v \in V(G)$. Hence, the 1-WL$_{\mathcal{F}}$ only differs from the 1-WL at the initialization step. In Proposition 24, we show that the 1-WL$_{\mathcal{F}}$ is more expressive than the 1-WL. We can also define a 1-WLOA variant of the 1-WL$_{\mathcal{F}}$, which we denote by 1-WLOA$_{\mathcal{F}}$. See Appendix D for how to derive kernels based on the 1-WL$_{\mathcal{F}}$.

Message-passing graph neural networks Intuitively, MPNNs learn a vectorial representation, i.e., a d-dimensional real-valued vector, representing each vertex in a graph by aggregating information from neighboring vertices. Formally, let $G = (V(G), E(G), \ell)$ be a labeled graph with initial vertex features $\boldsymbol{h}^{(0)}_v \in \mathbb{R}^d$ that are *consistent* with ℓ. That is, each vertex v is annotated with a feature $\boldsymbol{h}^{(0)}_v \in \mathbb{R}^d$ such that $\boldsymbol{h}^{(0)}_v = \boldsymbol{h}^{(0)}_u$ if, and only, if $\ell(v) = \ell(u)$. An example is a one-hot encoding of the labels $\ell(u)$ and $\ell(v)$. An MPNN architecture consists of a stack of neural network layers, i.e., a composition of permutation-equivariant parameterized functions. Following, Scarselli et al. (2009) and Gilmer et al. (2017), in each layer, $t > 0$, we compute vertex features $\boldsymbol{h}^{(t)}_v \coloneqq$

$$\mathsf{UPD}^{(t)}\Big(\boldsymbol{h}^{(t-1)}_v, \mathsf{AGG}^{(t)}\big(\{\!\!\{\boldsymbol{h}^{(t-1)}_u \mid u \in N(v)\}\!\!\}\big)\Big) \in \mathbb{R}^d,$$

for each $v \in V(G)$, where $\mathsf{UPD}^{(t)}$ and $\mathsf{AGG}^{(t)}$ may be differentiable parameterized functions, e.g., neural networks. In the case of graph-level tasks, e.g., graph classification, one uses

$$\boldsymbol{h}_G \coloneqq \mathsf{READOUT}\big(\{\!\!\{\boldsymbol{h}^{(L)}_v \mid v \in V(G)\}\!\!\}\big) \in \mathbb{R}^d, \qquad (1)$$

to compute a single vectorial representation based on learned vertex features after iteration L. Again, READOUT may be a differentiable parameterized function. To adapt the parameters of the above three functions, they are optimized end-to-end, usually through a variant of stochastic gradient descent, e.g., Kingma & Ba (2015), together with the parameters of a neural network used for classification or regression.

More expressive MPNNs Since the expressive power of MPNNs is strictly limited by the 1-WL in distinguishing non-isomorphic graphs (Morris et al., 2019; Xu et al., 2019), a large set of more expressive extensions of MPNNs (Morris et al., 2021) exists. Here, we introduce the *MPNN$_{\mathcal{F}}$ architecture*, an MPNN variant of the 1-WL$_{\mathcal{F}}$; see Section 2. In essence, an MPNN$_{\mathcal{F}}$ is a standard MPNN, where we set the initial features consistent with the initial vertex-labeling of the 1-WL$_{\mathcal{F}}$, e.g., one-hot encodings of $\ell_{\mathcal{F}}$. Following Morris et al. (2019), it is straightforward to derive an MPNN$_{\mathcal{F}}$ architecture that has the same expressive power as the 1-WL$_{\mathcal{F}}$ in distinguishing non-isomorphic graphs.

Notation In the subsequent sections, we use the following notation for MPNNs. We denote the class of all (labeled) graphs by $\mathcal{G}$. For $d, l > 0$, we denote the class of MPNNs using summation for aggregation, and such that update and readout functions are *multilayer perceptrons* (MLPs), all of a width of at most d, by $\mathsf{MPNN}_{\mathsf{mlp}}(d, L)$. We refer to elements in $\mathsf{MPNN}_{\mathsf{mlp}}(d, L)$ as *simple MPNNs*; see Appendix C for details. We stress that simple MPNNs are already expressive enough to be equivalent to the 1-WL in distinguishing non-isomorphic graphs (Morris et al., 2019). The class $\mathsf{MPNN}_{\mathsf{mlp},\mathcal{F}}(d, L)$ is defined similarly, based on MPNN$_{\mathcal{F}}$s.

3. When more expressivity matters

We start by investigating under which conditions using more expressive power leads to better generalization performance and, when not, using the *data's margin*. To this aim, we first prove lower and upper bounds on the VC dimension of 1-WL-based kernels, MPNNs, and their more expressive generalizations.

Margin-based bounds on the VC dimension We first derive a general condition to prove margin-based lower and upper bounds. For a subset $\mathbb{S} \subseteq \mathbb{R}^d, d > 0$, we consider the following set of partial concepts from $\mathbb{S}$ to $\{0, 1, \star\}$,

$$\mathbb{H}_{r,\lambda}(\mathbb{S}) \coloneqq \Big\{ h \in \{0,1,\star\}^{\mathbb{S}} \;\Big|\; \forall \boldsymbol{x}_1, \ldots, \boldsymbol{x}_s \in \mathsf{supp}(h) \colon (\boldsymbol{x}_1, h(\boldsymbol{x}_1)), \ldots, (\boldsymbol{x}_s, h(\boldsymbol{x}_s)) \text{ is } (r,\lambda)\text{-separable} \Big\}.$$

For the upper bound, since $\mathbb{S} \subseteq \mathbb{R}^d$, the VC dimension of $\mathbb{H}_{r,\lambda}(\mathbb{S})$ is upper-bounded by the VC dimension of $\mathbb{H}_{r,\lambda}(\mathbb{R}^d)$. As already mentioned, the latter is known to be bounded by r^2/λ^2 (Bartlett & Shawe-Taylor, 1999; Alon et al., 2021). For the lower bound, the following lemma, implicit in Alon et al. (2021), states sufficient conditions for $\mathbb{S}$ such that the VC dimension of $\mathbb{H}_{r,\lambda}(\mathbb{S})$ is also lower-bounded by r^2/λ^2.

Lemma 1. Let $\mathbb{S} \subseteq \mathbb{R}^d$. If $\mathbb{S}$ contains $m \coloneqq \lfloor r^2/\lambda^2 \rfloor$ vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m \in \mathbb{R}^d$ with $\boldsymbol{b}_i \coloneqq (\boldsymbol{b}_i^{(1)}, \boldsymbol{b}_i^{(2)})$ and $\boldsymbol{b}_1^{(2)}, \ldots, \boldsymbol{b}_m^{(2)}$ being pairwise orthogonal, $\|\boldsymbol{b}_i\| = r'$, and $\|\boldsymbol{b}_i^{(2)}\| = r$, then $\mathsf{VC\text{-}dim}(\mathbb{H}_{r',\lambda}(\mathbb{S})) \in \Theta(r^2/\lambda^2)$.

Next, we derive lower- and upper-bounds on the VC dimension of graphs separable by some graph embedding, e.g., the 1-WL kernel. For $n, d > 0$, let $\mathcal{E}(n,d)$ be a class of graph embedding methods consisting of mappings from $\mathcal{G}_n$ to $\mathbb{R}^d$, e.g., 1-WL feature vectors. A (graph) sample $(G_1, y_1), \ldots, (G_s, y_s) \in \mathcal{G}_n \times \{0,1\}$ is (r,λ)-$\mathcal{E}(n,d)$-*separable* if there is an embedding $\mathsf{emb} \in \mathcal{E}(n,d)$ such that $(\mathsf{emb}(G_1), y_1), \ldots, (\mathsf{emb}(G_s), y_s) \in \mathbb{R}^d \times \{0,1\}$ is (r,λ)-separable, resulting in the set of partial concepts $\mathbb{H}_{r,\lambda}(\mathcal{E}(n,d)) \coloneqq$

$$\Big\{ h \in \{0,1,\star\}^{\mathcal{G}_n} \;\Big|\; \forall G_1, \ldots, G_s \in \mathsf{supp}(h) \colon (G_1, h(G_1)), \ldots, (G_s, h(G_s)) \text{ is } (r,\lambda)\text{-}\mathcal{E}(n,d)\text{-separable} \Big\}.$$

Now, consider the subset $\mathbb{S}(n,d) \coloneqq \{\mathsf{emb}(G) \in \mathbb{R}^d \mid G \in \mathcal{G}_n, \mathsf{emb} \in \mathcal{E}(n,d)\}$ of $\mathbb{R}^d$. It is clear that the VC dimension of $\mathbb{H}_{r,\lambda}(\mathcal{E}(n,d))$ is equal to the VC dimension of $\mathbb{H}_{r,\lambda}(\mathbb{S}(n,d))$, which in turn is upper-bounded by r^2/λ^2. We next use Lemma 1 to obtain lower bounds on the VC dimension of $\mathbb{H}_{r,\lambda}(\mathcal{E}(n,d))$ for specific classes of embeddings.

1-WL-based embeddings We first consider the class of graph embeddings obtained by the 1-WL feature map after $T \geq 0$ iterations, i.e., $\mathcal{E}_{\mathsf{WL}}(n, d_T) \coloneqq \{G \mapsto \phi_{\mathsf{WL}}^{(T)}(G) \mid G \in \mathcal{G}_n\}$ and its normalized counterpart $\bar{\mathcal{E}}_{\mathsf{WL}}(n, d_T) \coloneqq \{G \mapsto \overline{\phi_{\mathsf{WL}}^{(T)}}(G) \mid G \in \mathcal{G}_n\}$, where d_T is the dimension of the corresponding Hilbert space after T rounds of 1-WL; see Section 2 for details. The following result shows that the VC dimension of the normalized and unnormalized 1-WL kernel can be lower- and upper-bounded in the margin λ, the number of iterations, and the number of vertices.

Theorem 2. For any $T, \lambda > 0$, $r = \sqrt{T+1}n$, and $n \geq r^2/\lambda^2$, we have $\mathsf{VC\text{-}dim}(\mathbb{H}_{r,\lambda}(\mathcal{E}_{\mathsf{WL}}(n, d_T))) \in \Theta(r^2/\lambda^2)$. Further, for $r = \sqrt{T/(T+1)}$ and $n \geq r^2/\lambda^2$, we have $\mathsf{VC\text{-}dim}(\mathbb{H}_{1,\lambda}(\bar{\mathcal{E}}_{\mathsf{WL}}(n, d_T))) \in \Theta(1/\lambda^2)$.

In Appendix E.3.1, we derive margin-based bounds for graphs with a finite number of colors under the 1-WL, circumventing the dependence on the order in the above result.

Further, by defining $\mathcal{E}_{\mathsf{WL},\mathcal{F}}(n, d_T)$, $\mathcal{E}_{\mathsf{WLOA}}(n, d_T)$, and $\mathcal{E}_{\mathsf{WLOA},\mathcal{F}}(n, d_T)$ analogously, we can show the same or similar results for the 1-$\mathsf{WL}_{\mathcal{F}}$, 1-WLOA, and 1-$\mathsf{WLOA}_{\mathcal{F}}$. The only difference is that $\|\phi_{\mathsf{WL}}^{(t)}(G_i)\| \neq \|\phi_{\mathsf{WLOA}}^{(t)}(G_i)\|$ and thus the radii and bounds change slightly. Concretely, for the 1-$\mathsf{WL}_{\mathcal{F}}$, we get an identical dependency on the margin λ, the number of iterations, and the number of vertices.

Corollary 3. Let $\mathcal{F}$ be a finite set of graphs. For any $T, \lambda > 0$, $r = \sqrt{T+1}n$, and $n \geq r^2/\lambda^2$, we have, $\mathsf{VC\text{-}dim}(\mathbb{H}_{r,\lambda}(\mathcal{E}_{\mathsf{WL},\mathcal{F}}(n, d_T))) \in \Theta(r^2/\lambda^2)$. Further, for $r = \sqrt{T/(T+1)}$ and $n \geq r^2/\lambda^2$, we have $\mathsf{VC\text{-}dim}(\mathbb{H}_{1,\lambda}(\bar{\mathcal{E}}_{\mathsf{WL},\mathcal{F}}(n, d_T))) \in \Theta(1/\lambda^2)$.

Similarly, by changing the radii from $\sqrt{T}n$ to $\sqrt{Tn}$, we get the following results for the 1-WLOA and 1-$\mathsf{WLOA}_{\mathcal{F}}$ kernel.

Proposition 4. For any $T, \lambda > 0$, $r = \sqrt{(T+1)n}$, and $n \geq r^2/\lambda^2$, we have $\mathsf{VC\text{-}dim}(\mathbb{H}_{r,\lambda}(\mathcal{E}_{\mathsf{WLOA}}(n, d_T))) \in \Theta(r^2/\lambda^2)$. Further, for $r = \sqrt{T/(T+1)}$ and $n \geq r^2/\lambda^2$, we have $\mathsf{VC\text{-}dim}(\mathbb{H}_{1,\lambda}(\bar{\mathcal{E}}_{\mathsf{WLOA}}(n, d_T))) \in \Theta(1/\lambda^2)$.

Corollary 5. Let $\mathcal{F}$ be a finite set of graphs. For any $T, \lambda > 0$, for $r = \sqrt{(T+1)n}$, and $n \geq r^2/\lambda^2$, we have, $\mathsf{VC\text{-}dim}(\mathbb{H}_{r,\lambda}(\mathcal{E}_{\mathsf{WLOA},\mathcal{F}}(n, d_T))) \in \Theta(r^2/\lambda^2)$. Further, for $r = \sqrt{T/(T+1)}$ and $n \geq r^2/\lambda^2$, we have $\mathsf{VC\text{-}dim}(\mathbb{H}_{1,\lambda}(\bar{\mathcal{E}}_{\mathsf{WLOA},\mathcal{F}}(n, d_T))) \in \Theta(1/\lambda^2)$.

Therefore, using $\mathcal{F}$ permits the above statements to be feasible for smaller values of n or λ.

Margin-based bounds on the VC dimension of MPNNs and more expressive architectures In the following, we lift the above results to MPNNs. Assume a fixed but arbitrary number of layers $T \geq 0$, vertices $n > 0$, and an embedding dimension $d > 0$. In addition, we denote the following class of graph embeddings $\mathcal{E}_{\mathsf{MPNN}}(n, d, T) \coloneqq \{G \mapsto m(G) \mid G \in \mathcal{G}_n \text{ and } m \in \mathsf{MPNN}_{\mathsf{mlp}}(d, T)\}$, i.e., the set

of d-dimensional vectors computable by simple T-layer MPNNs over the set of n-order graphs. Now, the following result lifts Theorem 2 to MPNNs.

Proposition 6. For any $n, T > 0$, sufficiently large $d > 0$, and $r = \sqrt{T+1}n$ and $n \geq r^2/\lambda^2$, we have, $\mathsf{VC\text{-}dim}(\mathbb{H}_{r,\lambda}(\mathcal{E}_{\mathsf{MPNN}}(n,d,T))) \in \Theta(r^2/\lambda^2)$. Further, for $r = \sqrt{T/(T+1)}$ and $n \geq r^2/\lambda^2$, we have, $\mathsf{VC\text{-}dim}(\mathbb{H}_{1,\lambda}(\mathcal{E}_{\mathsf{MPNN}}(n,d,T))) \in \Theta(1/\lambda^2)$.

Moreover, we can lift Corollary 3 to $\mathrm{MPNN}_{\mathcal{F}}$ architectures by defining $\mathcal{E}_{\mathsf{MPNN},\mathcal{F}}(n,d,T)$ analogously to the above.

Corollary 7. Let $\mathcal{F}$ be a finite set of graphs. For any $n, T > 0$, sufficiently large $d > 0$, and $r = \sqrt{T+1}n$ and $n \geq r^2/\lambda^2$, we have $\mathsf{VC\text{-}dim}(\mathbb{H}_{r,\lambda}(\mathcal{E}_{\mathsf{MPNN},\mathcal{F}}(n,d,T))) \in \Theta(r^2/\lambda^2)$. For $r = \sqrt{T/(T+1)}$ and $n \geq r^2/\lambda^2$, we have $\mathsf{VC\text{-}dim}(\mathbb{H}_{1,\lambda}(\mathcal{E}_{\mathsf{MPNN},\mathcal{F}}(n,d,T))) \in \Theta(1/\lambda^2)$.

We can also lift the results to the MPNN versions of the 1-WLOA and 1-$\mathsf{WLOA}_{\mathcal{F}}$; see the appendix for details. The above results are somewhat restrictive since we only consider MPNNs that behave like linear classifiers by definition of the considered functions. The above implies that the upper bound does not hold for general MPNNs since they can separate non-linearly separable data under mild conditions.

Implications of the results Previous lower and upper bounds only considered the feature space's dimensionality, implying worse generalization performance for more expressive variants of the 1-WL, not aligned with empirical results, e.g., (Bouritsas et al., 2020). Our results show that more expressive power only sometimes results in worse generalization properties. Hence, a more fine-grained analysis is needed to understand when more expressive power, e.g., through the 1-$\mathsf{WL}_{\mathcal{F}}$ or 1-$\mathsf{WLOA}_{\mathcal{F}}$, improves generalization performance. For example, if more expressive power makes the data linearly separable, leading to a positive margin, or increases the margin, our results imply improved generalization performance. In fact, in the following, we leverage our results to understand when more expressivity leads to linear separability and an increased margin. Hence, our results indicate that the data's margin can be used as a yardstick to assess the generalization properties of Weisfeiler–Leman-based kernels, MPNNs, and their more expressive variants in a more fine-grained and data-dependent manner.

3.1. Examples of when more power separates the data

We next aim to understand when 1-WL's more expressive variants, such as the 1-$\mathsf{WL}_{\mathcal{F}}$, can linearly separate the data, resulting in a positive margin. Therefore, we first derive data distributions where the 1-WL kernel cannot separate the data points, while 1-$\mathsf{WL}_{\mathcal{F}}$ separates them with the largest possible margin in the case of normalized feature vectors.

Proposition 8. For every $n \geq 6$, there exists a pair of non-isomorphic n-order graphs (G_n, H_n) and a graph F such that, for $\mathcal{F} \coloneqq \{F\}$ and for all number of rounds $T \geq 0$,

$$\left\| \overline{\phi^{(T)}_{\mathsf{WL}}}(G_n) - \overline{\phi^{(T)}_{\mathsf{WL}}}(H_n) \right\| = 0, \text{ and}$$
$$\left\| \overline{\phi^{(T)}_{\mathsf{WL},\mathcal{F}}}(G_n) - \overline{\phi^{(T)}_{\mathsf{WL},\mathcal{F}}}(H_n) \right\| = \sqrt{2}.$$

Moreover, we can also lift the above result to MPNN and $\mathrm{MPNN}_{\mathcal{F}}$ architectures; see Proposition 41 in the appendix.

However, more than *merely distinguishing* the graphs based on their structure is often required. The following result shows that data distributions exist such that the 1-WL kernel can perfectly separate each pair of non-isomorphic graphs while *not being able to separate the data linearly*. In fact, the construction implies data distributions where the 1-WL kernel cannot do better than random guessing on the test set, which we also empirically verify in Section 5.

Proposition 9. For every $n \geq 10$, there exists a set of pairwise non-isomorphic (at most) n-order graphs S, a concept $c\colon S \to \{0,1\}$, and a graph F, such that the graphs in the set S are (1) pair-wise distinguishable by 1-WL after one round; (2) are *not* linearly separable under the normalized 1-WL feature vector $\overline{\phi^{(T)}_{\mathsf{WL}}}$, concerning the concept c, for any $T \geq 0$, and (3) are linearly separable under the normalized 1-$\mathsf{WL}_{\mathcal{F}}$ feature vector $\overline{\phi^{(T)}_{\mathsf{WL},\mathcal{F}}}$, concerning the concept c for all $T \geq 0$, where $\mathcal{F} \coloneqq \{F\}$. The results also work for the unnormalized case.

In addition, we can derive more general results when placing stronger conditions on the data distribution; see Proposition 43 in the appendix.

3.2. Examples of when more power shrinks the margin

While the previous results showed that more expressive power can make the data linearly separable and improve generalization performance, adding expressive power might also decrease the data's margin. The following result shows that data distributions exist such that more expressive power leads to a smaller margin, implying, by Section 3, a worsened generalization.

Proposition 10. For every $n \geq 10$, there exists a pair of $2n$-order graphs (G_n, H_n) and a graph F, such that, for $\mathcal{F} \coloneqq \{F\}$ and for all number of rounds $T > 0$, it holds that

$$\left\| \overline{\phi^{(T)}_{\mathsf{WL},\mathcal{F}}}(G_n) - \overline{\phi^{(T)}_{\mathsf{WL},\mathcal{F}}}(H_n) \right\| < \left\| \overline{\phi^{(T)}_{\mathsf{WL}}}(G_n) - \overline{\phi^{(T)}_{\mathsf{WL}}}(H_n) \right\|.$$

The above results easily generalize to other 1-$\mathsf{WL}_{\mathcal{F}}$-based kernels, i.e., more expressive power does not always result in increased generalization performance. Hence, in the following subsection, we derive precise conditions when more expressivity provably leads to better generalization.

3.3. When more power grows the margin

Here, we study when added expressive power provably leads to improved generalization performance. We start with the 1-$\mathsf{WL}_{\mathcal{F}}$ and then move to the 1-$\mathsf{WLOA}_{\mathcal{F}}$, where more interesting results can be shown.

The 1-$\mathsf{WL}_{\mathcal{F}}$ kernel The following result shows that, under some assumptions, data distributions exist such that the 1-$\mathsf{WL}_{\mathcal{F}}$ kernel leads to a larger margin than the 1-WL kernel.

Proposition 11. Let $n > 0$ and G_n and H_n be two *connected* n-order graphs. Further, let $\mathcal{F} := \{F\}$ such that there is at least one vertex in $V(G_n)$ contained in a subgraph of G_n isomorphic to the graph F. For the graph H_n, no such vertices exist. Further, let $T \geq 0$ be the number of rounds to reach the stable partition of G_n and H_n under 1-WL, and assume $(\phi^{(T)}_{\mathsf{WL}}(G_n), 1), (\phi^{(T)}_{\mathsf{WL}}(H_n), 0)$ is (r_1, λ_1)-separable, with margin $\lambda_1 < \sqrt{2n}$. Then, $(\phi^{(T)}_{\mathsf{WL},\mathcal{F}}(G_n), 1), (\phi^{(T)}_{\mathsf{WL},\mathcal{F}}(H_n), 0)$ is (r_2, λ_2)-separable, with margin $r_2 \leq r_1$ and $\lambda_2 \geq \lambda_1$.

Hence, in terms of generalization properties, we observe that $r_1^2/\lambda_1^2 \geq r_2^2/\lambda_2^2$ and hence we obtain lower margin-based bounds by using $\mathcal{F}$.

The 1-$\mathsf{WLOA}_{\mathcal{F}}$ kernel It is challenging to improve Proposition 11, i.e., to derive weaker conditions such that 1-$\mathsf{WL}_{\mathcal{F}}$ provably leads to an increase of the margin over the 1-WL kernel. This becomes more feasible, however, for the 1-$\mathsf{WLOA}_{\mathcal{F}}$ kernel. Since, by Proposition 23 in the appendix, the 1-$\mathsf{WL}_{\mathcal{F}}$ computes a finer color partition than the 1-WL, pairwise distances can not decrease compared to the 1-WLOA.

Proposition 12. Let $\mathcal{F}$ be a finite set of graphs. Given two graphs G and H, we have the inequality

$$\left\|\phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G) - \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(H)\right\| \geq \left\|\phi^{(T)}_{\mathsf{WLOA}}(G) - \phi^{(T)}_{\mathsf{WLOA}}(H)\right\|.$$

we can derive conditions under which the 1-$\mathsf{WLOA}_{\mathcal{F}}$ leads to a strict margin increase. That is, we get a strict increase in distances if, and only, if the 1-$\mathsf{WL}_{\mathcal{F}}$ splits up a color c under 1-WL such that the occurrences of this color c are larger or equal in one graph over the other, while for at least one of the resulting colors under 1-$\mathsf{WL}_{\mathcal{F}}$, refining the color c, the relation is strictly reversed.

Proposition 13. Let G and H be n-order graphs and let $\mathcal{F}$ be a finite set of graphs and let $T \geq 0$, and $C_{\mathcal{F}}(c)$ be the set of colors that color c under 1-WL is split into under 1-$\mathsf{WL}_{\mathcal{F}}$, i.e., $\phi_t(G)_c = \sum_{c' \in C_{\mathcal{F}}(c)} \phi_{\mathcal{F},t}(G)_{c'}$. The following statements are equivalent,

1. $\left\|\phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G) - \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(H)\right\| > \left\|\phi^{(T)}_{\mathsf{WLOA}}(G) - \phi^{(T)}_{\mathsf{WLOA}}(H)\right\|$.
2. There exists $t \in [T] \cup \{0\}$ and $c \in \Sigma_t$ such that $\neg(\phi_t(G)_c \geq \phi_t(H)_c \Leftrightarrow \forall\, c' \in C_{\mathcal{F}}(c) \colon \phi_{\mathcal{F},t}(G)_{c'} \geq \phi_{\mathcal{F},t}(H)_{c'})$.

Specifically, this implies that using Proposition 13 as an assumption on the distances between graphs within one class and between two classes, respectively, implies a margin increase via Theorem 47 in the appendix. Then Theorem 29 and Theorem 30 in the appendix imply a decrease in VC-dimension and consequently an increase in generalization performance when using $\mathcal{F}$. See Corollary 49 in the appendix for an example where the above conditions are met.

4. Large margins and gradient flow

Proposition 6 and Corollary 7 (in the appendix) ensure the existence of parameter assignment such that MPNN and MPNN$_{\mathcal{F}}$ architectures generalize. However, it remains unclear how to find them. Hence, building on the results in Ji & Telgarsky (2019), we now show that, under some assumptions, MPNNs exhibit an "alignment" property whereby gradient flow pushes the network's weights toward the maximum margin solution.

Formal setup We consider MPNNs following Section 2 and consider graph classification tasks using a readout layer. We make some simplifying assumptions and consider *linear* MPNNs. That is, set the aggregation function AGG to summation and UPD at layer i is summation followed by a dense layer with trainable weight matrix $\boldsymbol{W}^{(i)} \in \mathbb{R}^{d_i \times d_{i-1}}$. Let G be an n-order graph, if we pack the node embeddings $\boldsymbol{h}_v^{(i)}$ into an $d_i \times n$ matrix $\boldsymbol{X}^{(i)}$ whose v^{th} column is $\boldsymbol{h}_v^{(i)}$, then

$$\boldsymbol{X}^{(i+1)} = \boldsymbol{W}^{(i+1)} \boldsymbol{X}^{(i)} \boldsymbol{A}'(G),$$

where $\boldsymbol{A}'(G) := \boldsymbol{A}(G) + \boldsymbol{I}_n$, $\boldsymbol{I}_n \in \mathbb{R}^{n \times n}$ is the n-dimensional identity matrix, and $\boldsymbol{X} = \boldsymbol{X}^{(0)}$ is the $d_0 \times n$ matrix whose columns correspond to vertices' initial features; we also write $d = d_0$. For the permutation-invariant readout layer, we use simple summation of the final node embeddings and assume that $\boldsymbol{X}^{(L)}$ is transformed into a prediction $\hat{y}$ as follows,

$$\hat{y} = \mathsf{READOUT}\big(\boldsymbol{X}^{(L)}\big) = \boldsymbol{X}^{(L)} \cdot \mathbf{1}_n.$$

Since we desire a scalar output, we will have $d_L = 1$.

Suppose our training dataset is $\{(G_i, \boldsymbol{X}_i, y_i)\}_{i=1}^k$, where $\boldsymbol{X}_i \in \mathbb{R}^{d \times n_i}$ is a set of d-dimensional node features over an n_i-order graph G_i with order n_i, and $y_i \in \{-1, +1\}$ for all i. We use a loss function ℓ with the following assumption.

Assumption 14. The loss function $\ell \colon \mathbb{R} \to \mathbb{R}^+$ has a continuous derivative ℓ' such that $\ell'(x) < 0$ for all x, $\lim_{x \to -\infty} \ell(x) = \infty$, and $\lim_{x \to \infty} \ell(x) = 0$.

The empirical risk induced by the MPNN is

$$\mathcal{R}(\boldsymbol{W}^{(L)}, \ldots, \boldsymbol{W}^{(1)}) = \frac{1}{k} \sum_{i=1}^{k} \ell(y_i, \hat{y}_i) = \frac{1}{k} \sum_{i=1}^{k} \ell(\boldsymbol{W}_{\text{prod}} \boldsymbol{Z}_i \boldsymbol{A}'(G)^L \mathbf{1}_{n_i}),$$

where $\boldsymbol{W}_{\text{prod}} = \boldsymbol{W}^{(L)} \boldsymbol{W}^{(L-1)} \cdots \boldsymbol{W}^{(1)}$, and $\boldsymbol{Z}_i = y_i \boldsymbol{X}_i$.

We consider gradient flow and gradient descent. In gradient flow, the evolution of $\boldsymbol{W} = (\boldsymbol{W}^{(L)}, \boldsymbol{W}^{(L-1)}, \ldots, \boldsymbol{W}^{(1)})$ is given by $\{\boldsymbol{W}(t) \colon t \geq 0\}$, where there is an initial state $\boldsymbol{W}(0)$ at $t = 0$, and

$$\frac{d\boldsymbol{W}(t)}{dt} = -\nabla \mathcal{R}(\boldsymbol{W}(t)).$$

We make one additional assumption on the initialization of the network.

Assumption 15. The initialization of $\boldsymbol{W}$ at $t = 0$ satisfies $\nabla \mathcal{R}(\boldsymbol{W}(0)) \neq \mathcal{R}(0) = \ell(0)$.

Alignment yheorems We now assume the data is MPNN-separable, i.e., there is a set of weights that correctly classifies every data point. More specifically, assume there is a vector $\bar{\boldsymbol{u}} \in \mathbb{R}^d$ such that $y_i \cdot \bar{\boldsymbol{u}}^\intercal \boldsymbol{X}_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i} > 0$ for all i. Furthermore, the maximum margin is given by

$$\gamma = \max_{\|\bar{\boldsymbol{u}}\|=1} \min_{1 \leq i \leq k} y_i \cdot \bar{\boldsymbol{u}}^\intercal \boldsymbol{X}_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i} > 0.$$

while the corresponding solution $\bar{\boldsymbol{u}} \in \mathbb{R}^d$ is given by

$$\arg\max_{\|\bar{\boldsymbol{u}}\|=1} \min_{1 \leq i \leq k} y_i \cdot \bar{\boldsymbol{u}}^\intercal \boldsymbol{X}_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i}.$$

Furthermore, those $\boldsymbol{v}_i = \boldsymbol{Z}_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i}$ for which $\langle \bar{\boldsymbol{u}}, \boldsymbol{v}_i \rangle = \gamma$ are called *support vectors*.

Our first main result shows that under gradient flow, the trainable weight vectors of our MPNN architecture get "aligned."

Theorem 16. Suppose Assumption 14 and Assumption 15 hold. Let $\boldsymbol{u}_i(t) \in \mathbb{R}^{d_i}$ and $\boldsymbol{v}_i(t) \in \mathbb{R}^{d_{i-1}}$ denote the left and right singular vectors, respectively, of $\boldsymbol{W}^{(i)}(t) \in \mathbb{R}^{d_i \times d_{i-1}}$. Then, we have the following using the Frobenius norm $\|\cdot\|_F$:

- For $j = 1, 2, \ldots, L$, we have

$$\lim_{t \to \infty} \left\| \frac{\boldsymbol{W}^{(j)}(t)}{\|\boldsymbol{W}^{(j)}(t)\|_F} - \boldsymbol{u}_j(t) \boldsymbol{v}_j(t)^\intercal \right\|_F = 0.$$

- Also,

$$\lim_{t \to \infty} \left| \left\langle \frac{(\boldsymbol{W}^{(L)}(t) \cdots \boldsymbol{W}^{(1)}(t))^\intercal}{\prod_{j=1}^{L} \|\boldsymbol{W}^{(j)}(t)\|_F}, \boldsymbol{v}_1 \right\rangle \right| = 1.$$

Furthermore, we show that under mild assumptions, the weights converge to the maximum margin solution $\bar{\boldsymbol{u}}$.

Assumption 17. The support vectors $\boldsymbol{v}_i = \boldsymbol{Z}_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i}$ span $\mathbb{R}^d$.

Note that, for unlabeled graphs, due to separability, the above assumption is trivially fulfilled.

Theorem 18 (Convergence to the maximum margin solution)**.** Suppose Assumption 14 and Assumption 17 hold. Then, for the exponential loss function $\ell(x) = e^{-x}$, under gradient flow, we have that the learned weights of the MPNN converge to the maximum margin solution, i.e.,

$$\lim_{t \to \infty} \frac{\boldsymbol{W}^{(L)}(t) \boldsymbol{W}^{(L-1)}(t) \cdots \boldsymbol{W}^{(1)}(t)}{\|\boldsymbol{W}^{(L)}(t)\|_F \|\boldsymbol{W}^{(L-1)}(t)\|_F \cdots \|\boldsymbol{W}^{(1)}(t)\|_F} = \bar{\boldsymbol{u}}.$$

We note here that the results can be straightforwardly adjusted to MPNN$_{\mathcal{F}}$ architectures.

5. Experimental evaluation

In the following, we investigate to what extent our theoretical results translate into practice. Specifically, we answer the following questions.

Q1 Does adding expressive power make datasets more linearly separable?

Q2 Can the increased generalization performance of a more expressive variant of the 1-WL algorithm be explained by an increased margin?

Q3 Does the 1-WLOA$_{\mathcal{F}}$ lead to increased predictive performance?

Q4 Do the results lift to MPNNs?

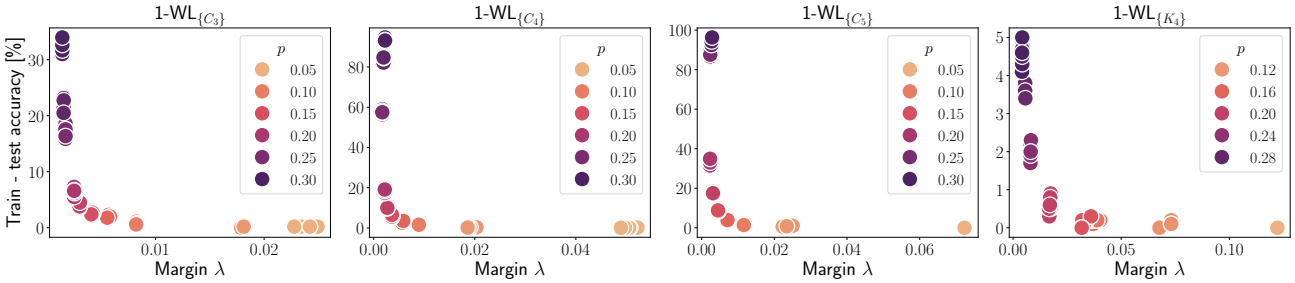
See Appendix G for further details, the experimental protocol, and model configurations. The source code of all methods and evaluation procedures is available at `https://www.github.com/chrsmrrs/wl_vc_expressivity`.

Results and discussion In the following, we answer questions **Q1** to **Q4**.

Q1 (*"Does adding expressive power make datasets more linearly separable?"*) See Tables 1, 3 and 6 (in the appendix). Table 1 confirms Proposition 9, i.e., the 1-WL and 1-WLOA kernels do not achieve accuracies better than random and cannot linearly separate the training data. The 1-WL$_{\mathcal{F}}$ and 1-WLOA$_{\mathcal{F}}$ kernel, linearly separate the data while achieving perfect test accuracies. In addition, Table 3 also confirms this for the ER graphs, i.e., for all datasets, the 1-WL and 1-WLOA kernels cannot separate the training data while the 1-WL$_{\mathcal{F}}$ and 1-WLOA$_{\mathcal{F}}$ can. Moreover, the subgraph-based kernels achieve the overall best predictive performance over all datasets, e.g., on the dataset using edge probability 0.2 and $\mathcal{F} = \{C_5\}$ the test accuracies of the 1-WL$_{\mathcal{F}}$ improves

Table 1: Experimental validation of Proposition 9 for different numbers of vertices (n), reporting mean test accuraccies and margins. NLS—Not linearly separable. DNC—Did not compute due to implicit kernel.

Algorithm	**Number of vertices** (n).			
	16	32	64	128
1-WL	46.6 ± 1.1 NLS	47.3 ± 1.7 NLS	47.1 ± 1.1 NLS	46.5 ± 0.7 NLS
1-WLOA	36.8 ± 1.3 DNC	37.4 ± 1.7 DNC	37.3 ± 0.9 DNC	37.8 ± 1.2 DNC
MPNN	47.9 ± 0.7 DNC	49.0 ± 1.6 DNC	48.3 ± 1.3 DNC	47.9 ± 1.6 DNC
1-WL$_{\mathcal{F}}$	**100.0** ± 0.0 **0.006** < 0.0001	**100.0** ± 0.0 **0.014** < 0.0001	**100.0** ± 0.0 **0.030** < 0.0001	**100.0** ± 0.0 **0.062** < 0.0001
1-WLOA$_{\mathcal{F}}$	100.0 ± 0.0 DNC	100.0 ± 0.0 DNC	100.0 ± 0.0 DNC	100.0 ± 0.0 DNC
MPNN$_{\mathcal{F}}$	100.0 ± 0.1 DNC	100.0 ± 0.1 DNC	100.0 < 0.1 DNC	100.0 ± 0.1 DNC



(a) C_3 (b) C_4 (c) C_5 (d) K_4

Figure 1: Plot illustrating the relation between the margin and generalization error for the ER graphs for different choices of $\mathcal{F}$.

over the 1-WL by more than 57 %. Similar effects can be observed for the MPNN architectures; see Table 6.

Q2 (*"Can the increased generalization performance of a more expressive variant of the* 1-*WL algorithm be explained by an increased margin?"*) See Tables 3 and 5 and Figure 1 (in the appendix). On the TUDATASETS, an increased margin often leads to less difference between train and test accuracy; see Table 5. For example, on the PROTEINS dataset, the 1-WL$_{\mathcal{F}}$, leads to a larger difference for all $\mathcal{F}$, while its margin is always *strictly smaller* than 1-WL's margin. Hence, the empirical results align with our theory, i.e., a smaller margin worsens the generalization error. Similar effects can be observed for all other datasets, except MUTAG. On the ER dataset, comparing the 1-WL and 1-WL$_{\mathcal{F}}$, for all $\mathcal{F}$, we can clearly confirm the theoretical results. That is, the 1-WL cannot separate any dataset with a positive margin, while the 1-WL$_{\mathcal{F}}$ can, and we observe a decreased difference between 1-WL$_{\mathcal{F}}$'s train and test accuracies compared to the 1-WL. Analyzing the 1-WL$_{\mathcal{F}}$ further, for all $\mathcal{F}$, a decreasing margin always results in an increased difference between test and train accuracies. For example, for $\mathcal{F} = \{C_4\}$ and $p = 0.05$, the 1-WL$_{\mathcal{F}}$ achieves a margin of 0.037 with a difference of 0.1 %, for $p = 0.1$, it achieves a margin of 0.009 with a difference of 1.8 %, for $p = 0.2$, it achieves a margin of 0.002 with a difference of 31.2 %, and, for $p = 0.3$, it achieve a margin of 0.003 with difference of 95.8 %. Moreover, see Figure 1 of visual illustration of this observation.

See Appendix G for **Q3** and **Q4**.

6. Conclusion

Here, we focused on determining the precise conditions under which increasing the expressive power of MPNN or kernel architectures leads to a provably increased generalization performance. When viewed through graph isomorphism, we first showed that an architecture's expressivity offers limited insights into its generalization performance. Additionally, we focused on augmenting 1-WL with subgraph information and derived tight upper and lower bounds for the architectures' VC dimension parameterized by the margin. Based on this, we derived data distributions where increased expressivity either leads to improved generalization performance or not. Finally, we introduced variations of expressive 1-WL-based kernels and neural architectures with provable generalization properties. Our empirical study confirmed the validity of our theoretical findings. Our theoretical results constitute an essential initial step in unraveling the conditions under which more expressive MPNN and kernel architectures yield enhanced generalization performance. Hence, our theory lays a solid foundation for the systematic and principled design of novel expressive MPNN architectures.

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

Christopher Morris is partially funded by a DFG Emmy Noether grant (468502433) and RWTH Junior Principal Investigator Fellowship under Germany's Excellence Strategy. Floris Geerts is supported by an FWO grant (G019222N).

References

Aamand, A., Chen, J. Y., Indyk, P., Narayanan, S., Rubinfeld, R., Schiefer, N., Silwal, S., and Wagner, T. Exponentially improving the complexity of simulating the Weisfeiler-Lehman test with graph neural networks. *ArXiv preprint*, 2022.

Abboud, R., Ceylan, İ. İ., Grohe, M., and Lukasiewicz, T. The surprising power of graph neural networks with random node initialization. In *Joint Conference on Artificial Intelligence*, pp. 2112–2118, 2021.

Alon, N., Hanneke, S., Holzman, R., and Moran, S. A theory of PAC learnability of partial concept classes. In *Annual Symposium on Foundations of Computer Science*, pp. 658–671, 2021.

Amir, T., Gortler, S. J., Avni, I., Ravina, R., and Dym, N. Neural injective functions for multisets, measures and graphs via a finite witness theorem. *ArXiv preprint*, 2023.

Anthony, M. and Bartlett, P. L. *Neural Network Learning - Theoretical Foundations*. Cambridge University Press, 2002.

Arora, S., Cohen, N., and Hazan, E. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, pp. 244–253, 2018.

Arvind, V., Köbler, J., Rattan, G., and Verbitsky, O. On the power of color refinement. In *International Symposium on Fundamentals of Computation Theory*, pp. 339–350, 2015.

Azizian, W. and Lelarge, M. Characterizing the expressive power of invariant and equivariant graph neural networks. In *International Conference on Learning Representations*, 2021.

Balcilar, M., Héroux, P., Gaüzère, B., Vasseur, P., Adam, S., and Honeine, P. Breaking the limits of message passing graph neural networks. In *International Conference on Machine Learning*, pp. 599–608, 2021.

Baranwal, A., Fountoulakis, K., and Jagannath, A. Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization. In *International Conference on Machine Learning*, 2021.

Barceló, P., Kostylev, E. V., Monet, M., Pérez, J., Reutter, J. L., and Silva, J. P. The logical expressiveness of graph neural networks. In *International Conference on Learning Representations*, 2020.

Barceló, P., Geerts, F., Reutter, J. L., and Ryschkov, M. Graph neural networks with local graph parameters. In *Advances in Neural Information Processing Systems*, pp. 25280–25293, 2021.

Barceló, P., Galkin, M., Morris, C., and Orth, M. A. R. Weisfeiler and Leman go relational. In *Learning of Graphs Conference*, 2022.

Bartlett, P. and Shawe-Taylor, J. Generalization performance of support vector machines and other pattern classifiers. *Advances in Kernel methods—support vector learning*, pp. 43–54, 1999.

Baskin, I. I., Palyulin, V. A., and Zefirov, N. S. A neural device for searching direct correlations between structures and properties of chemical compounds. *Journal of Chemical Information and Computer Sciences*, 37(4):715–721, 1997.

Beaini, D., Passaro, S., Létourneau, V., Hamilton, W. L., Corso, G., and Lió, P. Directional graph networks. In *International Conference on Machine Learning*, pp. 748–758, 2021.

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.

Bodnar, C., Frasca, F., Otter, N., Wang, Y. G., Liò, P., Montúfar, G., and Bronstein, M. M. Weisfeiler and Lehman go cellular: CW networks. In *Advances in Neural Information Processing Systems*, pp. 2625–2640, 2021a.

Bodnar, C., Frasca, F., Wang, Y., Otter, N., Montúfar, G. F., Lió, P., and Bronstein, M. M. Weisfeiler and Lehman go topological: Message passing simplicial networks. In *International Conference on Machine Learning*, pp. 1026–1037, 2021b.

Borgwardt, K. M. and Kriegel, H.-P. Shortest-path kernels on graphs. In *IEEE International Conference on Data Mining*, pp. 74–81, 2005.

Borgwardt, K. M., Ghisu, M. E., Llinares-López, F., O'Bray, L., and Rieck, B. Graph kernels: State-of-the-art and future challenges. *Foundations and Trends in Machine Learning*, 13(5–6), 2020.

Bouritsas, G., Frasca, F., Zafeiriou, S., and Bronstein, M. M. Improving graph neural network expressivity via subgraph isomorphism counting. *ArXiv preprint*, 2020.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. Spectral networks and deep locally connected networks on graphs. In *International Conference on Learning Representation*, 2014.

Böker, J., Levie, R., Huang, N., Villar, S., and Morris, C. Fine-grained expressivity of graph neural networks. In *Advances in Neural Information Processing Systems*, 2023.

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

Cappart, Q., Chételat, D., Khalil, E. B., Lodi, A., Morris, C., and Veličković, P. Combinatorial optimization and reasoning with graph neural networks. In *Joint Conference on Artificial Intelligence*, pp. 4348–4355, 2021.

Chang, C.-C. and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, pp. 27:1–27:27, 2011. ACM.

Chen, Z., Villar, S., Chen, L., and Bruna, J. On the equivalence between graph isomorphism testing and function approximation with gnns. In *Advances in Neural Information Processing Systems*, pp. 15868–15876, 2019.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

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

Dasoulas, G., Santos, L. D., Scaman, K., and Virmaux, A. Coloring graph neural networks for node disambiguation. In *International Joint Conference on Artificial Intelligence*, pp. 2126–2132, 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*, (2): 786–797, 1991.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pp. 3837–3845, 2016.

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

Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems*, pp. 2224–2232, 2015.

Easley, D. and Kleinberg, J. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

El-Yaniv, R. and Pechyony, D. Transductive rademacher complexity and its applications. In *Annual Conference on Learning Theory*, pp. 157–171, 2007.

Esser, P. M., Vankadara, L. C., and Ghoshdastidar, D. Learning theory can (sometimes) explain generalisation in graph neural networks. In *Advances in Neural Information Processing Systems*, pp. 27043–27056, 2021.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, pp. 1871–1874, 2008.

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*, 2022.

Finkelshtein, B., Huang, X., Bronstein, M., and Ceylan, İ. İ. Cooperative graph neural networks. *ArXiv preprint*, 2023.

Franks, B. J., Anders, M., Kloft, M., and Schweitzer, P. A systematic approach to universal random features in graph neural networks. *Transactions on Machine Learning Research*, 2023.

Frasca, F., Bevilacqua, B., Bronstein, M. M., and Maron, H. Understanding and extending subgraph GNNs by rethinking their symmetries. *ArXiv preprint*, 2022.

Gama, F., Marques, A. G., Leus, G., and Ribeiro, A. Convolutional neural network architectures for signals supported on graphs. *IEEE Transactions on Signal Processing*, 67 (4):1034–1049, 2019.

Garg, V. K., Jegelka, S., and Jaakkola, T. S. Generalization and representational limits of graph neural networks. In *International Conference on Machine Learning*, pp. 3419–3430, 2020.

Geerts, F. The expressive power of kth-order invariant graph networks. *ArXiv preprint*, 2020.

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

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

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pp. 1263–1272, 2017.

Goldreich, O. Introduction to testing graph properties. In *Property Testing*. Springer, 2010.

Goller, C. and Küchler, A. Learning task-dependent distributed representations by backpropagation through structure. In *International Conference on Neural Networks*, pp. 347–352, 1996.

Grohe, M. The descriptive complexity of graph neural networks. *ArXiv preprint*, 2023.

Grønlund, A., Kamma, L., and Larsen, K. G. Near-tight margin-based generalization bounds for support vector machines. In *International Conference on Machine Learning*, pp. 3779–3788, 2020.

Hamilton, W. L., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017.

Hammer, B. Generalization ability of folding networks. *IEEE Trans. Knowl. Data Eng.*, (2):196–206, 2001.

Helma, C., King, R. D., Kramer, S., and Srinivasan, A. The Predictive Toxicology Challenge 2000–2001 . *Bioinformatics*, 17(1):107–108, 01 2001.

Horn, M., Brouwer, E. D., Moor, M., Moreau, Y., Rieck, B., and Borgwardt, K. M. Topological graph neural networks. In *International Conference on Learning Representations*, 2022.

Huang, Y., Peng, X., Ma, J., and Zhang, M. Boosting the cycle counting power of graph neural networks with I^2-GNNs. *ArXiv preprint*, 2022.

Ji, Z. and Telgarsky, M. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2019.

Ju, H., Li, D., Sharma, A., and Zhang, H. R. Generalization in graph neural networks: Improved pac-bayesian bounds on graph diffusion. *ArXiv preprint*, 2023.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. Highly accurate protein structure prediction with AlphaFold. *Nature*, 2021.

Karpinski, M. and Macintyre, A. Polynomial bounds for VC dimension of sigmoidal and general Pfaffian neural networks. *Journal of Computer and System Sciences*, 54 (1):169–176, 1997.

Kim, J., Nguyen, T. D., Min, S., Cho, S., Lee, M., Lee, H., and Hong, S. Pure transformers are powerful graph learners. *ArXiv preprint*, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Kireev, D. B. Chemnet: A novel neural network based method for graph/property mapping. *Journal of Chemical Information and Computer Sciences*, 35(2):175–180, 1995.

Kriege, N. M. and Mutzel, P. Subgraph matching kernels for attributed graphs. In *International Conference on Machine Learning*, 2012.

Kriege, N. M., Giscard, P., and Wilson, R. C. On valid optimal assignment kernels and applications to graph classification. In *Advances in Neural Information Processing Systems*, pp. 1615–1623, 2016.

Kriege, N. M., Morris, C., Rey, A., and Sohler, C. A property testing framework for the theoretical expressivity of graph kernels. In *International Joint Conference on Artificial Intelligence*, pp. 2348–2354, 2018.

Kriege, N. M., Johansson, F. D., and Morris, C. A survey on graph kernels. *Applied Network Science*, 5(1):6, 2020.

Levie, R. A graphon-signal analysis of graph neural networks. In *Advances in Neural Information Processing Systems*, 2023.

Levie, R., Monti, F., Bresson, X., and Bronstein, M. M. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2019.

Li, P., Wang, Y., Wang, H., and Leskovec, J. Distance encoding: Design provably more powerful neural networks for graph representation learning. In *Advances in Neural Information Processing Systems*, 2020.

Liao, R., Urtasun, R., and Zemel, R. S. A PAC-Bayesian approach to generalization bounds for graph neural networks. In *International Conference on Learning Representations*, 2021.

Maehara, T. and NT, H. A simple proof of the universality of invariant/equivariant graph neural networks. *ArXiv preprint*, 2019.

Maron, H., Ben-Hamu, H., Serviansky, H., and Lipman, Y. Provably powerful graph networks. In *Advances in Neural Information Processing Systems*, pp. 2153–2164, 2019.

Martinkus, K., Papp, P. A., Schesch, B., and Wattenhofer, R. Agent-based graph neural networks. *ArXiv preprint*, 2022.

Maskey, S., Lee, Y., Levie, R., and Kutyniok, G. Generalization analysis of message passing neural networks on large random graphs. In *Advances in Neural Information Processing Systems*, 2022.

Merkwirth, C. and Lengauer, T. Automatic generation of complementary descriptors with molecular graph networks. *Journal of Chemical Information and Modeling*, 45(5):1159–1168, 2005.

Micheli, A. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009.

Micheli, A. and Sestito, A. S. A new neural network model for contextual processing of graphs. In *Italian Workshop on Neural Nets Neural Nets and International Workshop on Natural and Artificial Immune Systems*, pp. 10–17, 2005.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of Machine Learning*. MIT Press, 2012.

Monti, F., Boscaini, D., Masci, J., Rodolà, E., Svoboda, J., and Bronstein, M. M. Geometric deep learning on graphs and manifolds using mixture model cnns. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5425–5434, 2017.

Morris, C., Kersting, K., and Mutzel, P. Glocalized Weisfeiler-Lehman kernels: Global-local feature maps of graphs. In *IEEE International Conference on Data Mining*, pp. 327–336, 2017.

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 *AAAI Conference on Artificial Intelligence*, pp. 4602–4609, 2019.

Morris, C., Kriege, N. M., Bause, F., Kersting, K., Mutzel, P., and Neumann, M. TUDataset: A collection of benchmark datasets for learning with graphs. *ArXiv preprint*, 2020a.

Morris, C., Rattan, G., and Mutzel, P. Weisfeiler and Leman go sparse: Towards higher-order graph embeddings. In *Advances in Neural Information Processing Systems*, 2020b.

Morris, C., L., 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*, 2021.

Morris, C., Rattan, G., Kiefer, S., and Ravanbakhsh, S. SpeqNets: Sparsity-aware permutation-equivariant graph networks. In *International Conference on Machine Learning*, pp. 16017–16042, 2022.

Morris, C., Geerts, F., Tönshoff, J., and Grohe, M. WL meet VC. In *International Conference on Machine Learning*, pp. 25275–25302, 2023.

Müller, L., Galkin, M., Morris, C., and Rampásek, L. Attending to graph transformers. *ArXiv preprint*, 2023.

Murphy, R. L., Srinivasan, B., Rao, V. A., and Ribeiro, B. Relational pooling for graph representations. In *International Conference on Machine Learning*, pp. 4663–4673, 2019.

Nguyen, H. and Maehara, T. Graph homomorphism convolution. In *International Conference on Machine Learning*, pp. 7306–7316, 2020.

Papp, P. A. and Wattenhofer, R. A theoretical comparison of graph neural network extensions. In *International Conference on Machine Learning*, pp. 17323–17345, 2022.

Papp, P. A., K. Martinkus, L. F., and Wattenhofer, R. DropGNN: Random dropouts increase the expressiveness of graph neural networks. In *Advances in Neural Information Processing Systems*, 2021.

Puny, O., Lim, D., Kiani, B. T., Maron, H., and Lipman, Y. Equivariant polynomials for graph neural networks. *ArXiv preprint*, 2023.

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

Qian, C., Manolache, A., Ahmed, K., Zeng, Z., den Broeck, G. V., Niepert, M., and Morris, C. Probabilistically rewired message-passing neural networks. *ArXiv preprint*, 2023.

Rosenbluth, E., Tönshoff, J., and Grohe, M. Some might say all you need is sum. *ArXiv preprint*, 2023.

Sato, R., Yamada, M., and Kashima, H. Random features strengthen graph neural networks. In *SIAM International Conference on Data Mining*, pp. 333–341, 2021.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

Scarselli, F., Tsoi, A. C., and Hagenbuchner, M. The Vapnik-Chervonenkis dimension of graph and recursive neural networks. *Neural Networks*, pp. 248–259, 2018.

Schomburg, I., Chang, A., Ebeling, C., Gremse, M., Heldt, C., Huhn, G., and Schomburg, D. Brenda, the enzyme database: updates and major new developments. *Nucleic acids research*, pp. D431–3, 2004.

Shervashidze, N., Vishwanathan, S. V. N., Petri, T. H., Mehlhorn, K., and Borgwardt, K. M. Efficient graphlet kernels for large graph comparison. In *International Conference on Artificial Intelligence and Statistics*, pp. 488–495, 2009.

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

Simonovsky, M. and Komodakis, N. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 29–38, 2017.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19: 70:1–70:57, 2018.

Sperduti, A. and Starita, A. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–35, 1997.

Stokes, J., Yang, K., Swanson, K., Jin, W., Cubillos-Ruiz, A., Donghia, N., MacNair, C., French, S., Carfrae, L., Bloom-Ackerman, Z., Tran, V., Chiappino-Pepe, A., Badran, A., Andrews, I., Chory, E., Church, G., Brown, E., Jaakkola, T., Barzilay, R., and Collins, J. A deep learning approach to antibiotic discovery. *Cell*, pp. 688–702.e13, 2020.

Talak, R., Hu, S., Peng, L., and Carlone, L. Neural trees for learning on graphs. *ArXiv preprint*, 2021.

Thiede, E. H., Zhou, W., and Kondor, R. Autobahn: Automorphism-based graph neural nets. In *Advances in Neural Information Processing Systems*, pp. 29922–29934, 2021.

Tolstikhin, I. O. and Lopez-Paz, D. Minimax lower bounds for realizable transductive classification. *ArXiv preprint*, 2016.

Tönshoff, J., Ritzert, M., Wolf, H., and Grohe, M. Graph learning with 1D convolutions on random walks. *ArXiv preprint*, 2021.

Vapnik, V. *Statistical learning theory*. Wiley, 1998.

Vapnik, V. N. *The Nature of Statistical Learning Theory*. Springer, 1995.

Vapnik, V. N. and Chervonenkis, A. A note on one class of perceptrons. *Avtomatika i Telemekhanika*, 24(6):937–945, 1964.

Velingker, A., Sinop, A. K., Ktena, I., Velickovic, P., and Gollapudi, S. Affinity-aware graph networks. *ArXiv preprint*, 2022.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Verma, S. and Zhang, Z. Stability and generalization of graph convolutional neural networks. In *International Conference on Knowledge Discovery & Data Mining*, pp. 1539–1548, 2019.

Vignac, C., Loukas, A., and Frossard, P. Building powerful and equivariant graph neural networks with structural message-passing. In *Advances in Neural Information Processing Systems*, 2020.

Weisfeiler, B. and Leman, A. The reduction of a graph to canonical form and the algebra which appears therein. *Nauchno-Technicheskaya Informatsia*, 2 (9):12–16, 1968. English translation by G. Ryabov is available at `https://www.iti.zcu.cz/wl2018/pdf/wl_paper_translation.pdf`.

Wijesinghe, A. and Wang, Q. A new perspective on "how graph neural networks go beyond weisfeiler-lehman?". In *International Conference on Learning Representations*, 2022.

Wong, F., Zheng, E. J., Valeri, J. A., Donghia, N. M., Anahtar, M. N., Omori, S., Li, A., Cubillos-Ruiz, A., Krishnan, A., Jin, W., Manson, A. L., Friedrichs, J., Helbig, R., Hajian, B., Fiejtek, D. K., Wagner, F. F., Soutter, H. H., Earl, A. M., Stokes, J. M., Renner, L. D., and Collins, J. J. Discovery of a structural class of antibiotics with explainable deep learning. *Nature*, 2023.

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

Yanardag, P. and Vishwanathan, S. V. N. A structural smoothing framework for robust graph comparison. In *Advances in Neural Information Processing Systems*, pp. 2134–2142, 2015a.

Yanardag, P. and Vishwanathan, S. V. N. Deep graph kernels. In *International Conference on Knowledge Discovery and Data Mining*, pp. 1365–1374, 2015b.

Yehudai, G., Fetaya, E., Meirom, E. A., Chechik, G., and Maron, H. From local structures to size generalization in graph neural networks. In *International Conference on Machine Learning*, pp. 11975–11986, 2021.

You, J., Gomes-Selman, J., Ying, R., and Leskovec, J. Identity-aware graph neural networks. In *AAAI Conference on Artificial Intelligence*, 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*, 2023a.

Zhang, B., Luo, S., Wang, L., and He, D. Rethinking the expressive power of GNNs via graph biconnectivity. *ArXiv preprint*, 2023b.

Zhang, M. and Li, P. Nested graph neural networks. In *Advances in Neural Information Processing Systems*, pp. 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. Extended related work

Here, we give a more detailled account of related work.

A.1. Graph kernels based on the 1-WL

Shervashidze et al. (2011) were the first to utilize the 1-WL as a graph kernel. Later, Morris et al. (2017; 2020b; 2022) generalized this to variants of the k-WL. Moreover, Kriege et al. (2016) derived the *Weisfeiler-Leman optimal assignment kernel*, using the 1-WL to compute optimal assignments between vertices of two given graphs. Yanardag & Vishwanathan (2015a) successfully employed the Weisfeiler–Leman kernels within frameworks for smoothed (Yanardag & Vishwanathan, 2015a) and deep graph kernels (Yanardag & Vishwanathan, 2015b). For a theoretical investigation of graph kernels based on the 1-WL, see (Kriege et al., 2018). See also (Morris et al., 2021) for an overview of the Weisfeiler–Leman algorithm in machine learning and Borgwardt et al. (2020); Kriege et al. (2020) for a detailed review of graph kernels.

A.2. MPNNs

Recently, MPNNs (Gilmer et al., 2017; Scarselli et al., 2009) emerged as the most prominent graph representation learning architecture. Notable instances of this architecture include, e.g., Duvenaud et al. (2015); Hamilton et al. (2017), and Veličković et al. (2018), which can be subsumed under the message-passing framework introduced in Gilmer et al. (2017). In parallel, approaches based on spectral information were introduced in, e.g., Bruna et al. (2014); Defferrard et al. (2016); Gama et al. (2019); Kipf & Welling (2017); Levie et al. (2019), and Monti et al. (2017)—all of which descend from early work in Baskin et al. (1997); Goller & Küchler (1996); Kireev (1995); Merkwirth & Lengauer (2005); Micheli & Sestito (2005); Micheli (2009); Scarselli et al. (2009), and Sperduti & Starita (1997). Rcently, connections between MPNNs and Weisfeiler–Leman-type algorithms have been shown (Barceló et al., 2020; Geerts et al., 2021; Morris et al., 2019; Xu et al., 2019). Specifically, Morris et al. (2019) and Xu et al. (2019) showed that the 1-WL limits the expressive power of any possible MPNN architecture in distinguishing non-isomorphic graphs. (Bouritsas et al., 2020) showed how to make MPNNs more expressive by incorporating subgraph information; see Appendix A for an extended discussion on more expressive MPNNs.

A.3. Expressive power of MPNNs

Recently, connections between MPNNs and Weisfeiler–Leman-type algorithms have been shown (Barceló et al., 2020; Geerts et al., 2021; Morris et al., 2019; Xu et al., 2019). Specifically, Morris et al. (2019) and Xu et al. (2019) showed that the 1-WL limits the expressive power of any possible MPNN architecture in distinguishing non-isomorphic graphs. In turn, these results have been generalized to the k-WL, e.g., Azizian & Lelarge (2021); Geerts (2020); Maron et al. (2019); Morris et al. (2019; 2020b; 2022), and connected to the permutation-equivariant function approximation over graphs, see, e.g., Chen et al. (2019); Geerts & Reutter (2022); Maehara & NT (2019); Azizian & Lelarge (2021). Furthermore, Aamand et al. (2022); Amir et al. (2023) devised an improved analysis using randomization and moments of neural networks, respectively. Recent works have extended the expressive power of MPNNs, e.g., by encoding vertex identifiers (Murphy et al., 2019; Vignac et al., 2020), using random features (Abboud et al., 2021; Dasoulas et al., 2020; Sato et al., 2021) or individualization-refinement algorithms (Franks et al., 2023), affinity measures (Velingker et al., 2022), equivariant graph polynomials (Puny et al., 2023), homomorphism and subgraph counts (Barceló et al., 2021; Bouritsas et al., 2020; Nguyen & Maehara, 2020), spectral information (Balcilar et al., 2021), simplicial (Bodnar et al., 2021b) and cellular complexes (Bodnar et al., 2021a), persistent homology (Horn et al., 2022), random walks (Tönshoff et al., 2021; Martinkus et al., 2022), graph decompositions (Talak et al., 2021), relational (Barceló et al., 2022), distance (Li et al., 2020) and directional information (Beaini et al., 2021), graph rewiring (Qian et al., 2023) and adaptive message passing (Finkelshtein et al., 2023), subgraph information (Bevilacqua et al., 2022; Cotta et al., 2021; Feng et al., 2022; Frasca et al., 2022; Huang et al., 2022; Morris et al., 2021; Papp et al., 2021; Papp & Wattenhofer, 2022; Qian et al., 2022; Thiede et al., 2021; Wijesinghe & Wang, 2022; You et al., 2021; Zhang & Li, 2021; Zhao et al., 2022; Zhang et al., 2023a), and biconnectivity (Zhang et al., 2023b). See Morris et al. (2021) for an in-depth survey on this topic. Geerts & Reutter (2022) devised a general approach to bound the expressive power of a large variety of MPNNs using 1-WL or k-WL.

Recently, Kim et al. (2022) showed that transformer architectures (Müller et al., 2023) can simulate the 2-WL. Grohe (2023) showed tight connections between MPNNs' expressivity and circuit complexity. Moreover, Rosenbluth et al. (2023) investigated the expressive power of different aggregation functions beyond sum aggregation. Finally, Böker et al. (2023) defined a continuous variant of the 1-WL, deriving a more fine-grained topological characterization of the expressive power of MPNNs.

A.4. Generalization abilities of graph kernels and MPNNs

Scarselli et al. (2018) used classical techniques from learning theory (Karpinski & Macintyre, 1997) to show that MPNNs' VC dimension (Vapnik, 1995) with piece-wise polynomial activation functions on a *fixed* graph, under various assumptions, is in $\mathcal{O}(P^2 n \log n)$, where P is the number of parameters and n is the order of the input graph; see also Hammer (2001). We note here that Scarselli et al. (2018) analyzed a different type of MPNN not aligned with modern MPNN architectures (Gilmer et al., 2017). Garg et al. (2020) showed that the empirical Rademacher complexity (see, e.g., Mohri et al. (2012)) of a specific, simple MPNN architecture, using sum aggregation, is bounded in the maximum degree, the number of layers, Lipschitz constants of activation functions, and parameter matrices' norms. We note here that their analysis assumes weight sharing across layers. Liao et al. (2021) refined these results via a PAC-Bayesian approach, further refined in Ju et al. (2023). Maskey et al. (2022) used random graphs models to show that MPNNs' generalization ability depends on the (average) number of vertices in the resulting graphs. In addition, Levie (2023) defined a measure of a natural graph-signal similarity notion, resulting in a generalization bound for MPNNs depending on the covering number and the number of vertices. Verma & Zhang (2019) studied the generalization abilities of 1-layer MPNNs in a transductive setting based on algorithmic stability. Similarly, Esser et al. (2021) used stochastic block models to study the transductive Rademacher complexity (El-Yaniv & Pechyony, 2007; Tolstikhin & Lopez-Paz, 2016) of standard MPNNs. For semi-supervised node classification, (Baranwal et al., 2021) studied the classification of a mixture of Gaussians, where the data corresponds to the node features of a stochastic block model, under which conditions the mixture model is linearly separable using the GCN layer (Kipf & Welling, 2017). Most recently, (Morris et al., 2023) made progress connecting MPNNs' expressive power and generalization ability via the Weisfeiler–Leman hierarchy. They studied the influence of graph structure and the parameters' encoding lengths on MPNNs' generalization by tightly connecting 1-WL's expressivity and MPNNs' Vapnik–Chervonenkis (VC) dimension. They derived that MPNNs' VC dimension depends tightly on the number of equivalence classes computed by the 1-WL over a given set of graphs. Moreover, they showed that MPNNs' VC dimension depends logarithmically on the number of colors computed by the 1-WL and polynomially on the number of parameters. Kriege et al. (2018) leveraged results from graph property testing (Goldreich, 2010) to study the sample complexity of learning to distinguish various graph properties, e.g., planarity or triangle freeness, using graph kernels (Borgwardt et al., 2020; Kriege et al., 2020). Finally, (Yehudai et al., 2021) showed negative results for MPNNs' generalization ability to larger graphs.

B. Extended notation

Let $\mathbb{N} \coloneqq \{1, 2, 3, \dots\}$. For $n \geq 1$, let $[n] \coloneqq \{1, \dots, n\} \subset \mathbb{N}$. We use $\{\!\{\dots\}\!\}$ to denote multisets, i.e., the generalization of sets allowing for multiple instances for each of its elements. For two sets X and Y, let X^Y denote the set of functions mapping from Y to X. Let $S \subset \mathbb{R}^d$, then the *convex hull* $\mathsf{conv}(S)$ is the minimal convex set containing the set S. For $\boldsymbol{p} \in \mathbb{R}^d, d > 0$, and $\varepsilon > 0$, the *ball* $B(\boldsymbol{p}, \varepsilon, d) \coloneqq \{\boldsymbol{s} \in \mathbb{R}^d \mid \|\boldsymbol{p} - \boldsymbol{s}\| \leq \varepsilon\}$. Here, and in the remainder of the paper, $\|\cdot\|$ refers to the 2-*norm* $\|\boldsymbol{x}\| \coloneqq \sqrt{x_1^2 + \cdots + x_d^2}$ for $\boldsymbol{x} \in \mathbb{R}^d$.

Graphs An *(undirected) graph* G is a pair $(V(G), E(G))$ with *finite* sets of *vertices* or *nodes* $V(G)$ and *edges* $E(G) \subseteq \{\{u, v\} \subseteq V(G) \mid u \neq v\}$. For ease of notation, we denote an edge $\{u, v\}$ in $E(G)$ by (u, v) or (v, u). The *order* of a graph G is its number $|V(G)|$ of vertices. If not stated otherwise, we set $n \coloneqq |V(G)|$ and call G an n-order graph. We denote the set of all n-order graphs by $\mathcal{G}_n$. For a graph $G \in \mathcal{G}_n$, we denote its *adjacency matrix* by $\boldsymbol{A}(G) \in \{0, 1\}^{n \times n}$, where $A(G)_{vw} = 1$ if, and only, if $(v, w) \in E(G)$. We use standard notation

The *neighborhood* of $v \in V(G)$ is denoted by $N(v) \coloneqq \{u \in V(G) \mid (v, u) \in E(G)\}$ and the *degree* of a vertex v is $|N(v)|$. A *(vertex-)labeled graph* G is a triple $(V(G), E(G), \ell)$ with a (vertex-)label function $\ell \colon V(G) \to \mathbb{N}$. Then $\ell(v)$ is a *label* of v, for $v \in V(G)$. For $X \subseteq V(G)$, the graph $G[X] \coloneqq (X, E_X)$ is the *subgraph induced by* X, where $E_X \coloneqq \{(u, v) \in E(G) \mid u, v \in X\}$. Two graphs G and H are *isomorphic*, and we write $G \simeq H$ if there exists a bijection $\varphi \colon V(G) \to V(H)$ preserving the adjacency relation, i.e., (u, v) is in $E(G)$ if, and only, if $(\varphi(u), \varphi(v))$ is in $E(H)$. Then φ is an *isomorphism* between G and H. In the case of labeled graphs, we additionally require that $l(v) = l(\varphi(v))$ for all v in $V(G)$. We denote the *complete graph* on n vertices by K_n and a *cycle* on n vertices by C_n. for $r \geq 0$, a graph is r-*regular* if all of its vertices have degree r. Given two graphs G and H with disjoint vertex sets, we denote their disjoint union by $G \dot{\cup} H$.

C. Simple MPNNs

Here, we provide more details on the simple MPNNs mentioned in Section 2. That is, for given d and $L \in \mathbb{N}$, we define the class $\mathsf{MPNN}_{\mathsf{mlp}}(d, L)$ of simple MPNNs as L-layer MPNNs for which, according to Section 2, for each $t \in [L]$, the

aggregation function $\mathsf{AGG}^{(t)}$ is summation and the update function $\mathsf{UPD}^{(t)}$ is a multilayer perceptron $\mathsf{mlp}^{(t)} \colon \mathbb{R}^{2d} \to \mathbb{R}^d$ of width at most d. Similarly, the readout function in Equation (1) consists of a multilayer perceptron $\mathsf{mlp} : \mathbb{R}^d \to \mathbb{R}^d$ applied on the sum of all vertex features computed in layer L.[1] More specifically, MPNNs in $\mathsf{MPNN}_{\mathsf{mlp}}(d, L)$ compute on a labeled graph $G = (V(G), E(G), \ell)$ with d-dimensional initial vertex features $\boldsymbol{h}_v^{(0)} \in \mathbb{R}^d$, consistent with ℓ, the following vertex features, for each $v \in V(G)$,

$$\boldsymbol{h}_v^{(t)} \coloneqq \mathsf{mlp}^{(t)}\Big(\boldsymbol{h}_v^{(t-1)}, \sum_{u \in N(v)} \boldsymbol{h}_u^{(t-1)}\Big) \in \mathbb{R}^d,$$

for $t \in [L]$, and

$$\boldsymbol{h}_G \coloneqq \mathsf{mlp}\Big(\sum_{v \in V(G)} \boldsymbol{h}_v^{(L)}\Big) \in \mathbb{R}^d.$$

Note that the class $\mathsf{MPNN}_{\mathsf{mlp}}(d, L)$ encompasses the GNN architecture derived in Morris et al. (2019) that has the same expressive power as the 1-WL in distinguishing non-isomorphic graphs.

Notation In the subsequent sections, we use the following notation for MPNNs. We denote the class of all (labeled) graphs by $\mathcal{G}$. For $d, l > 0$, we denote the class of MPNNs using summation for aggregation, and such that update and readout functions are *multilayer perceptrons* (MLPs), all of a width of at most d, by $\mathsf{MPNN}_{\mathsf{mlp}}(d, L)$. We refer to elements in $\mathsf{MPNN}_{\mathsf{mlp}}(d, L)$ as *simple MPNNs*; see Appendix C for details. We stress that simple MPNNs are already expressive enough to be equivalent to the 1-WL in distinguishing non-isomorphic graphs (Morris et al., 2019). The class $\mathsf{MPNN}_{\mathsf{mlp},\mathcal{F}}(d, L)$ is defined similarly, based on $\mathrm{MPNN}_{\mathcal{F}}$s.

D. Graph kernels based on the 1-$\mathsf{WL}_{\mathcal{F}}$

Similar to the 1-WL, we can also define a graph kernel based on the 1-$\mathsf{WL}_{\mathcal{F}}$. Let G be a graph, we run the 1-$\mathsf{WL}_{\mathcal{F}}$ for $T \geq 0$ iterations, resulting in a coloring function $C_t^{1,\mathcal{F}} \to \Sigma_t$ for each iteration $t \leq T$. Let Σ_t denote the *range* of $C_t^{1,\mathcal{F}}$, i.e., $\Sigma_t \coloneqq \{c \mid \exists\, v \in V(G) \colon C_t^{1,\mathcal{F}}(v) = c\}$. Again, we assume Σ_t to be ordered by the natural order of $\mathbb{N}$, i.e., we assume that Σ_t consists of $c_1 < \cdots < c_{|\Sigma_t|}$. After each iteration, we compute a feature vector $\phi_{\mathcal{F},t}(G) \in \mathbb{R}^{|\Sigma_t|}$ for each graph G. Each component $\phi_{\mathcal{F},t}(G)_i$ counts the number of occurrences of vertices of G labeled by $c_i \in \Sigma_t$. The overall feature vector $\phi_{\mathsf{WL}_{\mathcal{F}}}(G)$ is defined as the concatenation of the feature vectors of all T iterations, i.e.,

$$\phi_{\mathsf{WL}_{\mathcal{F}}}^{(T)}(G) \coloneqq \big[\phi_{\mathcal{F},0}(G), \ldots, \phi_{\mathcal{F},T}(G)\big],$$

where $[\ldots]$ denote column-wise vector concatenation. We then define the kernel and its normalized counterpart in the same way as with the 1-WL.

E. Proofs missing from the main paper

Here, we outline proofs missing in the main paper.

E.1. Fundamentals

Here, we prove some fundamental statements for later use.

Margin optimization Let $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n) \in \mathbb{R}^d \times \{0, 1\}$, $d > 0$, be a linearly separable sample, and let $I^+ \coloneqq \{i \in [n] \mid y_i = 1\}$ and $I^- \coloneqq \{i \in [n] \mid y_i = 0\}$. Consider the well-known alternative—to the typical hard-margin SVM formulation—optimization problem for finding the minimum distance between the convex sets induced by the two classes,

[1] For simplicity, we assume that all feature dimensions of the layers are fixed to $d \in \mathbb{N}$.

i.e.,

$$
\begin{aligned}
2\lambda := \min_{\boldsymbol{\alpha}\in\mathbb{R}^{|I^+|},\boldsymbol{\beta}\in\mathbb{R}^{|I^-|}} \quad & \|\boldsymbol{x}^+_{\boldsymbol{\alpha}} - \boldsymbol{x}^-_{\boldsymbol{\beta}}\| \\
\text{s.t.} \quad & \boldsymbol{x}^+_{\boldsymbol{\alpha}} = \sum_{i\in I^+} \alpha_i \boldsymbol{x}_i, \quad \boldsymbol{x}^-_{\boldsymbol{\beta}} = \sum_{j\in I^-} \beta_j \boldsymbol{x}_j \\
& \sum_{i\in I^+} \alpha_i = 1, \quad \sum_{j\in I^-} \beta_j = 1, \\
& \forall\, i \in I^+, j \in I^- \colon \alpha_i \geq 0, \beta_j \geq 0,
\end{aligned} \tag{2}
$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the variables determining the convex combinations for both the positive and negative classes. Moreover, λ is exactly the margin that is computed by the typical hard-margin SVM and from the optimal arguments $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$, we can compute the usual hard-margin solution $\mathbf{w}$ and b as:

$$
\begin{aligned}
\mathbf{w} :=& \frac{\boldsymbol{x}^+_{\boldsymbol{\alpha}^*} - \boldsymbol{x}^-_{\boldsymbol{\beta}^*}}{\lambda^2} \\
b :=& \frac{\|\boldsymbol{x}^-_{\boldsymbol{\beta}^*}\|^2 - \|\boldsymbol{x}^+_{\boldsymbol{\alpha}^*}\|^2}{2\lambda^2}
\end{aligned}
$$

.

We can describe $\|\boldsymbol{x}^+_{\boldsymbol{\alpha}} - \boldsymbol{x}^-_{\boldsymbol{\beta}}\|^2$ by a sum of pairwise distances.

$$
\begin{aligned}
\left\|\boldsymbol{x}^+_{\boldsymbol{\alpha}} - \boldsymbol{x}^-_{\boldsymbol{\beta}}\right\|^2 &= \left\|\sum_{i\in I^+} \alpha_i \boldsymbol{x}_i - \sum_{j\in I^-} \beta_j \boldsymbol{x}_j\right\|^2 \\
&= \left\|\sum_{i\in I^+} \alpha_i \boldsymbol{x}_i \sum_{j\in I^-} \beta_j - \sum_{j\in I^-} \beta_j \boldsymbol{x}_j \sum_{i\in I^+} \alpha_i\right\|^2 \\
&= \left\|\sum_{i\in I^+}\sum_{j\in I^-} \alpha_i\beta_j \boldsymbol{x}_i - \sum_{i\in I^+}\sum_{j\in I^-} \alpha_i\beta_j \boldsymbol{x}_j\right\|^2 \\
&= \left\|\sum_{(i,j)\in I^+\times I^-} \delta_{i,j}(\boldsymbol{x}_i - \boldsymbol{x}_j)\right\|^2 \quad (\delta_{i,j} := \alpha_i\beta_j) \\
&= \sum_{(i,j)\in I^+\times I^-} \sum_{(k,l)\in I^+\times I^-} \delta_{i,j}\delta_{k,l}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\intercal(\boldsymbol{x}_k - \boldsymbol{x}_l) \\
&= \sum_{(i,j),(k,l)\in I^+\times I^-} \delta_{i,j}\delta_{k,l}(-\boldsymbol{x}_i^\intercal\boldsymbol{x}_l - \boldsymbol{x}_j^\intercal\boldsymbol{x}_k + \boldsymbol{x}_i^\intercal\boldsymbol{x}_k + \boldsymbol{x}_j^\intercal\boldsymbol{x}_l) \\
&= \frac{1}{2}\sum_{(i,j),(k,l)\in I^+\times I^-} \delta_{i,j}\delta_{k,l}(\|\boldsymbol{x}_i - \boldsymbol{x}_l\|^2 + \|\boldsymbol{x}_j - \boldsymbol{x}_k\|^2 - \|\boldsymbol{x}_i - \boldsymbol{x}_k\|^2 - \|\boldsymbol{x}_j - \boldsymbol{x}_l\|^2).
\end{aligned} \tag{3}
$$

We remark that the pairwise distances indexed by (i, l) and (j, k) represent inter-class distances, since $y_i = y_k = 1$ and $y_j = y_l = 0$. Along the same line, the pairwise distances indexed by (i, k) and (j, l) represent intra-class distances.

Proposition 19. Let $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ and $(\tilde{\boldsymbol{x}}_1, y_1), \ldots, (\tilde{\boldsymbol{x}}_n, y_n)$ in $\mathbb{R}^d$ be two linearly-separable samples, with margins λ and $\tilde{\lambda}$, respectively, with the same labels $y_i \in \{0, 1\}$. If

$$
\min_{y_i\neq y_j} \|\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j\|^2 - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 > \max_{y_i = y_j} \|\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j\|^2 - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2, \tag{4}
$$

then $\tilde{\lambda} > \lambda$. That is, we get an increase in margin if the minimum increase in distances between classes considering the two samples is strictly larger than the maximum increase in distance within each class.

Proof. Let

$$
\Delta_{\min} := \min_{y_i\neq y_j} \|\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j\|^2 - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2,
$$

and

$$\Delta_{\max} \coloneqq \max_{y_i = y_j} \|\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j\|^2 - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2.$$

By Equation (4), $\Delta_{\min} > \Delta_{\max}$. Starting at Equation (3),

$$\begin{aligned}
\|\boldsymbol{x}^+_{\boldsymbol{\alpha}} - \boldsymbol{x}^-_{\boldsymbol{\beta}}\|^2 &= \frac{1}{2} \sum_{(i,j)} \sum_{(k,l)} \alpha_{i,j} \alpha_{k,l} (\|\boldsymbol{x}_i - \boldsymbol{x}_l\|^2 + \|\boldsymbol{x}_j - \boldsymbol{x}_k\|^2 - \|\boldsymbol{x}_i - \boldsymbol{x}_k\|^2 - \|\boldsymbol{x}_j - \boldsymbol{x}_l\|^2) \\
&< \frac{1}{2} \sum_{(i,j)} \sum_{(k,l)} \alpha_{i,j} \alpha_{k,l} (\|\boldsymbol{x}_i - \boldsymbol{x}_l\|^2 + \Delta_{\min} + \|\boldsymbol{x}_j - \boldsymbol{x}_k\|^2 + \Delta_{\min} \\
&\qquad - \|\boldsymbol{x}_i - \boldsymbol{x}_k\|^2 - \Delta_{\max} - \|\boldsymbol{x}_j - \boldsymbol{x}_l\|^2 - \Delta_{\max} \\
&\leq \frac{1}{2} \sum_{(i,j)} \sum_{(k,l)} \alpha_{i,j} \alpha_{k,l} (\|\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_l\|^2 + \|\tilde{\boldsymbol{x}}_j - \tilde{\boldsymbol{x}}_k\|^2 - \|\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_k\|^2 - \|\tilde{\boldsymbol{x}}_j - \tilde{\boldsymbol{x}}_l\|^2) \\
&= \|\tilde{\boldsymbol{x}}^+_{\boldsymbol{\alpha}} - \tilde{\boldsymbol{x}}^-_{\boldsymbol{\alpha}}\|^2,
\end{aligned}$$

where $\tilde{\boldsymbol{x}}^+_{\boldsymbol{\alpha}}$ and $\tilde{\boldsymbol{x}}^-_{\boldsymbol{\alpha}}$ are derived from applying the optimization (2) to the datapoints $(\tilde{\boldsymbol{x}}_1, y_1), \ldots, (\tilde{\boldsymbol{x}}_n, y_n)$.

In the following, we omit all of the conditions from Equation (2) for simplicity. Let $\boldsymbol{x}^{+*}$ and $\boldsymbol{x}^{-*}$ be the representatives of the optimal solution to Equation (2), then

$$\forall \boldsymbol{\alpha} : \gamma = \|\boldsymbol{x}^{+*} - \boldsymbol{x}^{-*}\| \leq \|\boldsymbol{x}^+_{\boldsymbol{\alpha}} - \boldsymbol{x}^-_{\boldsymbol{\alpha}}\|.$$

Hence,

$$\forall \boldsymbol{\alpha} : \gamma = \|\boldsymbol{x}^{+*} - \boldsymbol{x}^{-*}\| \leq \|\boldsymbol{x}^+_{\boldsymbol{\alpha}} - \boldsymbol{x}^-_{\boldsymbol{\alpha}}\| < \|\tilde{\boldsymbol{x}}^+_{\boldsymbol{\alpha}} - \tilde{\boldsymbol{x}}^-_{\boldsymbol{\alpha}}\|,$$

which implies that

$$\gamma < \min_{\boldsymbol{\alpha}} \|\tilde{\boldsymbol{x}}^+_{\boldsymbol{\alpha}} - \tilde{\boldsymbol{x}}^-_{\boldsymbol{\alpha}}\| \eqqcolon \tilde{\gamma},$$

showing the desired result. □

Concatenating feature vectors We will consider concatenating two feature vectors and analyze how this affects attained margins. To this end, let $X \coloneqq \{(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \{0,1\} \mid i \in [n]\}$. When we split up $\mathbb{R}^d$ into $\mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$, we write $\boldsymbol{x}_i \coloneqq (\boldsymbol{x}^1_i, \boldsymbol{x}^2_i)$ with $\boldsymbol{x}^1_i \in \mathbb{R}^{d_1}$ and $\boldsymbol{x}^2_i \in \mathbb{R}^{d_2}$.

Proposition 20. If $X \coloneqq \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ is a sample, such that

1. $(\boldsymbol{x}^1_1, y_1), \ldots, (\boldsymbol{x}^1_n, y_n)$ is (r_1, γ_1)-separable and
2. $(\boldsymbol{x}^2_1, y_1), \ldots, (\boldsymbol{x}^2_n, y_n)$ is (r_2, γ_2)-separable,

then $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ is $(\sqrt{r_1^2 + r_2^2}, \sqrt{\gamma_1^2 + \gamma_2^2})$-separable.

Proof. Let $I \coloneqq I^+ \dot\cup I^-$ satisfying $y_i = 1$ if, and only, if $i \in I^+$ and $y_i = 0$ if, and only, if $i \in I^-$, $p \coloneqq |I|$, $p^+ \coloneqq |I^+|$ and $p^- \coloneqq |I^-|$. Further, let $\boldsymbol{x}^+_i \coloneqq \boldsymbol{x}_i$, $(\boldsymbol{x}^1_i)^+ \coloneqq (\boldsymbol{x}^1_i, 0)$, $(\boldsymbol{x}^2_i)^+ \coloneqq (0, \boldsymbol{x}^2_i)$ for $i \in I^+$, and $\boldsymbol{x}^-_i \coloneqq \boldsymbol{x}_i$, $(\boldsymbol{x}^1_i)^- \coloneqq (\boldsymbol{x}^1_i, 0)$, and $(\boldsymbol{x}^2_i)^- \coloneqq (0, \boldsymbol{x}^2_i)$ for $i \in I^-$. We collect $\boldsymbol{x}^+_i$, $\boldsymbol{x}^-_i$, $(\boldsymbol{x}^1_i)^+$, $(\boldsymbol{x}^2_i)^+$, $(\boldsymbol{x}^1_i)^-$, and $(\boldsymbol{x}^2_i)^-$ into matrices $\boldsymbol{X}^+ \in \mathbb{R}^{p^+ \times d}$, $\boldsymbol{X}^- \in \mathbb{R}^{p^- \times d}$, $\boldsymbol{X}^+_1, \boldsymbol{X}^+_2 \in \mathbb{R}^{p^+ \times d}$, and $\boldsymbol{X}^-_1, \boldsymbol{X}^-_2 \in \mathbb{R}^{p^- \times d}$.

The margins γ_1, γ_2, and γ (the margin of $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$) are given by

$$\begin{aligned}
\gamma_1 &\coloneqq \min_{\boldsymbol{\alpha} \in (\mathbb{R}^{+,p^+}, \boldsymbol{\beta} \in \mathbb{R}^{+,p^-}, \mathbf{1}^\intercal \boldsymbol{\alpha} = 1 = \mathbf{1}^\intercal \boldsymbol{\beta}} \|(\boldsymbol{X}^+_1)^\intercal \boldsymbol{\alpha} - (\boldsymbol{X}^-_1)^\intercal \boldsymbol{\beta}\| \\
\gamma_2 &\coloneqq \min_{\boldsymbol{\alpha} \in \mathbb{R}^{+,p^+}, \boldsymbol{\beta} \in \mathbb{R}^{+,p^-}, \mathbf{1}^\intercal \boldsymbol{\alpha} = 1 = \mathbf{1}^\intercal \boldsymbol{\beta}} \|(\boldsymbol{X}^+_2)^\intercal \boldsymbol{\alpha} - (\boldsymbol{X}^-_2)^\intercal \boldsymbol{\beta}\| \\
\gamma &\coloneqq \min_{\boldsymbol{\alpha} \in \mathbb{R}^{+,p^+}, \boldsymbol{\beta} \in \mathbb{R}^{+,p^-}, \mathbf{1}^\intercal \boldsymbol{\alpha} = 1 = \mathbf{1}^\intercal \boldsymbol{\beta}} \|(\boldsymbol{X}^+)^\intercal \boldsymbol{\alpha} - (\boldsymbol{X}^-)^\intercal \boldsymbol{\beta}\|,
\end{aligned}$$

where $\mathbb{R}^+$ is the set of positive real numbers and $\mathbf{1}$ is a vector of ones of appropriate size. We have

$$\begin{aligned}\|(\boldsymbol{X}^+)^\intercal\boldsymbol{\alpha} - (\boldsymbol{X}^-)^\intercal\boldsymbol{\beta}\|^2 &= \|(\boldsymbol{X}_1^+)^\intercal\alpha_1 + (\boldsymbol{X}_2^+)^\intercal\alpha_2 - (\boldsymbol{X}_1^-)^\intercal\beta_1 - (\boldsymbol{X}_2^-)^\intercal\beta_2\|^2\\ &= \|((\boldsymbol{X}_1^+)^\intercal\alpha_1 - (\boldsymbol{X}_1^-)^\intercal\beta_1) + ((\boldsymbol{X}_2^+)^\intercal\alpha_2 - (\boldsymbol{X}_2^-)^\intercal\beta_2)\|^2\\ &= \|(\boldsymbol{X}_1^+)^\intercal\alpha_1 - (\boldsymbol{X}_1^-)^\intercal\beta_1\|^2 + \|(\boldsymbol{X}_2^+)^\intercal\alpha_2 - (\boldsymbol{X}_2^-)^\intercal\beta_2\|^2.\end{aligned}$$

The latter terms attain, by assumption, minimal values of γ_1 and γ_2, respectively. Thus, $\gamma^2 = \gamma_1^2 + \gamma_2^2$. Also note that $\|\boldsymbol{x}_i\|^2 \leq r_1^2 + r_2^2$ for all $i \in I$. This implies that $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ is $(\sqrt{r_1^2 + r_2^2}, \sqrt{\gamma_1^2 + \gamma_2^2})$-separable. □

Existence of regular graphs The following result ensures the existence of enough regular graphs needed for the proof of Theorem 2 and its variants.

Lemma 21. For any even n and all $i \in \{0, \ldots, n-1\}$, there exists an i-regular graph with one orbit containing all vertices.

Proof. Let n be even, and let c be an arbitrary natural number. We define

$$E_{\text{odd}} \coloneqq \{(i, i + {}^{n}\!/_{2}) \mid i \in [{}^{n}\!/_{2}]\},$$

and

$$E_c \coloneqq \{(i, i + c \bmod n) \mid i \in [n]\},$$

where mod is the modulo operator with equivalence classes $[n]$. It is easily verified that for any $C \in \mathbb{N}$, $([n], \bigcup_{c\in[C]} E_c)$ is a $2C$-regular graph. Also, $([n], E_{\text{odd}} \cup \bigcup_{c\in[C]} E_c)$ is a $2C+1$-regular graph. The permutation, in cycle notation, $(1, 2, \ldots, n)$ is an automorphism for both graphs, implying that all vertices are in the same orbit. □

Remark 22. For any odd n, no i-regular graph exists with i odd. This is a classical textbook question that can be verified by handshaking. For regular graphs,

$$\sum_{i\in[n]} \mathsf{deg}(i) = i \cdot n.$$

Summing the degrees for each vertex counts each edge twice. Thus, $i \cdot n$ must be even, and since n is odd, i must be even.

E.2. Expressive power of enhanced variants

We now prove results on the expressive power of the 1-$\mathsf{WL}_{\mathcal{F}}$.

Proposition 23. Let G be a graph and $\mathcal{F}$ be a set of graphs. Then, for all rounds, the 1-$\mathsf{WL}_{\mathcal{F}}$ distinguishes at least the same vertices as the 1-WL.

Proof. Using, induction on t, we show that, for all vertices $v, w \in V(G)$,

$$C_t^{1,\mathcal{F}}(v) = C_t^{1,\mathcal{F}}(w) \text{ implies } C_t^1(v) = C_t^1(w). \tag{5}$$

The base case, $t = 0$, is clear since 1-$\mathsf{WL}_{\mathcal{F}}$ refines the single color class induced by C_0^1. For the induction, assume that Equation (5) holds and assume that, $C_{t+1}^{1,\mathcal{F}}(v) = C_{t+1}^{1,\mathcal{F}}(w)$ holds. Hence, $C_t^1(v) = C_t^1(w)$ and

$$\{\!\!\{C_t^{1,\mathcal{F}}(a) \mid a \in N(v)\}\!\!\} = \{\!\!\{C_t^{1,\mathcal{F}}(b) \mid b \in N(w)\}\!\!\}$$

holds. Hence, there is a *color-preserving bijection* $\varphi\colon N(v) \to N(w)$ between the above two multisets, i.e., $C_t^{1,\mathcal{F}}(a) = C_t^{1,\mathcal{F}}(\varphi(a))$, for $a \in N(v)$. Hence, by Equation (5), $C_t^1(a) = C_t^1(\varphi(a))$, for $a \in N(v)$. Consequently, it holds that $C_{t+1}^1(v) = C_{t+1}^1(w)$, proving the desired result. □

In addition, by choosing the set of graphs $\mathcal{F}$ appropriately, 1-$\mathsf{WL}_{\mathcal{F}}$ gets strictly more expressive than 1-WL in distinguishing non-isomorphic graphs.

Proposition 24. For every $n \geq 6$, there exists at least one pair of non-isomorphic graphs and a set of graphs $\mathcal{F}$ containing a single constant-order graph, such that, for all rounds, 1-WL does not distinguish them while 1-$\mathsf{WL}_{\mathcal{F}}$ distinguishes them after a single round.

Proof. For $n = 6$, we can choose a pair of a 6-cycle and the disjoint union of two 3-cycles. Since both graphs are 2-regular, the 1-WL cannot distinguish them. By choosing $\mathcal{F} = \{C_3\}$, the 1-$\mathsf{WL}_{\mathcal{F}}$ distinguishes them. For $n > 6$, we can simply pad the graphs with $n - 6$ isolated vertices. □

E.3. Margin-based upper and lower bounds on the VC dimension of Weisfeiler–Leman-based kernels

We first state the upper bound that we will be using for all the following cases, which is a classical result, for instance based on fat-shattering.

Lemma 25 (Theorem 1.6 in (Bartlett & Shawe-Taylor, 1999))**.** Let $\mathbb{S} \subseteq \mathbb{R}^d$.

$$\mathsf{VC\text{-}dim}(\mathbb{H}_{r,\lambda}(\mathbb{S})) \in O(r^2/\lambda^2).$$

We now prove the VC dimension theory results from the main paper. In the following, we will reuse our notation of splitting up $\mathbb{R}^d$ into $\mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$. We write $\boldsymbol{x}_i = (\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)})$ with $\boldsymbol{x}_i^{(1)} \in \mathbb{R}^{d_1}$ and $\boldsymbol{x}_i^{(2)} \in \mathbb{R}^{d_2}$. Further, let $(\boldsymbol{x}_i^{(1)})^+ \coloneqq (\boldsymbol{x}_i^{(1)}, 0)$, and $(\boldsymbol{x}_i^{(2)})^+ \coloneqq (0, \boldsymbol{x}_i^{(2)})$.

Lemma 26 (Lemma 1 in the main paper)**.** Let $\mathbb{S} \subseteq \mathbb{R}^d$. If $\mathbb{S}$ contains $m \coloneqq \lfloor r^2/\lambda^2 \rfloor$ vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m \in \mathbb{R}^d$ with $\boldsymbol{b}_i \coloneqq (\boldsymbol{b}_i^{(1)}, \boldsymbol{b}_i^{(2)})$ and $\boldsymbol{b}_1^{(2)}, \ldots, \boldsymbol{b}_m^{(2)}$ being pairwise orthogonal, $\|\boldsymbol{b}_i\| = r'$, and $\|\boldsymbol{b}_i^{(2)}\| = r$, then

$$\mathsf{VC\text{-}dim}(\mathbb{H}_{r',\lambda}(\mathbb{S})) \in \Omega(r^2/\lambda^2).$$

Proof. Following the argument in Alon et al. (2021), we show that the vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ can be shattered. Indeed, let A and B be two arbitary sets partitioning $[m]$. Consider the vector

$$\boldsymbol{w} \coloneqq \frac{\lambda}{r^2}\left(\sum_{i \in A} (\boldsymbol{b}_i^{(2)})^+ - \sum_{i \in B} (\boldsymbol{b}_i^{(2)})^+\right).$$

We observe that, because of assumptions underlying the vectors $\boldsymbol{b}_i$, we have

$$\boldsymbol{w}^\intercal \boldsymbol{b}_j = \begin{cases} \left(\frac{\lambda}{r^2}\right) \cdot (\boldsymbol{b}_j^{(2)})^\intercal \boldsymbol{b}_j^{(2)} = \lambda & \text{if } j \in A \\ -\left(\frac{\lambda}{r^2}\right) \cdot (\boldsymbol{b}_j^{(2)})^\intercal \boldsymbol{b}_j^{(2)} = -\lambda & \text{if } j \in B. \end{cases}$$

In other words, $\boldsymbol{w}$ witnesses that the distance between the convex hull of $\{\boldsymbol{b}_i \mid i \in A\}$ and $\{\boldsymbol{b}_i \mid i \in B\}$ is at least 2λ, implying the result. □

In the following, we will heavily rely upon Lemma 26 and more specifically we can construct $m = \lfloor r^2/\lambda^2 \rfloor$ graphs. Since we will be using regular graphs for simplicity where each regular graph has different regularity, we require $n \geq m$, which is true for $n \geq r^2/\lambda^2$. Notice that this requirement can be relaxed, and we could, for instance, consider graphs with nodes of two regularities, which would significantly lower the requirement on n. However, for these graphs, the construction and proofs would become significantly more complex as we would have to additionally deal with signal propagation within these graphs until we can guarantee orthogonality of the 1-WL-feature vectors. For this type of proof, we believe $e^{n/e} \geq r^2/\lambda^2$ would need to hold. However, we leave this to future research.

Theorem 27. For any $T, \lambda > 0$, we have,

$$\mathsf{VC\text{-}dim}(\mathbb{H}_{\sqrt{T+1}n,\lambda}(\mathcal{E}_{\mathsf{WL}}(n, d_T))) \in \Omega(r^2/\lambda^2), \text{ for } r = \sqrt{T}n \text{ and } n \geq r^2/\lambda^2,$$
$$\mathsf{VC\text{-}dim}(\mathbb{H}_{1,\lambda}(\bar{\mathcal{E}}_{\mathsf{WL}}(n, d_T))) \in \Omega(1/\lambda^2), \text{ for } r = \sqrt{T/(T+1)} \text{ and } n \geq r^2/\lambda^2.$$

Proof. The upper bounds follow from the general upper bound described earlier. For the lower bound, we show that for even $n \geq r^2/\lambda^2$, there exist $m = \lfloor r^2/\lambda^2 \rfloor$ graphs $G_1, \ldots, G_m$ in $\mathcal{G}_n$ such that the vectors $\boldsymbol{b}_i \coloneqq \phi^{(1)}_{\mathsf{WL}}(G_i)$ and $\bar{\boldsymbol{b}}_i \coloneqq \bar{\phi}^{(1)}_{\mathsf{WL}}(G_i)$ satisfy the assumptions of Lemma 26. Indeed, we can simply consider G_i to be an $(i-1)$-regular graph of order n; see Lemma 21. We break up the feature vectors into two parts: a one-dimensional part corresponding to the information related to the initial color and the remaining part containing all other information. We remark that for the 1-WL and for unlabeled graphs, all vertices have the same initial color. The interesting information is contained in the second part. If we inspect the 1-WL feature vectors, excluding the initial colors, for $T = 1$ of G_i, we obtain $(0, \ldots, \underbrace{n}_{\text{pos } i}, \ldots, 0)$ in the unnormalized case, and $\frac{1}{\sqrt{1+1n}}(0, \ldots, \underbrace{n}_{\text{pos } i}, \ldots, 0)$ in the normalized case. It is readily verified that $\boldsymbol{b}_i^{(2)} \coloneqq (0, \ldots, \underbrace{n}_{\text{pos } i}, \ldots, 0)$ and $\boldsymbol{b}_i^{(1)}$ being

the remaining initial colors are vectors satisfying the assumptions of Lemma 26 in the unnormalized case. For larger T, $\boldsymbol{b}_i^{(2)}$ is $\phi^{(1)}_{\mathsf{WL}}(G_i)$ except for the initial colors. Note that $\|\boldsymbol{b}_i^{(2)}\| = \sqrt{T}n = r$ and $\|\boldsymbol{b}_i\| = \sqrt{T+1}n \colon = r'$. For the normalized case, one simply needs to rescale with $1/r'$. Note that for $T > 0$, $1/2 \leq r^2/r'^2 < 1$. This implies a lower bound of $\Omega(\frac{r^2}{\lambda^2})$ in the unnormalized case, and $\Omega(\frac{r^2}{\lambda^2 r'^2}) = \Omega(\frac{1}{\lambda^2})$ in the normalized case.

So far, we assumed n to be even. For odd n, there is a slight technicality in that we can construct all r-regular graphs where r is even, i.e., we can construct $n+1/2$ regular graphs. Analogously this means for odd n and $n+1/2 \geq r^2/\lambda^2$, which is equivalent to $n \geq 2r^2/\lambda^2 - 1$, by a slight variant of Lemma 26 this implies a lower bound of $\Omega(2r^2/\lambda^2 - 1) = \Omega(r^2/\lambda^2)$. Analogous to the normalized case can be considered for odd n and results in the same bound, which proves the desired result. □

Theorem 28. Let $\mathcal{F}$ be a finite set of graphs. For any $T, \lambda > 0$, we have,

$$\mathsf{VC\text{-}dim}(\mathbb{H}_{\sqrt{T+1}n,\lambda}(\mathcal{E}_{\mathsf{WL},\mathcal{F}}(n, d_T))) \in \Omega(r^2/\lambda^2), \text{ for } r = \sqrt{T}n \text{ and } n \geq r^2/\lambda^2$$
$$\mathsf{VC\text{-}dim}(\mathbb{H}_{1,\lambda}(\bar{\mathcal{E}}_{\mathsf{WL},\mathcal{F}}(n, d_T))) \in \Omega(1/\lambda^2), \text{ for } r = \sqrt{T/(T+1)} \text{ and } n \geq r^2/\lambda^2.$$

Proof. This proof is analogous to the proof of Theorem 27. Note that in the proof above, we can choose the regular graphs such that all vertices in one graph are in the same orbit; see Lemma 21. This implies that if one vertex is colored according to $\mathcal{F}$, all vertices are colored in the same color, and the feature vectors $\phi^{(1)}_{\mathsf{WL},\mathcal{F}}(G_i)$ look exactly as described before, implying the result. □

A careful reader might wonder why we did not consider the initial colors in the proofs above. In the 1-WL-case, the initial colors are the same for all graphs in $\mathcal{G}_n$, i.e., the 1-WL feature vectors take the form $(n, \dots)$. We could leverage this to reduce the radius of the hypothesis class slightly. However, when considering the 1-WL$_{\mathcal{F}}$-case, the graphs in $\mathcal{F}$ change the initial colors. Because of our regular graph construction from Lemma 21, all nodes within one graph share the same color, determined by a subset $F \subseteq \mathcal{F}$, where F contains all graphs that are subgraphs of the regular graph in question. Hence, $2^{|\mathcal{F}|}$ possible initial colorings of graphs in $\mathcal{G}_n$ exists. Also, in both cases, our regular graphs are not necessarily orthogonal in the dimensions of these initial colors. Therefore, we disregarded them in the constructions of $\boldsymbol{w}$ above.

Theorem 29. For any $T, \lambda > 0$, we have,

$$\mathsf{VC\text{-}dim}(\mathbb{H}_{\sqrt{(T+1)n},\lambda}(\mathcal{E}_{\mathsf{WLOA}}(n, d_T))) \in \Omega(r^2/\lambda^2), \text{ for } r = \sqrt{Tn} \text{ and } n \geq r^2/\lambda^2,$$
$$\mathsf{VC\text{-}dim}(\mathbb{H}_{1,\lambda}(\bar{\mathcal{E}}_{\mathsf{WLOA}}(n, d_T))) \in \Omega(1/\lambda^2), \text{ for } r = \sqrt{T/(T+1)} \text{ and } n \geq r^2/\lambda^2.$$

Proof. This proof is analogous to the proof of Theorem 27 except $\|\phi^{(1)}_{\mathsf{WLOA}}(G_i)\| = \sqrt{(T+1)n} =: r'$ and $\|e_i\| = \sqrt{Tn} = r$. This implies a lower bound of $\Omega(\frac{r^2}{\lambda^2})$ in the unnormalized case, and $\Omega(\frac{r^2}{\lambda^2 r'^2}) = \Omega(\frac{1}{\lambda^2})$ in the normalized case, as desired. □

Theorem 30. Let $\mathcal{F}$ be a finite set of graphs. For any $T, \lambda > 0$, we have,

$$\mathsf{VC\text{-}dim}(\mathbb{H}_{\sqrt{(T+1)n},\lambda}(\mathcal{E}_{\mathsf{WLOA},\mathcal{F}}(n, d_T))) \in \Omega(r^2/\lambda^2), \text{ for } r = \sqrt{Tn} \text{ and } n \geq r^2/\lambda^2,$$
$$\mathsf{VC\text{-}dim}(\mathbb{H}_{1,\lambda}(\bar{\mathcal{E}}_{\mathsf{WLOA},\mathcal{F}}(n, d_T))) \in \Omega(1/\lambda^2), \text{ for } r = \sqrt{T/(T+1)} \text{ and } n \geq r^2/\lambda^2.$$

Proof. This proof is analogous to the proofs of Theorems 28 and 29. □

Note that the upper bound and the previous theorems on lower bounds imply tight bounds in $\mathcal{O}$-notation.

Corollary 31 (Theorem 2 in the main paper)**.** For any $T, \lambda > 0$, we have,

$$\mathsf{VC\text{-}dim}(\mathbb{H}_{r,\lambda}(\mathcal{E}_{\mathsf{WL}}(n, d_T))) \in \Theta(r^2/\lambda^2), \text{ for } r = \sqrt{T+1}n \text{ and } n \geq r^2/\lambda^2,$$
$$\mathsf{VC\text{-}dim}(\mathbb{H}_{1,\lambda}(\bar{\mathcal{E}}_{\mathsf{WL}}(n, d_T))) \in \Theta(1/\lambda^2), \text{ for } r = \sqrt{T/(T+1)} \text{ and } n \geq r^2/\lambda^2.$$

Corollary 32 (Corollary 3 in the main paper)**.** Let $\mathcal{F}$ be a finite set of graphs. For any $T, \lambda > 0$, we have,

$$\mathsf{VC\text{-}dim}(\mathbb{H}_{r,\lambda}(\mathcal{E}_{\mathsf{WL},\mathcal{F}}(n, d_T))) \in \Theta(r^2/\lambda^2), \text{ for } r = \sqrt{T+1}n \text{ and } n \geq r^2/\lambda^2$$
$$\mathsf{VC\text{-}dim}(\mathbb{H}_{1,\lambda}(\bar{\mathcal{E}}_{\mathsf{WL},\mathcal{F}}(n, d_T))) \in \Theta(1/\lambda^2), \text{ for } r = \sqrt{T/(T+1)} \text{ and } n \geq r^2/\lambda^2.$$

Corollary 33 (Proposition 4 in the main paper)**.** For any $T, \lambda > 0$, we have,

$$\mathsf{VC\text{-}dim}(\mathbb{H}_{r,\lambda}(\mathcal{E}_{\mathsf{WLOA}}(n, d_T))) \in \Theta(r^2/\lambda^2), \text{ for } r = \sqrt{(T+1)n} \text{ and } n \geq r^2/\lambda^2,$$
$$\mathsf{VC\text{-}dim}(\mathbb{H}_{1,\lambda}(\bar{\mathcal{E}}_{\mathsf{WLOA}}(n, d_T))) \in \Theta(1/\lambda^2), \text{ for } r = \sqrt{T/(T+1)} \text{ and } n \geq r^2/\lambda^2.$$

Corollary 34 (Corollary 5 in the main paper)**.** Let $\mathcal{F}$ be a finite set of graphs. For any $T, \lambda > 0$, we have,

$$\mathsf{VC\text{-}dim}(\mathbb{H}_{r,\lambda}(\mathcal{E}_{\mathsf{WLOA},\mathcal{F}}(n, d_T))) \in \Theta(r^2/\lambda^2), \text{ for } r = \sqrt{(T+1)n} \text{ and } n \geq r^2/\lambda^2,$$
$$\mathsf{VC\text{-}dim}(\mathbb{H}_{1,\lambda}(\bar{\mathcal{E}}_{\mathsf{WLOA},\mathcal{F}}(n, d_T))) \in \Theta(1/\lambda^2), \text{ for } r = \sqrt{T/(T+1)} \text{ and } n \geq r^2/\lambda^2.$$

E.3.1. Colored margin bounds

Given $T \geq 0$ and $C \subseteq \mathbb{N}$, we say that a graph G has *color complexity* (C, T) if the first T iterations of 1-WL assign colors to G in the set C. Let $\mathcal{G}_{C,T}$ be the class of all graphs of color complexity (C, T). We note that $\mathcal{G}_{C,T}$ possibly contains infinitely many graphs. Indeed, if C corresponds to the color assigned by 1-WL to degree two nodes, then $\mathcal{G}_{C,T}$ contains all 2-regular graphs.

Let $\mathcal{E}(C, T, d)$ be a class of graph embedding methods consisting of mappings from $\mathcal{G}_{C,T}$ to $\mathbb{R}^d$. Separability is lifted to the setting by considering the set of partial concepts defined on $\mathcal{G}_{C,T}$, as follows

$$\mathbb{H}_{r,\lambda}(\mathcal{E}(C, T, d)) \coloneqq \Big\{ h \in \{0, 1, \star\}^{\mathcal{G}_{C,T}} \;\Big|\; \forall\, G_1, \ldots, G_s \in \mathsf{supp}(h) \colon (G_1, h(G_1)), \ldots, (G_s, h(G_s)) \text{ is } (r, \lambda)\text{-}\mathcal{E}(n, d)\text{-separable} \Big\}.$$

Let $\bar{\mathcal{E}}_{\mathsf{WL}}(C, T, d)$ be the class of embeddings corresponding to the normalized 1-WL kernel, i.e., $\bar{\mathcal{E}}_{\mathsf{WL}}(C, T, d) \coloneqq \{G \mapsto \overline{\phi}^{(T)}_{\mathsf{WL}}(G) \mid G \in \mathcal{G}_{C,T}\}$. We note that d is a constant depending on $|C|$ and T we denote this constant by $d_{C,T}$. An immediate consequence of the proof of Theorem 2 is that we can obtain a margin-bound for infinite classes of graphs.

Corollary 35. For any $T > 0$, $C \subseteq \mathbb{N}$, and $\lambda > 0$, such that $\mathcal{G}_{C,T}$ contains all regular graphs of degree $0, 1, \ldots, r^2/\lambda^2$, for $r = \sqrt{T/(T+1)}$, we have

$$\mathsf{VC\text{-}dim}(\mathbb{H}_{1,\lambda}(\bar{\mathcal{E}}_{\mathsf{WL}}(C, T, d_{C,T}))) \in \Theta(1/\lambda^2).$$ □

E.4. Margin-based bounds on the VC dimension of MPNNs and more expressive architectures

We now lift the above results for the 1-WL kernel to MPNNs. To prove Proposition 6, we show that $\mathcal{E}_{\mathsf{MPNN}}(n, d, T)$ contains $\mathcal{E}_{\mathsf{WL}}(n, d_T)$. Thereto, the following result shows that MPNNs can compute the 1-WL feature vector.

Proposition 36. Let $\mathcal{G}_n$ be the set of n-order graphs and let $S \subseteq \mathcal{G}_n$. Then, for all $T \geq 0$, there exists a sufficiently wide T-layered simple MPNN architecture $\mathsf{mpnn}_n \colon S \to \mathbb{R}^d$, for an appropriately chosen $d > 0$, such that, for all $G \in S$,

$$\mathsf{mpnn}_n(G) = \phi^{(T)}_{\mathsf{WL}}(G).$$

Proof. The proof follows the construction outlined in the proof of (Morris et al., 2023, Proposition 2). Let $s \coloneqq |S|$. Hence, sn is an upper bound for the number of colors computed by 1-WL over all s graphs in one iteration.

Now, by Morris et al. (2019, Theorem 2), there exists an MPNN architecture with feature dimension (at most) n and consisting of t layers such that for each graph $G \in S$ it computes 1-WL-equivalent vertex features $\boldsymbol{f}^{(t)}_v$ in $\mathbb{R}^{1\times n}$ for $v \in V(G)$. That is, for vertices v and w in $V(G)$ it holds that

$$\boldsymbol{f}^{(t)}_v = \boldsymbol{f}^{(t)}_w \iff C^1_T(v) = C^1_T(w).$$

We note, by the construction outlined in the proof of Morris et al. (2019, Theorem 2), that $\boldsymbol{f}_v^{(t)}$, for $v \in V(G)$, is defined over the rational numbers. We further note that we can construct a single MPNN architecture for all s graphs by applying Morris et al. (2019, Theorem 2) over the disjoint union of the graphs in S. This increases the width from n to sn. We now show how to compute the 1-WL feature vector of a single iteration t. The overall feature vector can be obtained by (column-wise) concatenation over all layers.

Since the vertex features are rational, there exists a number M in $\mathbb{N}$ such that $M \cdot \boldsymbol{f}_v^{(t)}$ is in $\mathbb{N}^{1\times sn}$ for all $v \in V(G)$ and $G \in S$, i.e., a vector over $\mathbb{N}$. Now, let

$$\boldsymbol{W}' = \begin{bmatrix} K^{sn-1} & \cdots & K^{sn-1} \\ \vdots & \cdots & \vdots \\ K^0 & \cdots & K^0 \end{bmatrix} \in \mathbb{N}^{sn\times 2sn},$$

for a sufficiently large $K > 0$, then $\mathbf{k}_v \coloneqq M \cdot \boldsymbol{f}_v^{(t)}\boldsymbol{W}'$, for vertex $v \in V(G)$ and graph $G \in S$, computes a vector $\mathbf{k}_v$ in $\mathbb{N}^{2sn}$ containing $2sn$ occurrences of a natural number uniquely encoding the color of the vertex v. We next turn $\mathbf{k}_v$ into a one-hot encoding. More specifically, we define

$$\mathbf{h}'_v = \mathsf{lsig}(\mathbf{k}_v \circ (\mathbf{w}'')^\intercal + \mathbf{b}),$$

where $\circ$ denotes element-wise multiplication, with $\mathbf{w}'' = (1, -1, 1, -1, \ldots, 1, -1) \in \mathbb{R}^{2sn}$ and $\mathbf{b} = (-c_1 - 1, c_1 + 1, -c_2 - 1, c_2 + 1, \ldots, -c_{sn} - 1, c_{sn} + 1) \in \mathbb{R}^{2sn}$with c_i the number encoding the ith color under 1-WL at iteration t on the set S. We note that for odd i,

$$(h'_v)_i \coloneqq \mathsf{lsig}(C^1_t(v) - c_i - 1) = \begin{cases} 1 & C^1_t(v) \geq c_i \\ 0 & \text{otherwise.} \end{cases}$$

and for even i,

$$(h'_v)_i \coloneqq \mathsf{lsig}(-C^1_t(v) + c_i + 1) = \begin{cases} 1 & C^1_t(v) \leq c_i \\ 0 & \text{otherwise.} \end{cases}$$

In other words, $((h'_v)_i, (h'_v)_{i+1})$ are both 1 if and only if $C^1_t(v) = c_i$. We thus obtain one-hot encoding of the color $C^1_t(v)$ by combining $((h'_v)_i, (h'_v)_{i+1})$ using an "AND" encoding (e.g., $\mathsf{lsig}(x + y - 1)$) applied to pairs of consecutive entries in $\mathbf{h}'_v$. That is,

$$\boldsymbol{h}_v \coloneqq \mathsf{lsig}\left(\boldsymbol{h}'_v \cdot \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 1 \end{pmatrix} - (1, 1, \ldots, 1) \right) \in \mathbb{R}^{sn}.$$

We obtain the overall 1-WL vector by row-wise summation and concatenation over all layers. We remark that, for a single iteration, the maximal width of the whole construction is $2sn$. □

By the above proposition, MPNNs of sufficient width can compute the 1-WL feature vectors. Moreover, the normalization can be included in the MPNN computation. Hence, we can prove the lower bound by simulating the proof of Theorem 2. The upper bound follows by the same arguments as described at the beginning of Section 3. The above result can be easily extended to the 1-$\mathsf{WL}_{\mathcal{F}}$, implying Corollary 7.

Corollary 37. Let $\mathcal{G}_n$ be the set of n-order graphs, let $S \subseteq \mathcal{G}_n$, and let $\mathcal{F}$ be a set of graphs. Then, for all $T \geq 0$, there exists a sufficiently wide T-layered MPNN architecture $\mathsf{mpnn}_n \colon S \to \mathbb{R}^d$, for an appropriately chosen $d > 0$, such that, for all $G \in S$,

$$\mathsf{mpnn}_n(G) = \phi^{(T)}_{1\text{-}\mathsf{WL}_{\mathcal{F}}}(G).$$

Proof sketch. By definition of the 1-$\mathsf{WL}_{\mathcal{F}}$, the algorithm is essentially the 1-WL operating on a specifically vertex-labeled graph. Since Morris et al. (2019, Theorem 2) also works for vertex-labeled graphs, the proof technique for Proposition 36 can be straightforwardly lifted to the 1-$\mathsf{WL}_{\mathcal{F}}$. □

We can also extend Proposition 36 to the 1-WLOA and 1-WLOA$_{\mathcal{F}}$, i.e., derive an MPNN architecture that can compute 1-WLOA's and 1-WLOA$_{\mathcal{F}}$'s feature vectors. By that, we can extend Proposition 4 and Corollary 5 to their corresponding MPNN versions.

Proposition 38. Let $\mathcal{G}_n$ be the set of n-order graphs and let $S \subseteq \mathcal{G}_n$. Then, for all $T \geq 0$, there exists a sufficiently wide T-layered MPNN architecture $\mathsf{mpnn}_n \colon S \to \mathbb{R}^d$, for an appropriately chosen $d > 0$, such that, for all $G \in S$,

$$\mathsf{mpnn}_n(G) = \phi^{(T)}_{\mathsf{WLOA}}(G).$$

Proof. By Proposition 36, there exists a T-layered MPNN architecture $\mathsf{mpnn}_n \colon S \to \mathbb{R}^d$, for an appropriately chosen $d > 0$, such that, for all $G \in S$,

$$\mathsf{mpnn}_n(G) = \phi^{(T)}_{\text{1-WL}}(G).$$

We now show how to transform $\phi^{(T)}_{\text{1-WL}}(G)$ into $\phi^{(T)}_{\mathsf{WLOA}}(G)$. We show the transformation for a single iteration $t \leq T$, i.e., transforming $\phi_{t,\text{1-WL}}(G)$ into $\phi_{t,\text{1-WLOA}}(G)$. Let C denote the number of colors at iteration t of the 1-WL over all $|S|$ graphs. Since n is finite, C is finite as well. That is, $\phi_{t,\text{1-WL}}(G)$ has C entries. Hence, the number of components for $\phi_{t,\mathsf{WLOA}}(G)$ is at most Cn. By multiplying $\phi_{t,\text{1-WL}}(G)$ with an appropriately chosen matrix $\boldsymbol{M} \in \{0,1\}^{C \times Cn}$, we get a vector $\boldsymbol{r} \in \mathbb{R}^{Cn}$, where each entry of $\phi_{t,\text{1-WL}}(G)$ is repeated n times. Specifically,

$$\boldsymbol{M} \coloneqq \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \in \{0,1\}^{C \times Cn}.$$

Now let

$$\boldsymbol{b} \coloneqq (1, 2, \ldots, n, 1, 2, \ldots, n, \ldots, 1, 2, \ldots, n) \in \mathbb{R}^{Cn} \quad \text{and} \quad \boldsymbol{r}' \coloneqq \mathsf{sign}(\boldsymbol{r} - \boldsymbol{b}).$$

Observe that $\boldsymbol{r}' = \phi_{t,\text{1-WLOA}}(G)$, implying the result □

In a similar way as for Corollary 37, we can lift the above result to the 1-WLOA$_{\mathcal{F}}$.

Corollary 39. Let $\mathcal{G}_n$ be the set of n-order graphs, let $S \subseteq \mathcal{G}_n$, and let $\mathcal{F}$ be a set of graphs. Then, for all $T \geq 0$, there exists a sufficiently wide T-layered MPNN architecture $\mathsf{mpnn}_n \colon S \to \mathbb{R}^d$, for an appropriately chosen $d > 0$, such that, for all $G \in S$,

$$\mathsf{mpnn}_n(G) = \phi^{(T)}_{\text{1-WLOA}_{\mathcal{F}}}(G).$$ □

E.5. Increased separation power of the 1-WL$_{\mathcal{F}}$

We now prove how more expressive architectures help to separate the data.

Lemma 40 (Proposition 8 in the main paper)**.** For every $n \geq 6$, there exists a pair of non-isomorphic n-order graphs (G_n, H_n) and a graph F such that, for $\mathcal{F} \coloneqq \{F\}$ and for all number of rounds $T \geq 0$, it holds that

$$\left\| \overline{\phi^{(T)}_{\mathsf{WL}}}(G_n) - \overline{\phi^{(T)}_{\mathsf{WL}}}(H_n) \right\| = 0, \quad \text{and} \quad \left\| \overline{\phi^{(T)}_{\mathsf{WL},\mathcal{F}}}(G_n) - \overline{\phi^{(T)}_{\mathsf{WL},\mathcal{F}}}(H_n) \right\| = \sqrt{2}.$$

Proof. Let $G_n \coloneqq C_n$, a cycle on n vertices. Further, let $H_n \coloneqq C_{\lceil n/2 \rceil} \dot\cup C_{\lfloor n/2 \rfloor}$, a disjoint union of cycles on $\lceil n/2 \rceil$ and $\lfloor n/2 \rfloor$ vertices, respectively. Since the graphs G_n and H_n are regular, 1-WL cannot distinguish them. Hence, the distances of the two feature vectors $\overline{\phi^{(T)}_{\mathsf{WL}}}(G_n)$ and $\overline{\phi^{(T)}_{\mathsf{WL}}}(H_n)$ is zero. Moreover, by setting $\mathcal{F} \coloneqq \{C_{\lfloor n/2 \rfloor}\}$, 1-WL$_{\mathcal{F}}$ reaches the stable

coloring at iteration $t = 0$ on both graphs and distinguishes the two graphs. In addition, the feature vectors $\overline{\phi^{(T)}_{\mathsf{WL},\mathcal{F}}}(G_n)$ and $\overline{\phi^{(T)}_{\mathsf{WL},\mathcal{F}}}(H_n)$ are orthonormal for all choices of $T \geq 0$, implying the result. □

Proposition 41. For every $n \geq 6$, there exists a pair of non-isomorphic n-order graphs (G_n, H_n) and set of graphs $\mathcal{F}$ of cardinality one, such that, for all number of layers $T \geq 0$, and widths $d > 0$, and all $m \in \mathsf{MPNN}_{\mathsf{mlp}}(d, T)$, it holds that

$$\left\| m(G_n) - m(H_n) \right\| = 0,$$

while for sufficiently large $d > 0$, there exists an $\widehat{m} \in \mathsf{MPNN}_{\mathsf{mlp},\mathcal{F}}(d, T)$, such that

$$\left\| \widehat{m}(G_n) - \widehat{m}(H_n) \right\| = \sqrt{2}.$$

Proof. We use the same graphs as in the proof of Lemma 40. The first statement is a direct implication of Morris et al. (2019, Theorem 1), i.e., any MPNN is upper bounded by the 1-WL in distinguishing vertices. The second statement follows from Proposition 36 and Corollary 37. That is, an $\mathsf{MPNN}_{\mathcal{F}}$ architecture of sufficient width exactly computes the $1\text{-}\mathsf{WL}_{\mathcal{F}}$ feature vectors for the graphs of Lemma 40, as shown in the proof of Proposition 36. □

Proposition 42 (Proposition 9 in the main paper)**.** For every $n \geq 10$, there exists a set of pair-wise non-isomorphic (at most) n-order graphs S, a concept $c \colon S \to \{0, 1\}$, and a graph F, such that the graphs in the set S,

1. are pair-wise distinguishable by 1-WL after one round,
2. are *not* linearly separable under the normalized 1-WL feature vector $\overline{\phi^{(T)}_{\mathsf{WL}}}$, concerning the concept c, for any $T \geq 0$,
3. and are linearly separable under the normalized $1\text{-}\mathsf{WL}_{\mathcal{F}}$ feature vector $\overline{\phi^{(T)}_{\mathsf{WL},\mathcal{F}}}$, concerning the concept c and , for all $T \geq 0$, where $\mathcal{F} \coloneqq \{F\}$.

Moreover, the results also work for the unnormalized feature vectors.

Proof. We first outline the construction for the set S with $|S| = 4$. However, the construction can easily be generalized to larger cardinalities. Concretely, we construct the graphs $G_1, \ldots, G_4$. For odd i, the graph G_i consists of the disjoint union of i isolated vertices and a disjoint union of cycles on $\lceil n/2 \rceil - 2$ and $\lfloor n/2 \rfloor - 2$ vertices. Set $c(G_i) \coloneqq 0$. For even i, the graph G_i consists of the disjoint union of i isolated vertices and a cycle on $n - 4$ vertices. Set $c(G_i) \coloneqq 1$.

We proceed by showing items 1 to 3. Since the order of the graphs G_1 to G_4 is pair-wise different 1-WL pair-wise distinguishes the graphs, showing item 1. We proceed with item 2. Since the disjoint union of cycles on $\lceil n/2 \rceil - 2$ and $\lfloor n/2 \rfloor - 2$ vertices are 2-regular, the 1-WL cannot distinguish them. Therefore, the four 1-WL feature vectors have the following forms,

$$\begin{aligned}
\phi^{(T)}_{\mathsf{WL}}(G_1) &= (n-4, 1, n-4, 1, \ldots, n-4, 1),\\
\phi^{(T)}_{\mathsf{WL}}(G_2) &= (n-4, 2, n-4, 2, \ldots, n-4, 2),\\
\phi^{(T)}_{\mathsf{WL}}(G_3) &= (n-4, 3, n-4, 3, \ldots, n-4, 3),\\
\phi^{(T)}_{\mathsf{WL}}(G_4) &= (n-4, 4, n-4, 4, \ldots, n-4, 4).
\end{aligned}$$

Since, for each round $t \geq 0$, the feature vectors have the form $(n-4, 1)$, $(n-4, 2)$, $(n-4, 3)$, and $(n-4, 4)$, respectively, the resulting four unnormalized 1-WL feature vectors are co-linear and, by the construction, of the concept c, they are not linearly separable.

Note that, for each round $t \geq 0$, the feature vectors only differ by at most 3 in the second components. Hence, their respective ℓ_2 norms are controlled by n, and their respective ℓ_2 norms are also close. For a single iteration, when projecting the feature vectors onto the 1-dimensonal unit sphere, by dividing by their respective ℓ_2, the lexicographic order is preserved on the unit sphere, making them not linearly separable. For $t \geq 1$, notice that the vectors are just concatenations of the above. We can write the original vectors for $t = 0$ as $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, $\boldsymbol{g}_3$, and $\boldsymbol{g}_4$ and for larger $t > 0$ as $[\boldsymbol{g}_1, \ldots, \boldsymbol{g}_1], \ldots, [\boldsymbol{g}_4, \ldots, \boldsymbol{g}_4]$. Assume, for $t > 0$, that the aforementioned vectors are linearly separable by a hyperplane $0 = \boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} + b$, then, by

definition, we can decompose $\boldsymbol{w} = [\boldsymbol{w}_0, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_t]$ such that $(\mathsf{sign}(\sum_{i=0}^{T} \boldsymbol{w}_i^{\intercal} \boldsymbol{g}_i + b) + 1)/2 = c(\boldsymbol{g}_j)$ and by construction $\sum_{i=0}^{T} \boldsymbol{w}_i^{\intercal} \boldsymbol{g}_j + b = (\sum_{i=0}^{T} \boldsymbol{w}_i)^{\intercal} \boldsymbol{g}_j + b$. Thus, such $\boldsymbol{w}$ and b would verify that $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_4$ are linearly separable, which is impossible. This implies that even for larger $t > 0$ $\phi^{(T)}_{\mathsf{WL}}(G_1), \ldots, \phi^{(T)}_{\mathsf{WL}}(G_4)$ are not linearly separable.

For item 3, we set $\mathcal{F} \coloneqq \{C_{n-4}\}$, the cycle on $n-4$ vertices. Observe that the 1-$\mathsf{WL}_{\mathcal{F}}$ feature vectors at $T = 0$ for all four graphs have three components. Concretely,

$$\begin{aligned}
\phi^{(0)}_{\mathsf{WL},\mathcal{F}}(G_1) &= (1, 0, n-4),\\
\phi^{(0)}_{\mathsf{WL},\mathcal{F}}(G_2) &= (2, n-4, 0),\\
\phi^{(0)}_{\mathsf{WL},\mathcal{F}}(G_3) &= (3, 0, n-4),\\
\phi^{(0)}_{\mathsf{WL},\mathcal{F}}(G_4) &= (4, n-4, 0).
\end{aligned}$$

Further, note that, for all four graphs, 1-$\mathsf{WL}_{\mathcal{F}}$ reaches the stable coloring at $T = 0$. Hence, for all $T \geq 0$, the four vectors are linearly separable concerning the concept c. Further, by dividing by their respective ℓ_2 norms, this property is preserved. That is, for graphs with class label 0, the first and third graph, third component of the 1-WL feature vectors are equal to $(n-4)$ while for the other two graphs, this component is 0. Hence, we can easily find a vector $\boldsymbol{w} \in \mathbb{R}^3$ that linearly separates the normalized 1-WL feature vectors. □

Proposition 43. Let $n \geq 6$ and let $\mathcal{F}$ be a finite set of graphs. Further, let $c \colon \mathcal{G}_n \to \{0, 1\}$ be a concept such that, for all $T \geq 0$, the graphs are *not* linearly separable under the normalized 1-WL feature vector $\overline{\phi^{(T)}_{\mathsf{WL}}}$, concerning the concept c. Further, assume that for all graphs $G \in \mathcal{G}_n$ for which $c(G) = 0$, it holds that there is at least one vertex $v \in V(G)$ such it is contained in a subgraph of G that is isomorphic to a graph in the set $\mathcal{F}$, while no such vertices exist in graphs G for which $c(G) = 1$. Then the graphs are linearly separable under the normalized 1-$\mathsf{WL}_{\mathcal{F}}$ feature vector $\overline{\phi^{(T)}_{\mathsf{WL},\mathcal{F}}}$, concerning the concept c.

Proof. By assumption, for graphs G with $c(G) = 0$ it holds that there exists an index $i \geq 0$ such that $\phi^{(0)}_{\mathsf{WL},\mathcal{F}}(G)_i \neq 0$ while for all graphs H with $c(H) = 1$ it holds that $\phi^{(0)}_{\mathsf{WL},\mathcal{F}}(H)_i = 0$. Without loss of generality, assume this is the case for $i = 0$. Hence, for all $T \geq 0$, we can find a vector $\boldsymbol{w} \coloneqq (1, 0, \ldots, 0) \in \mathbb{R}^d$ with an appropriate number of components d and $C > 0$ such that

$$\left\langle \boldsymbol{w}, \phi^{(T)}_{\mathsf{WL},\mathcal{F}}(G) \right\rangle = \begin{cases} > C, & \text{if } c(G) = 0,\\ < C, & \text{if } c(G) = 1. \end{cases}$$

Hence, the data is linearly separable by the 1-$\mathsf{WL}_{\mathcal{F}}$ for $T \geq 0$, showing the result. □

E.6. Results on shrinking the margin

Here, we prove negative results, showing that using more expressive architecture can also decrease the margin.

Proposition 44 (Proposition 10 in the main paper)**.** For every $n \geq 10$, there exists a pair of $2n$-order graphs (G_n, H_n) and a graph F, such that, for $\mathcal{F} \coloneqq \{F\}$ and for all number of rounds $T > 0$, it holds that

$$\left\| \overline{\phi^{(T)}_{\mathsf{WL},\mathcal{F}}}(G_n) - \overline{\phi^{(T)}_{\mathsf{WL},\mathcal{F}}}(H_n) \right\| < \left\| \overline{\phi^{(T)}_{\mathsf{WL}}}(G_n) - \overline{\phi^{(T)}_{\mathsf{WL}}}(H_n) \right\|.$$

Proof. Let G_n be the disjoint union of a complete graph on n vertices K_n and n isolated vertices. Further, let H_n be the disjoint union of the complete graph on three vertices K_3, the cycle on $(n-3)$ vertices C_{n-3}, and n isolated vertex. By construction and definition of the 1-WL feature vector, for $T > 0$,

$$\begin{aligned}
\phi^{(T)}_{\mathsf{WL}}(G_n) &= (2n, n, 0, n, \ldots, n, 0, n) \in \mathbb{R}^{1+3T}, \text{ and}\\
\phi^{(T)}_{\mathsf{WL}}(H_n) &= (2c, 0, n, n, \ldots, 0, n, n) \in \mathbb{R}^{1+3T}.
\end{aligned}$$

Moreover, by setting $\mathcal{F} \coloneqq \{C_3\}$ and by definition of the 1-$\mathsf{WL}_{\mathcal{F}}$ feature vector, for $T > 0$,

$$\begin{aligned}
\phi^{(T)}_{\mathsf{WL},\mathcal{F}}(G_n) &= (0, 2n, n, 0, 0, n, \ldots, n, 0, 0, n) \in \mathbb{R}^{2+4T} \text{ and}\\
\phi^{(T)}_{\mathsf{WL},\mathcal{F}}(H_n) &= (3, 2n-3, 0, n-3, 3, n, \ldots, 0, n-3, 3, n) \in \mathbb{R}^{2+4T}.
\end{aligned}$$

Since $\sqrt{2n^2} = \sqrt{2}n > \sqrt{3 \cdot 3^2 + n^2 + (n-3)^2}$ for $n \geq 10$, the statement is clear for the unnormalized 1-WL feature vector is clear. Now, for the normalized 1-WL feature vector, the component counting the n isolated vertices in each iteration $T > 0$ will sufficiently downgrade the (constant) contribution of the 1-$\mathsf{WL}_{\mathcal{F}}$ feature vector for iteration 0, leading to a larger ℓ_2 distance of the 1-WL feature vectors for iterations $T > 0$. That is, for the first two iterations $\|\overline{\phi^{(1)}_{\mathsf{WL}}}(G_n) - \overline{\phi^{(1)}_{\mathsf{WL}}}(H_n)\| = \sqrt{2/6}$, which, is always strictly larger than $\|\overline{\phi^{(1)}_{\mathsf{WL},\mathcal{F}}}(G_n) - \overline{\phi^{(1)}_{\mathsf{WL},\mathcal{F}}}(H_n)\|$, which can be verified by tedious calculation. The argument can be extended to iterations $t > 1$. □

E.7. Results on growing the margin

We now prove results showing that more expressive architectures often result in an increased margin.

Proposition 45 (Proposition 11 in the main paper)**.** Let $n > 0$ and G_n and H_n be two connected n-order graphs. Further, let $\mathcal{F} \coloneqq \{F\}$ such that there is at least one vertex in $V(G_n)$ contained in a subgraph of G_n isomorphic to the graph F. For the graph H_n, no such vertices exist. Further, let $T \geq 0$ be the number of rounds to reach the stable partition of G_n and H_n under 1-WL, and assume

$$(\phi^{(T)}_{\mathsf{WL}}(G_n), 1), (\phi^{(T)}_{\mathsf{WL}}(H_n), 0) \text{ is } (r_1, \lambda_1)\text{-separable, with } \lambda_1 < \sqrt{2n}.$$

Then,

$$(\phi^{(T)}_{\mathsf{WL},\mathcal{F}}(G_n), 1), (\phi^{(T)}_{\mathsf{WL},\mathcal{F}}(H_n), 0) \text{ is } (r_2, \lambda_2)\text{-separable, with } r_2 \leq r_1 \text{ and } \lambda_2 \geq \lambda_1.$$

Proof. First, by assumption, there exists a vertex $v \in V(G_n)$ that, by definition of the 1-$\mathsf{WL}_{\mathcal{F}}$, gets assigned a color at initialization of the 1-$\mathsf{WL}_{\mathcal{F}}$ that never gets assigned to any vertex in the graphs G_n and H_n under the 1-WL. Secondly, since the graphs G_n and H_n are connected, at the stable partition of G_n and H_n under 1-$\mathsf{WL}_{\mathcal{F}}$, it holds that

$$\langle \phi_{\mathcal{F},T}(G_n), \phi_{\mathcal{F},T}(H_n) \rangle = 0,$$

i.e., the feature vector of G_n and H_n under 1-WL are orthogonal. Hence, the ℓ_2 distance between the two vectors is $\sqrt{\|\phi_{\mathcal{F},T}(G_n)\|^2 + \|\phi_{\mathcal{F},T}(H_n)\|^2}$, due to the pythagorean theorem. This ℓ_2 distance is minimized for minimized norms, which, since the sum of all elements is n and all elements are in $\mathbb{N}$, is minimized for $\|\phi_{\mathcal{F},T}(G_n)\|^2 = n = \|\phi_{\mathcal{F},T}(H_n)\|^2$. Thus the above ℓ_2 distance is at least $\sqrt{2n}$m and

$$(\phi_{\mathcal{F},T}(G_n), 1), (\phi_{\mathcal{F},T}(H_n), 0) \text{ is } (n, \sqrt{2n})\text{-separable.}$$

We now just need to lower-bound by how much previous iterations can decrease this distance. Again, since G_n is connected, we know that at every iteration $t < T$, one vertex in the graphs G_n gets assigned a color that never gets assigned to H_n under the 1-$\mathsf{WL}_{\mathcal{F}}$. Since we want to lower bound the ℓ_2 distance between feature vectors between 1-$\mathsf{WL}_{\mathcal{F}}$ for all iteration $i < T$, we can assume that for the graph G_n, the normalized 1-$\mathsf{WL}_{\mathcal{F}}$ feature vector has the form $(n - (i+1), i+1)$, since this maximizes the ℓ_2 norm of the vector. For the graph H, we can assume the form $(n, 0)$, again maximizing the vector's ℓ_2 norm. For simplicity, we can assume that their ℓ_2 distance is 0. By applying Proposition 20 T times iteratively, it follows that

$$(\phi^{(T)}_{\mathsf{WL},\mathcal{F}}(G_n), 1), (\phi^{(T)}_{\mathsf{WL},\mathcal{F}}(H_n), 0) \text{ is } (\sqrt{T}n, \sqrt{2n})\text{-separable.}$$

Due to Proposition 23, we can further reduce the radius $\sqrt{T}n$ to be $r_2 \leq r_1$, since the colors under 1-$\mathsf{WL}_{\mathcal{F}}$ are finer than under 1-WL. Hence, by assumption, $\lambda_2 \geq \sqrt{2n} > \lambda_1$ and the result follows. □

Notice, that under mild assumptions due to Theorem 27 and Theorem 28, Proposition 45 implies that the VC dimension for the 1-$\mathsf{WL}_{\mathcal{F}}$ decreases, since $\frac{r_1^2}{\lambda_1^2} > \frac{r_2^2}{\lambda_2^2}$, and thus generalization improves.

Growing the margin for 1-$\mathsf{WL}_{\mathcal{F}}$ and 1-$\mathsf{WLOA}_{\mathcal{F}}$ It is challenging to improve Proposition 11, i.e., to derive weaker conditions such that 1-$\mathsf{WL}_{\mathcal{F}}$ provably leads to an increase of the margin over the 1-WL kernel. This becomes more feasible,

however, for the 1-$\mathsf{WLOA}_{\mathcal{F}}$ kernel. First, note that for two graphs G and H, it holds that

$$\begin{aligned} k_{\mathsf{WLOA}}(G,H) &\coloneqq \sum_{t\in[T]\cup\{0\}} \sum_{c\in\Sigma_t} \min(\phi_t(G)_c, \phi_t(H)_c) \\ &= \sum_{t\in[T]\cup\{0\}} \sum_{c\in\Sigma_t} \sum_{j=1}^{n} \mathbb{1}_{\phi_t(G)_c\geq j \wedge \phi_t(H)_c \geq j} \\ &= \sum_{t\in[T]\cup\{0\}} \sum_{c\in\Sigma_t} \sum_{j=1}^{n} \mathbb{1}_{\phi_t(G)_c\geq j} \mathbb{1}_{\phi_t(H)_c \geq j}. \end{aligned} \tag{6}$$

Equation (6) provides an intuition for the feature map of the kernel k_{WLOA}, namely, a unary encoding of the count for each color in each iteration. Such a feature map is quite natural as the inner product between unary encodings of a and b is the minimum of a and b. This also implies that, for a graph $G \in \mathcal{G}_n$, it holds that $\|\phi^{(T)}_{\mathsf{WLOA}}(G)\| = \sqrt{Tn}$, i.e., we can easily bound the norm of the feature vector.

Now, the 1-WLOA simplifies the computation of distances between two graphs $G, H \in \mathcal{G}_n$, since

$$\begin{aligned} \left\|\phi^{(T)}_{\mathsf{WLOA}}(G) - \phi^{(T)}_{\mathsf{WLOA}}(H)\right\| &= \sqrt{\left\|\phi^{(T)}_{\mathsf{WLOA}}(G)\right\|^2 + \left\|\phi^{(T)}_{\mathsf{WLOA}}(H)\right\|^2 - 2\phi^{(T)}_{\mathsf{WLOA}}(G)^\intercal \phi^{(T)}_{\mathsf{WLOA}}(H)} \\ &= \sqrt{2Tn - 2k^{(T)}_{\mathsf{WLOA}}(G,H)}. \end{aligned}$$

Therefore, due to the monotonicity of the square root, margin increases are directly controlled by the kernel value $k^{(T)}_{\mathsf{WLOA}}(G,H)$. For the 1-$\mathsf{WLOA}_{\mathcal{F}}$, since by Proposition 23 the 1-$\mathsf{WL}_{\mathcal{F}}$ computes a finer color partition than the 1-WL, pairwise distances can not decrease compared to the 1-WLOA, resulting in the following statement.

Proposition 46 (Proposition 12 in the main paper)**.** Let $\mathcal{F}$ be a finite set of graphs. Given two graphs G and H,

$$\left\|\phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G) - \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(H)\right\| \geq \left\|\phi^{(T)}_{\mathsf{WLOA}}(G) - \phi^{(T)}_{\mathsf{WLOA}}(H)\right\|. \qquad \square$$

Proof. Consider Proposition 23 applied to the disjoint union of G and H, which implies that any color class c in G or H after $t \leq T$ steps of 1-WL gets refined to be finer, i.e., $c = \bigcup_j c_j$ for some color classes c_j in G or H after t steps when applying $\mathcal{F}$. Note also that each color class is naturally partitioned into subsets in G and H. This then implies that 1-WLOA can only increase the pairwise distance when $\mathcal{F}$ is added. Note that for any color class c shattered into $c_1, \ldots, c_f$ as above,

$$\phi_t(G)_c = \sum_{j=1}^{f} \phi_{\mathcal{F},t}(G)_{c_j}, \quad \phi_t(H)_c = \sum_{j=1}^{f} \phi_{\mathcal{F},t}(H)_{c_j}.$$

This trivially implies that

$$\sum_{j=1}^{f} \min(\phi_{\mathcal{F},t}(G)_{c_j}, \phi_{\mathcal{F},t}(H)_{c_j}) \leq \min(\phi_t(G)_c, \phi_t(H)_c).$$

Which, since it holds for all color classes, implies

$$\begin{aligned} k^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G,H) &\leq k^{(T)}_{\mathsf{WLOA}}(G,H) \\ \sqrt{2Tn - 2k^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G,H)} &\geq \sqrt{2Tn - 2k^{(T)}_{\mathsf{WLOA}}(G,H)} \\ \left\|\phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G) - \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(H)\right\| &\geq \left\|\phi^{(T)}_{\mathsf{WLOA}}(G) - \phi^{(T)}_{\mathsf{WLOA}}(H)\right\|. \end{aligned}$$

This concludes the proof. $\square$

We now look into conditions under which the 1-$\mathsf{WL}_{\mathcal{F}}$ and 1-$\mathsf{WLOA}_{\mathcal{F}}$ provably increase the margin. The first result that we can directly state that holds for both 1-$\mathsf{WL}_{\mathcal{F}}$ and 1-$\mathsf{WLOA}_{\mathcal{F}}$ is the following direct application of Proposition 19.

Theorem 47. Let $(G_1, y_1), \ldots, (G_s, y_s)$ in $\mathcal{G}_n \times \{0, 1\}$ be a (graph) sample that is linearly separable in the 1-WLOA feature space with margin γ. If

$$\min_{y_i \neq y_j} \left\| \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G_i) - \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G_j) \right\|^2 - \left\| \phi^{(T)}_{\mathsf{WLOA}}(G_i) - \phi^{(T)}_{\mathsf{WLOA}}(G_j) \right\|^2 >$$
$$\max_{y_i = y_j} \left\| \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G_i) - \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G_j) \right\|^2 - \left\| \phi^{(T)}_{\mathsf{WLOA}}(G_i) - \phi^{(T)}_{\mathsf{WLOA}}(G_j) \right\|^2,$$

that is, the minimum increase in distances between classes is strictly larger than the maximum increase in distance within each class, then the margin λ increases when $\mathcal{F}$ is considered.

Proof. Direct application of Proposition 19 by considering $\boldsymbol{x}_i \coloneqq \phi^{(T)}_{\mathsf{WLOA}}(G_i)$ and $\tilde{\boldsymbol{x}}_i \coloneqq \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G_i)$. □

In fact, for the pairwise distances to increase strictly, we can use the following equivalences. Let $C_{\mathcal{F}}(c)$ be the set of colors that color c under 1-WL is split into under 1-WL$_{\mathcal{F}}$. I.e., $\phi_t(G)_c = \sum_{c' \in C_{\mathcal{F}}(c)} \phi_{\mathcal{F},t}(G)_{c'}$.

Proposition 48 (Implies Proposition 13 in the main paper)**.** The following statements are equivalent,

1. $\left\| \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G) - \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(H) \right\| = \left\| \phi^{(T)}_{\mathsf{WLOA}}(G) - \phi^{(T)}_{\mathsf{WLOA}}(H) \right\|$.
2. $k^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G, H) = k^{(T)}_{\mathsf{WLOA}}(G, H)$.
3. $\forall\, t \in [T] \cup \{0\}, \forall c \in \Sigma_t : \min(\phi_t(G)_c, \phi_t(H)_c) = \sum_{c' \in C_{\mathcal{F}}(c)} \min(\phi_{\mathcal{F},t}(G)_{c'}, \phi_{\mathcal{F},t}(H)_{c'})$.
4. $\forall\, t \in [T] \cup \{0\}, \forall c \in \Sigma_t : (\phi_t(G)_c \geq \phi_t(H)_c \iff \forall c' \in C_{\mathcal{F}}(c) : \phi_{\mathcal{F},t}(G)_{c'} \geq \phi_{\mathcal{F},t}(H)_{c'})$.

Proof. (1) $\iff$ (2): This equivalence follows from the formulas for the kernel obtained in Section 3.3. Indeed

$$k^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G, H) = k^{(T)}_{\mathsf{WLOA}}(G, H)$$
$$\Updownarrow$$
$$\sqrt{2Tn - 2k^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G, H)} = \sqrt{2Tn - 2k^{(T)}_{\mathsf{WLOA}}(G, H)}$$
$$\Updownarrow \text{ Section 3.3}$$
$$\left\| \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G) - \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(H) \right\| = \left\| \phi^{(T)}_{\mathsf{WLOA}}(G) - \phi^{(T)}_{\mathsf{WLOA}}(H) \right\|.$$

(2) $\iff$ (3): Indeed, we have

$$k^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G, H) = k^{(T)}_{\mathsf{WLOA}}(G, H)$$
$$\Updownarrow \text{ (by definition)}$$
$$\sum_{t \in [T] \cup \{0\}} \sum_{c \in \Sigma_t} \min(\phi_t(G)_c, \phi_t(H)_c) = \sum_{t \in [T] \cup \{0\}} \sum_{c \in \Sigma_t} \sum_{c' \in C_{\mathcal{F}}(c)} \min(\phi_{\mathcal{F},t}(G)'_c, \phi_{\mathcal{F},t}(H)'_c).$$

Furthermore, by the proof of Proposition 46,

$$\forall\, t \in [T] \cup \{0\}, c \in \Sigma_t : \min(\phi_t(G)_c, \phi_t(H)_c) \geq \sum_{c' \in C_{\mathcal{F}}(c)} \min(\phi_{\mathcal{F},t}(G)'_c, \phi_{\mathcal{F},t}(H)'_c).$$

Hence,

$$k^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G, H) = k^{(T)}_{\mathsf{WLOA}}(G, H)$$
$$\Updownarrow$$
$$\forall\, t \in [T] \cup \{0\}, \forall\, c \in \Sigma_t : \min(\phi_t(G)_c, \phi_t(H)_c) = \sum_{c' \in C_{\mathcal{F}}(c)} \min(\phi_{\mathcal{F},t}(G)_{c'}, \phi_{\mathcal{F},t}(H)_{c'}).$$

(4) $\implies$ (3): Assume (4) to be true, i.e.,

$$\forall\, t \in [T] \cup \{0\}, \forall c \in \Sigma_t : \phi_t(G)_c \geq \phi_t(H)_c \iff \forall\, c' \in C_{\mathcal{F}}(c) : \phi_{\mathcal{F},t}(G)_{c'} \geq \phi_{\mathcal{F},t}(H)_{c'}.$$

Consider any $t \in [T] \cup \{0\}$ and $c \in \Sigma_t$, and assume, without loss of generality,

$$\phi_t(G)_c \geq \phi_t(H)_c,$$

then by assumption

$$\forall\, c' \in C_{\mathcal{F}}(c) \colon \phi_{\mathcal{F},t}(G)_{c'} \geq \phi_{\mathcal{F},t}(H)_{c'}.$$

This implies

$$\min(\phi_t(G)_c, \phi_t(H)_c) = \phi_t(H)_c,$$

and

$$\forall\, c' \in C_{\mathcal{F}}(c) \colon \min(\phi_t(G)_{c'}, \phi_t(H)_{c'}) = \phi_t(H)_{c'},$$

and thus,

$$\begin{aligned}
\forall\, t \in [T] \cup \{0\}, \forall c \in \Sigma_t : \min(\phi_t(G)_c, \phi_t(H)_c) &= \phi_t(H)_c \\
&= \sum_{c' \in C_{\mathcal{F}}(c)} \phi_{\mathcal{F},t}(H)_{c'} \\
&= \sum_{c' \in C_{\mathcal{F}}(c)} \min(\phi_{\mathcal{F},t}(G)_{c'}, \phi_{\mathcal{F},t}(H)_{c'}).
\end{aligned}$$

(3) $\implies$ (4): Assume (4) to be false, i.e.,

$$\begin{aligned}
\exists\, t \in [T] \cup \{0\}, \exists\, c \in \Sigma_t \exists\, c' \in C_{\mathcal{F}}(c) \colon (\phi_t(G)_c \geq \phi_t(H)_c \wedge \phi_{\mathcal{F},t}(G)_{c'} < \phi_{\mathcal{F},t}(H)_{c'}) \\
\vee (\phi_t(G)_c \leq \phi_t(H)_c \wedge \phi_{\mathcal{F},t}(G)_{c'} > \phi_{\mathcal{F},t}(H)_{c'}).
\end{aligned}$$

Without loss of generality, assume $\phi_t(G)_c \geq \phi_t(H)_c \wedge \phi_{\mathcal{F},t}(G)_{c'} < \phi_{\mathcal{F},t}(H)_{c'}$. This implies

$$\min(\phi_t(G)_c, \phi_t(H)_c) = \phi_t(H)_c,$$

and

$$\begin{aligned}
&\sum_{c'' \in C_{\mathcal{F}}(c)} \min(\phi_{\mathcal{F},t}(G)_{c''}, \phi_{\mathcal{F},t}(H)_{c''}) \\
&= \min(\phi_{\mathcal{F},t}(G)_{c'}, \phi_{\mathcal{F},t}(H)_{c'}) + \sum_{c'' \in C_{\mathcal{F}}(c), c'' \neq c'} \min(\phi_{\mathcal{F},t}(G)_{c''}, \phi_{\mathcal{F},t}(H)_{c''}) \\
&= \phi_{\mathcal{F},t}(G)_{c'} + \sum_{c'' \in C_{\mathcal{F}}(c), c'' \neq c'} \min(\phi_{\mathcal{F},t}(G)_{c''}, \phi_{\mathcal{F},t}(H)_{c''}) \\
&\leq \phi_{\mathcal{F},t}(G)_{c'} + \min(\phi_t(G)_c - \phi_{\mathcal{F},t}(G)_{c'}, \phi_t(H)_c - \phi_{\mathcal{F},t}(H)_{c'}) \\
&= \phi_{\mathcal{F},t}(G)_{c'} + \phi_t(H)_c - \phi_{\mathcal{F},t}(H)_{c'} \\
&< \phi_t(H)_c = \min(\phi_t(G)_c, \phi_t(H)_c),
\end{aligned}$$

proving that (3) is false if (4) is false and by contraposition (3) $\implies$ (4). □

Corollary 49. For $n \geq 8$, there exists a graph sample $(G_1, y_1), \ldots, (G_s, y_s) \in \mathcal{G}_n \times \{0, 1\}$ and $\mathcal{F}$ such that

1. $\forall i, j \in [s] \colon y_i = y_j \implies \left\| \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G_i) - \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G_j) \right\| = \left\| \phi^{(T)}_{\mathsf{WLOA}}(G_i) - \phi^{(T)}_{\mathsf{WLOA}}(G_j) \right\|.$
2. $\forall i, j \in [s] \colon y_i \neq y_j \implies \left\| \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G_i) - \phi^{(T)}_{\mathsf{WLOA},\mathcal{F}}(G_j) \right\| > \left\| \phi^{(T)}_{\mathsf{WLOA}}(G_i) - \phi^{(T)}_{\mathsf{WLOA}}(G_j) \right\|.$

Proof. In this proof, we will construct two graphs, i.e., $s = 2$, such that the above conditions hold. Note that the above conditions are the conditions from Proposition 48 and Proposition 13; thus, we can and will use their equivalent notions.

The graph sample will be $(G_1, 1), (G_2, 0) \in \mathcal{G}_8 \times \{0, 1\}$. For both graphs, we start with the 8-cycle C_8 and add edges. For G_1 we add all skip-lengths of 2, so $G_1 \coloneqq ([8], \{(i, i+1 \bmod 8) \mid i \in [8]\} \cup \{(i, i+2 \bmod 8) \mid i \in [8]\})$. As for G_2, we add all skip-lengths of 3, so $G_2 \coloneqq ([8], \{(i, i+1 \bmod 8) \mid i \in [8]\} \cup \{(i, i+3 \bmod 8) \mid i \in [8]\})$. Note that G_1 contains triangles, while G_2 does not. Also, note that both graphs only have one orbit (verified by the permutation in cycle notation $(1, 2, 3, 4, 5, 6, 7, 8)$). Consider $\mathcal{F} \coloneqq \{C_3\}$ to contain a triangle.

Notice that the first condition is true since we are considering one graph of each class and thus $G_i = G_j$. As for the second condition, we will consider the equivalent notion that

$$\exists\, t \in [T] \cup \{0\}\, \exists\, c \in \Sigma_t \,:\, \neg(\phi_t(G)_c \geq \phi_t(H)_c \iff \forall\, c' \in C_{\mathcal{F}}(c) : \phi_{\mathcal{F},t}(G)_{c'} \geq \phi_{\mathcal{F},t}(H)_{c'}).$$

We must prove this statement to be true for our construction. Consider $t = 0$, c to be the color that all nodes in G_1 and G_2 are colored (not considering $\mathcal{F}$) and let c' be the color of being contained in a triangle (considering $\mathcal{F}$). Then by definition

$$\phi_t(G_1)_c \leq \phi_t(G_2)_c \wedge \phi_{\mathcal{F},t}(G_1)_{c'} > \phi_{\mathcal{F},t}(G_2)_{c'}.$$

The first inequality holds since they are in fact equal $n = \phi_t(G_1)_c \leq \phi_t(G_2)_c = n$ and the second holds, since all nodes in G_1 are contained in triangles and no nodes in G_2 are contained in triangles and thus $n = \phi_{\mathcal{F},t}(G_1)_{c'} > \phi_{\mathcal{F},t}(G_2)_{c'} = 0$ which proves the statement above is true.

This construction can be extended such that both classes contain more than one graph by considering n-order 4-regular graphs, where one class contains graphs where all nodes are contained in triangles, and the other class contains graphs where no node is contained in a triangle. By definition, the first condition will always hold since the graphs in each class cannot be distinguished by 1-WL or 1-WL$_{\mathcal{F}}$, i.e., the intra-class distances will be 0 regardless of considering $\mathcal{F}$. □

F. Large margins and stochastic gradient descent

We present the proofs of our main results from Section 4, i.e., Theorem 57 and Theorem 60. We will need some supporting lemmas, which we state and prove next. We note that the proof structure is close to the one in Ji & Telgarsky (2019).

F.0.1. Setup

Recall that we consider *linear* L-layer MPNNs following Section 2 with trainable weight matrices $\boldsymbol{W}^{(i)} \in \mathbb{R}^{d_i \times d_{i-1}}$. Moreover, in our *linear MPNN*, after L layers, the final node embeddings $\boldsymbol{X}^{(L)}$ are given by

$$\boldsymbol{X}^{(L)} \coloneqq \boldsymbol{W}^{(L)} \boldsymbol{W}^{(L-1)} \cdots \boldsymbol{W}^{(1)} \boldsymbol{X}^{(0)} \boldsymbol{A}'(G)^L,$$

where $\boldsymbol{A}'(G) \coloneqq \boldsymbol{A}(G) + \boldsymbol{I}_n$, $\boldsymbol{I}_n \in \mathbb{R}^{n \times n}$ is the n-dimensional identity matrix, and $\boldsymbol{X} = \boldsymbol{X}^{(0)}$ is the $d_0 \times n$ matrix whose columns correspond to vertices' initial features; $d = d_0$.

These node embeddings are then converted into predictions

$$\hat{y} \coloneqq \mathsf{READOUT}\big(\boldsymbol{X}^{(L)}\big) = \boldsymbol{X}^{(L)} \mathbf{1}_n = \boldsymbol{W}^{(L)} \boldsymbol{W}^{(L-1)} \cdots \boldsymbol{W}^{(1)} \boldsymbol{X}^{(0)} \boldsymbol{A}'(G)^L \mathbf{1}_n.$$

In our analysis, we will often need to reason about the singular values of the weight matrices. For $j = 1, 2, \ldots, L$, we let $\sigma_j(t)$ denote the largest singular value of $\boldsymbol{W}^{(j)}(t)$, and we let $\boldsymbol{u}(t)$ and $\boldsymbol{v}(t)$ denote the left-singular and right-singular vectors, respectively, corresponding to this singular value.

Recall that the training dataset is $\{(G_i, \boldsymbol{X}_i, y_i)\}_{i=1}^k$, where $\boldsymbol{X}_i \in \mathbb{R}^{d_i \times n_i}$ is a set of d_i-dimensional node features over an n_i-order graph G_i with $|V(G_i)| =: n_i$, and $y_i \in \{-1, +1\}$ for all i. Also, we write $d = d_0$ for the input node feature dimension. We further recall that the loss function ℓ satisfies the following assumptions.

Assumption 14. The loss function $\ell \colon \mathbb{R} \to \mathbb{R}^+$ has a continuous derivative ℓ' such that $\ell'(x) < 0$ for all x, $\lim_{x \to -\infty} \ell(x) = \infty$, and $\lim_{x \to \infty} \ell(x) = 0$.

The empirical risk induced by the MPNN is

$$\mathcal{R}(\boldsymbol{W}^{(L)}, \ldots, \boldsymbol{W}^{(1)}) \coloneqq \frac{1}{k} \sum_{i=1}^{k} \ell(y_i, \hat{y}_i)$$
$$= \frac{1}{k} \sum_{i=1}^{k} \ell(\boldsymbol{W}_{\mathrm{prod}} \boldsymbol{Z}_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i}),$$

where $\boldsymbol{W}_{\mathrm{prod}} = \boldsymbol{W}^{(L)} \boldsymbol{W}^{(L-1)} \cdots \boldsymbol{W}^{(1)}$, and $\boldsymbol{Z}_i = y_i \boldsymbol{X}_i$.

For convenience, it will often be useful to write $\mathcal{R}$ as a function of the product $\boldsymbol{W}_{\mathrm{prod}}$. Let $\mathcal{R}_1$ be the risk function $\mathcal{R}$ written as a function of the product $\boldsymbol{W}_{\mathrm{prod}}$, i.e.,

$$\mathcal{R}_1(\boldsymbol{W}_{\mathrm{prod}}) \coloneqq \frac{1}{k} \sum_{i=1}^{k} \ell(\boldsymbol{W}_{\mathrm{prod}} \boldsymbol{Z}_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i}).$$

We will consider *gradient flow*. In gradient flow, the evolution of $\boldsymbol{W} = (\boldsymbol{W}^{(L)}, \boldsymbol{W}^{(L-1)}, \ldots, \boldsymbol{W}^{(1)})$ is given by $\{\boldsymbol{W}(t) \colon t \geq 0\}$, where there is an initial state $\boldsymbol{W}(0)$ at $t = 0$, and

$$\frac{d\boldsymbol{W}(t)}{dt} \coloneqq -\nabla \mathcal{R}(\boldsymbol{W}(t)).$$

Note that gradient flow satisfies the following:

$$\frac{d\mathcal{R}(\boldsymbol{W}(t))}{dt} = \left\langle \nabla \mathcal{R}(\boldsymbol{W}(t)), \frac{d\boldsymbol{W}(t)}{dt} \right\rangle = -\|\nabla \mathcal{R}(\boldsymbol{W})\|_2^2 = -\sum_{j=1}^{L} \left\| \frac{\partial \mathcal{R}}{\partial \boldsymbol{W}^{(j)}} \right\|_F^2, \tag{7}$$

which implies that the risk never increases. The discrete version of this is given by

$$\boldsymbol{W}(t+1) \coloneqq \boldsymbol{W}(t) - \eta_t \nabla \mathcal{R}(\boldsymbol{W}(t)),$$

which corresponds to gradient descent with step size η_t. Recall that we make the following assumption on the initialization of the network under consideration:

Assumption 15. The initialization of $\boldsymbol{W}$ at $t = 0$ satisfies $\nabla \mathcal{R}(\boldsymbol{W}(0)) \neq \mathcal{R}(0) = \ell(0)$.

F.0.2. Lemmas and Theorems

The proof structure of our main theorems largely follows that of Ji & Telgarsky (2019), except with the main change that $x_i \mapsto X_i \boldsymbol{A}'(G)^L \mathbf{1}_n$ and $z_i \mapsto Z_i \boldsymbol{A}'(G)^L \mathbf{1}_n$. Many of the lemmas follow directly from the relevant lemma in Ji & Telgarsky (2019) with this transformation; we therefore defer to their proofs for a number of lemmas.

We start with a lemma that relates the weight matrices at successive levels to each other under the dynamics of gradient flow. This is essentially Theorem 1 of Arora et al. (2018) applied to our setting—our $\mathcal{R}_1$ and $\mathcal{R}$ correspond to L^1 and L^N, respectively, in the aforementioned work.

Lemma 50 (Theorem 1 in Arora et al. (2018))**.** $(\boldsymbol{W}^{(j+1)})^\intercal(t) \boldsymbol{W}^{(j+1)}(t) - \boldsymbol{W}^{(j)}(t)(\boldsymbol{W}^{(j)})^\intercal(t)$ is a constant function of t.

Proof. For each $j = 1, 2, \ldots, L$,

$$\frac{\partial \mathcal{R}}{\partial \boldsymbol{W}^{(j)}} = \prod_{i=j+1}^{L} (\boldsymbol{W}^{(i)})^\intercal \cdot \frac{d\mathcal{R}_1}{d\boldsymbol{W}_{\mathrm{prod}}} (\boldsymbol{W}^{(L)} \boldsymbol{W}^{(L-1)} \cdots \boldsymbol{W}^{(1)}) \cdot \prod_{i=1}^{j-1} (\boldsymbol{W}^{(i)})^\intercal.$$

Hence, $\dot{\boldsymbol{W}}^{(j)} = \frac{d\boldsymbol{W}}{dt}$ is given by

$$\dot{\boldsymbol{W}}^{(j)} = -\nabla \mathcal{R}(\boldsymbol{W}(t))$$
$$= -\eta \prod_{i=j+1}^{L} (\boldsymbol{W}^{(i)}(t))^\intercal \cdot \frac{d\mathcal{R}}{d\boldsymbol{W}} (\boldsymbol{W}^{(L)}(t) \boldsymbol{W}^{(L-1)}(t) \cdots \boldsymbol{W}^{(1)}(t)) \cdot \prod_{i=1}^{j-1} (\boldsymbol{W}^{(i)}(t))^\intercal.$$

Right multiplying the equation for j by $(\boldsymbol{W}^{(j)})^\intercal(t)$ and left multiplying the equation for $j+1$ by $(\boldsymbol{W}^{(j+1)})^\intercal(t)$, we see that

$$(\boldsymbol{W}^{(j+1)})^\intercal(t)\dot{\boldsymbol{W}}^{(j+1)}(t) = \dot{\boldsymbol{W}}^{(j)}(t)(\boldsymbol{W}^{(j)})^\intercal(t).$$

Adding the above equation to its transpose, we obtain

$$(\boldsymbol{W}^{(j+1)})^\intercal(t)\dot{\boldsymbol{W}}^{(j+1)}(t) + (\dot{\boldsymbol{W}}^{(j+1)})^\intercal(t)\boldsymbol{W}^{(j+1)}(t) = \dot{\boldsymbol{W}}^{(j)}(t)(\boldsymbol{W}^{(j)})^\intercal(t) + \boldsymbol{W}^{(j)}(t)(\dot{\boldsymbol{W}}^{(j)})^\intercal(t).$$

Note that this is equivalent to

$$\frac{d}{dt}\Big[(\boldsymbol{W}^{(j+1)})^\intercal(t)\boldsymbol{W}^{(j+1)}(t)\Big] = \frac{d}{dt}\Big[\boldsymbol{W}^{(j+1)}(t)(\boldsymbol{W}^{(j+1)})^\intercal(t)\Big],$$

which implies that $(\boldsymbol{W}^{(j+1)})^\intercal(t)\boldsymbol{W}^{(j+1)}(t) - \boldsymbol{W}^{(j+1)}(t)(\boldsymbol{W}^{(j+1)})^\intercal(t)$ does not depend on t, as desired. □

For the remainder of this section, let $B(R)$ denote the set of $\boldsymbol{W} = (\boldsymbol{W}^{(L)}, \boldsymbol{W}^{(L-1)}, \ldots, \boldsymbol{W}^{(1)})$ for which each component is bounded by R in Frobenius norm, i.e.,

$$B(R) = \left\{\boldsymbol{W} \colon \max_{1\le j\le L} \|\boldsymbol{W}^{(j)}\|_F \le R\right\}.$$

We now present the following lemma, which shows that the partial derivative of the risk function with respect to the first weight matrix $\boldsymbol{W}^{(1)}$ is bounded away from 0 in the Frobenius norm.

Lemma 51. For any $R > 0$, there exists a constant $\epsilon_R > 0$ such that for any $t \ge 1$ and $(\boldsymbol{W}^{(L)}(t), \boldsymbol{W}^{(L-1)}(t), \cdots, \boldsymbol{W}^{(1)}(t)) \in B(R)$, we have $\|\partial\mathcal{R}(t)/\partial\boldsymbol{W}^{(1)}(t)\|_F \ge \epsilon_R$.

Proof. The lemma is the same as the first part of Lemma 2.3 in (Ji & Telgarsky, 2019). Therefore, we defer to the proof there. □

Our main interest in Lemma 51 is that it allows us to prove the following important corollary, which establishes that under gradient flow, the weight matrices grow unboundedly in Frobenius norm and do not spend much time inside a ball of any fixed finite radius.

Corollary 52. Under gradient flow subject to Assumption 14 and Assumption 15, $\{t \ge 0 \colon \boldsymbol{W}(t) \in B(R)\}$ has finite measure.

Proof. The corollary corresponds to the second part of Ji & Telgarsky (2019). We reproduce the proof here. Note that since $d\mathcal{R}(\boldsymbol{W}(t))/dt = -\|\nabla\mathcal{R}(\boldsymbol{W}(t))\|_F^2 \le 0$ for all $t \ge 0$ (see Equation (7)),

$$\begin{aligned}
\mathcal{R}(\boldsymbol{W}(0)) &\ge -\int_0^\infty \frac{dR(\boldsymbol{W}(t))}{dt}\,dt \\
&= \int_0^\infty \left\|\frac{\partial\mathcal{R}(t)}{\partial\boldsymbol{W}(t)}\right\|_F^2 dt \\
&= \int_0^\infty \left(\sum_{j=1}^L \left\|\frac{\partial\mathcal{R}(t)}{\partial\boldsymbol{W}^{(j)}(t)}\right\|_F^2\right) dt \\
&\ge \int_0^\infty \left\|\frac{\partial\mathcal{R}(t)}{\partial\boldsymbol{W}^{(1)}(t)}\right\|_F^2 dt \\
&\ge \int_1^\infty \left\|\frac{\partial\mathcal{R}(t)}{\partial\boldsymbol{W}^{(1)}(t)}\right\|_F^2 dt \\
&\ge \epsilon(R)^2 \int_1^\infty \mathbb{I}[\boldsymbol{W}(t) \in B(R)]\,dt,
\end{aligned}$$

where the final implication holds due to Lemma 51. Since $\mathcal{R}(\boldsymbol{W}(0))$ is finite, this implies that $\{t \ge 0 \colon \boldsymbol{W}(t) \in B(R)\}$ has finite measure. □

We now define the following notation for convenience:

$$\boldsymbol{B}_j(t) \coloneqq \boldsymbol{W}^{(j)}(t)(\boldsymbol{W}^{(j)})^\intercal(t) - \boldsymbol{W}^{(j+1)}(t)(\boldsymbol{W}^{(j+1)})^\intercal(t), \text{ and}$$
$$D \coloneqq \left(\max_{1\le j\le L} \|\boldsymbol{W}^{(j)}(0)\|_F^2\right) - \|\boldsymbol{W}^{(L)}(0)\|_F^2 + \sum_{j=1}^{L-1} \|\boldsymbol{B}_j(0)\|_2^2.$$

While the previous corollary allows us to show the unboundedness of the weight matrices in the Frobenius norm, we often need to reason about the weight matrices in the standard operator norm. The following lemma shows that the two norms can not differ by too much.

Lemma 53. For every $1 \le i \le L$, we have $\|\boldsymbol{W}^{(i)}\|_F^2 - \|\boldsymbol{W}^{(i)}\|_2^2 \le D$.

Proof. A proof appears in (Ji & Telgarsky, 2019); see part 1 of Lemma 2.6. □

The next lemma is the key to establishing the "alignment" property. Roughly speaking, it establishes that the largest left singular vector of a weight matrix gets minimally aligned with the largest right singular vector of the weight matrix in the successive round of message passing.

Lemma 54. For all $1 \le j \le L$, we have

$$\langle \boldsymbol{v}_{j+1}, \boldsymbol{u}_j\rangle^2 \ge 1 - \frac{D + \|\boldsymbol{W}^{(j)}(0)\|_2^2 + \|\boldsymbol{W}^{(j+1)}(0)\|_2^2}{\sigma_{j+1}^2}.$$

Proof. Once again, the proof appears in (Ji & Telgarsky, 2019) (see part 2 of Lemma 2.6). □

The previous two lemmas can be used to establish the following lemma, which shows that each (normalized) weight matrix tends to a rank-1 approximation given by its top left and right singular vectors, and the (normalized) partial product of weight matrices tend to the relevant right singular vector of the final weight matrix in the product. We note that the first part of the lemma appears in Theorem 2.2 of (Ji & Telgarsky, 2019); however, the second part does not appear explicitly in their work (although the proof is similar to the third part of Lemma 2.6 in (Ji & Telgarsky, 2019)). Therefore, we provide the proof below.

Lemma 55. Suppose $\min_{1\le j\le L} \|\boldsymbol{W}^{(j)}(t)\|_F \to \infty$ as $t \to \infty$. For any $1 \le j \le L$, we have,

- $\boldsymbol{W}^{(j)}(t)/\|\boldsymbol{W}^{(j)}(t)\|_F \to \boldsymbol{u}_j(t)\boldsymbol{v}_j(t)^\intercal$ as $t \to \infty$.
- Also,
$$\left|\frac{\boldsymbol{W}^{(L)}(t)\boldsymbol{W}^{(L-1)}(t)\cdots\boldsymbol{W}^{(j)}(t)}{\|\boldsymbol{W}^{(L)}(t)\|_F\|\boldsymbol{W}^{(L-1)}(t)\|_F\cdots\|\boldsymbol{W}^{(j)}(t)\|_F}\boldsymbol{v}_j(t)\right| \to 1$$
as $t \to \infty$.

Proof. Since $\|\boldsymbol{W}^{(j)}(t)\|_F \to \infty$, Lemma 53 implies that, as $t \to \infty$, $\|\boldsymbol{W}^{(j)}(t)\|_2 \to \infty$, and, moreover, the singular values of $\boldsymbol{W}^{(j)}(t)$ beyond the top singular value are dominated by $\|\boldsymbol{W}^{(j)}(t)\|_F$. Thus, $\boldsymbol{W}^{(j)}(t)/\|\boldsymbol{W}^{(j)}(t)\|_F \to \boldsymbol{u}_j(t)\boldsymbol{v}_j(t)^\intercal$, which establishes the first part.

For the second part, note that by Lemma 54 and the fact that $\sigma_j = \|\boldsymbol{W}^{(j)}(t)\|_2 \to \infty$, we have that $|\langle \boldsymbol{u}_j(t), \boldsymbol{v}_{j+1}(t)\rangle| \to 1$. Hence, for any j, we have

$$\begin{aligned}\left|\frac{\boldsymbol{W}^{(L)}\boldsymbol{W}^{(L-1)}\cdots\boldsymbol{W}^{(j)}}{\|\boldsymbol{W}^{(L)}\|_F\|\boldsymbol{W}^{(L-1)}\|_F\cdots\|\boldsymbol{W}^{(j)}\|_F}\boldsymbol{v}_j\right| &\to \left|(\boldsymbol{u}_L\boldsymbol{v}_L^\intercal)\cdots(\boldsymbol{u}_j\boldsymbol{v}_j^\intercal)\boldsymbol{v}_j\right| \\ &= \left|\boldsymbol{u}_L(\boldsymbol{v}_L^\intercal\boldsymbol{u}_{L-1})\cdots(\boldsymbol{v}_{j+1}^\intercal\boldsymbol{u}_j)(\boldsymbol{v}_j^\intercal\boldsymbol{v}_j)\right| \\ &\to |\boldsymbol{u}_L| \\ &= 1\end{aligned}$$

as $t \to \infty$, which completes the proof. □

The following theorem shows that under gradient flow, the risk goes to zero as $t \to \infty$, while the Frobenius norm of each weight matrix tends to infinity. The theorem corresponds to parts 1 and 2 of Theorem 2.2 in Ji & Telgarsky (2019); therefore, we defer to the proofs there.

Theorem 56 (Parts 1 and 2 of Theorem 2.2 in (Ji & Telgarsky, 2019))**.** We have the following:

- $\lim_{t\to\infty} \mathcal{R}(\boldsymbol{W}(t)) = 0$.
- For all $i = 1, 2, \ldots, L$, we have $\lim_{t\to\infty} \|\boldsymbol{W}^{(i)}(t)\|_F = \infty$.

Proof. See the proof of parts 1 and 2 of Theorem 2.2 in (Ji & Telgarsky, 2019). □

Our main alignment result for linear MPNNs is the following, whose proof follows easily from the previous lemmas.

Theorem 57. Suppose Assumption 14 and Assumption 15 hold. Let $\boldsymbol{u}_i(t) \in \mathbb{R}^{d_i}$ and $\boldsymbol{v}_i(t) \in \mathbb{R}^{d_{i-1}}$ denote the left and right singular vectors, respectively, of $\boldsymbol{W}^{(i)}(t) \in \mathbb{R}^{d_i \times d_{i-1}}$. Then, we have the following using the Frobenius norm $\|\cdot\|_F$:

- For $j = 1, 2, \ldots, L$, we have
$$\lim_{t\to\infty} \left\| \frac{\boldsymbol{W}^{(j)}(t)}{\|\boldsymbol{W}^{(j)}(t)\|_F} - \boldsymbol{u}_j(t)\boldsymbol{v}_j(t)^\intercal \right\|_F = 0.$$
- Also,
$$\lim_{t\to\infty} \left| \left\langle \frac{(\boldsymbol{W}^{(L)}(t) \cdots \boldsymbol{W}^{(1)}(t))^\intercal}{\prod_{j=1}^{L} \|\boldsymbol{W}^{(j)}(t)\|_F}, \boldsymbol{v}_1 \right\rangle \right| = 1.$$

Proof. Note that by Theorem 56, we have that $\|\boldsymbol{W}^{(j)}\|_F \to \infty$ for every j. Thus, the first part of Lemma 55 implies the first part of the theorem. Note that setting $j = 1$ in the second part of Lemma 55 implies the second part of the theorem, completing the proof. □

F.0.3. MARGIN

We now state results on the margin.

Lemma 58. Suppose the data set $\{(\boldsymbol{X}_i, y_i)\}_{i=1}^k$ and G_i on n_i nodes are sampled according to Assumption 17. Let $S \subset \{1, 2, \ldots, k\}$ be the set of indices for support vectors. Then,

$$\min_{\substack{\|\boldsymbol{\xi}\|_2 = 1 \\ \langle \boldsymbol{\xi}, \bar{\boldsymbol{u}} \rangle = 0}} \max_{i \in S} \left\langle \boldsymbol{\xi}, Z_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i} \right\rangle > 0 \tag{8}$$

with probability 1 over the sampling.

Proof. First, we note that there are $s \le d$ support vectors; furthermore, each support vector $Z_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i}$ has a corresponding dual variable α_i that is *positive*, so that

$$\sum_{i \in S} \alpha_i Z_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i} = \bar{\boldsymbol{u}}. \tag{9}$$

This follows from Soudry et al. (2018) (see Lemma 12 in Appendix B), which was also used by Ji & Telgarsky (2019)).

Next, assume for the sake of contradiction that there exists $\boldsymbol{\xi}$ with $\|\boldsymbol{\xi}\|_2 = 1$ and $\langle \boldsymbol{\xi}, \bar{\boldsymbol{u}} \rangle = 0$ but

$$\max_{1 \le i \le k} \left\langle \boldsymbol{\xi}, Z_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i} \right\rangle \le 0.$$

Then, note that

$$\begin{aligned}
0 &= \langle \boldsymbol{\xi}, \bar{\boldsymbol{u}} \rangle \\
&= \left\langle \boldsymbol{\xi}, \sum_{i \in S} \alpha_i Z_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i} \right\rangle \\
&= \sum_{i \in S} \alpha_i \big\langle \boldsymbol{\xi}, Z_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i} \big\rangle \\
&\leq 0.
\end{aligned}$$

This implies that $\big\langle \boldsymbol{\xi}, Z_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i} \big\rangle = 0$ for all $i \in S$, which contradicts our assumption that the support vectors span the entirety of $\mathbb{R}^d$. This completes the proof. □

Lemma 59. Suppose Assumption 17 holds. Let ℓ be the exponential loss given by $\ell(x) = e^{-x}$. For almost all data, if $\boldsymbol{w} \in \mathbb{R}^d$ satisfies $\langle \boldsymbol{w}, \boldsymbol{u} \rangle \geq 0$ and $\boldsymbol{w}^\perp$, the projection of $\boldsymbol{w}$ on to the subspace of $\mathbb{R}^d$ orthogonal to $\boldsymbol{u}$, satisfies $\|\boldsymbol{w}^\perp\|_2 \geq \frac{1+\ln(k)}{\alpha}$, then $\langle \boldsymbol{w}^\perp, \nabla \mathcal{R}(\boldsymbol{w}) \rangle \geq 0$ (recall α from Equation (9)).

Proof. Let $\boldsymbol{v}_j = \boldsymbol{Z}_j \boldsymbol{A}'(G_j)^L \mathbf{1} = y_j \boldsymbol{X}_j \boldsymbol{A}'(G_j)^L$. Moreover, for any $\boldsymbol{z} \in \mathbb{R}^d$ let $\boldsymbol{z} = \boldsymbol{z}^\parallel + \boldsymbol{z}^\perp$, where $\boldsymbol{z}^\parallel$ is the projection of $\boldsymbol{z}$ on to $\boldsymbol{u}$ and $\boldsymbol{z}^\perp$ is the component of $\boldsymbol{z}$ orthogonal to $\boldsymbol{u}$. Let $j' = \arg\max_{j \in S} \langle -\boldsymbol{w}^\perp, \boldsymbol{v}_j \rangle$ (recall that S is the index set for support vectors).. We note that $-\langle \boldsymbol{w}^\perp, \boldsymbol{v}_{j'}^\perp \rangle = -\langle \boldsymbol{w}^\perp, \boldsymbol{v}_{j'} \rangle \geq \alpha \|\boldsymbol{w}^\perp\|$, where α is the quantity on the lefthand side of (8).

Observe that

$$\begin{aligned}
\langle \boldsymbol{w}^\perp, \nabla \mathcal{R}(\boldsymbol{w}^\intercal) \rangle &= \frac{1}{k} \sum_{i=1}^{k} \ell'(\langle \boldsymbol{w}, \boldsymbol{v}_i \rangle) \cdot \langle \boldsymbol{w}^\perp, \boldsymbol{v}_i \rangle \\
&= -\frac{1}{k} \sum_{i=1}^{k} \exp(-\langle \boldsymbol{w}, \boldsymbol{v}_i \rangle) \cdot \langle \boldsymbol{w}^\perp, \boldsymbol{v}_i^\perp \rangle \\
&= -\frac{1}{k} \exp(-\langle \boldsymbol{w}, \boldsymbol{v}_{j'} \rangle) \cdot \langle \boldsymbol{w}^\perp, \boldsymbol{v}_{j'}^\perp \rangle - \frac{1}{k} \sum_{\substack{1 \leq i \leq k \\ \langle \boldsymbol{w}^\perp, \boldsymbol{v}_i^\perp \rangle \geq 0}} \exp(-\langle \boldsymbol{w}, \boldsymbol{v}_i \rangle) \cdot \langle \boldsymbol{w}^\perp, \boldsymbol{v}_i^\perp \rangle. \qquad (10)
\end{aligned}$$

The first term on the righthand side of (10) can be bounded as follows:

$$\begin{aligned}
-\frac{1}{k} \exp(-\langle \boldsymbol{w}, \boldsymbol{v}_{j'} \rangle) \cdot \langle \boldsymbol{w}^\perp, \boldsymbol{v}_{j'}^\perp \rangle &= -\frac{1}{k} \exp\Big(-\langle \boldsymbol{w}, \boldsymbol{v}_{j'}^\parallel \rangle - \langle \boldsymbol{w}, \boldsymbol{v}_{j'}^\perp \rangle\Big) \cdot \langle \boldsymbol{w}^\perp, \boldsymbol{v}_{j'}^\perp \rangle \\
&= -\frac{1}{k} \exp\Big(-\langle \boldsymbol{w}^\parallel, \boldsymbol{v}_{j'}^\parallel \rangle\Big) \exp\big(-\langle \boldsymbol{w}^\perp, \boldsymbol{v}_{j'}^\perp \rangle\big) \cdot \langle \boldsymbol{w}^\perp, \boldsymbol{v}_{j'}^\perp \rangle \\
&\geq \frac{1}{k} \exp(-\langle \boldsymbol{w}, \gamma \boldsymbol{u} \rangle) \exp(\alpha \|\boldsymbol{w}^\perp\|) \cdot \alpha \|\boldsymbol{w}^\perp\|. \qquad (11)
\end{aligned}$$

For the second term in (10), we have

$$\begin{aligned}
\sum_{\substack{1 \leq i \leq k \\ \langle \boldsymbol{w}^\perp, \boldsymbol{v}_i^\perp \rangle \geq 0}} -\frac{1}{k} \exp(-\langle \boldsymbol{w}, \boldsymbol{v}_i \rangle) \cdot \langle \boldsymbol{w}^\perp, \boldsymbol{v}_i^\perp \rangle &= \sum_{\substack{1 \leq i \leq k \\ \langle \boldsymbol{w}^\perp, \boldsymbol{v}_i^\perp \rangle \geq 0}} -\frac{1}{k} \exp(-\langle \boldsymbol{w}, \gamma \bar{\boldsymbol{u}} \rangle) \exp(-\langle \boldsymbol{w}, \boldsymbol{v}_i - \gamma \bar{\boldsymbol{u}} \rangle) \cdot \langle \boldsymbol{w}^\perp, \boldsymbol{v}_i^\perp \rangle \\
&\geq \sum_{\substack{1 \leq i \leq k \\ \langle \boldsymbol{w}^\perp, \boldsymbol{v}_i^\perp \rangle \geq 0}} -\frac{1}{k} \exp(-\langle \boldsymbol{w}, \gamma \bar{\boldsymbol{u}} \rangle) \exp(-\langle \boldsymbol{w}^\perp, \boldsymbol{v}_i^\perp \rangle) \cdot \langle \boldsymbol{w}^\perp, \boldsymbol{v}_i^\perp \rangle \\
&\geq \sum_{\substack{1 \leq i \leq k \\ \langle \boldsymbol{w}^\perp, \boldsymbol{v}_i^\perp \rangle \geq 0}} \frac{1}{k} \exp(-\langle \boldsymbol{w}, \gamma \bar{\boldsymbol{u}} \rangle)(-e^{-1}) \\
&\geq \exp(-\langle \boldsymbol{w}, \gamma \bar{\boldsymbol{u}} \rangle)(-e^{-1}), \qquad (12)
\end{aligned}$$

since $xe^{-x} \leq -e^{-1}$ for $x \geq 0$, and the assumption $\langle \boldsymbol{w}, \boldsymbol{u} \rangle \geq 0$ along with the fact that $\boldsymbol{v}_i$ has margin at least γ implies that $\langle \boldsymbol{w}, \boldsymbol{v}_i - \gamma \boldsymbol{u} - \boldsymbol{v}_i^{\perp} \rangle \geq 0$.

By plugging (11) and (12) into (10), we obtain

$$\langle \boldsymbol{w}^{\perp}, \nabla \mathcal{R}(\boldsymbol{w}^{\intercal}) \rangle \geq \exp(-\langle \boldsymbol{w}, \gamma \bar{\boldsymbol{u}} \rangle) \left[\frac{1}{k} \exp(\alpha \|\boldsymbol{w}^{\perp}\|) \cdot \alpha \|\boldsymbol{w}^{\perp}\| - e^{-1} \right].$$

Finally, note that since $\|\boldsymbol{w}^{\perp}\| \geq (1 + \ln(k))/\alpha$ (by the assumption in the lemma), $\frac{1}{k} \exp(\alpha \|\boldsymbol{w}^{\perp}\|) \cdot \alpha \|\boldsymbol{w}^{\perp}\| - e^{-1} \geq 0$, which completes the proof. □

Our main theorem establishes the convergence of linear MPNNs to the maximum margin solution.

Theorem 60 (Convergence to the maximum margin solution)**.** Suppose Assumption 14 and Assumption 17 hold. Then, for the exponential loss function $\ell(x) = e^{-x}$, under gradient flow, we have that the learned weights of the MPNN converge to the maximum margin solution, i.e.,

$$\lim_{t \to \infty} \frac{\boldsymbol{W}^{(L)}(t) \boldsymbol{W}^{(L-1)}(t) \cdots \boldsymbol{W}^{(1)}(t)}{\|\boldsymbol{W}^{(L)}(t)\|_F \|\boldsymbol{W}^{(L-1)}(t)\|_F \cdots \|\boldsymbol{W}^{(1)}(t)\|_F} = \bar{\boldsymbol{u}}.$$

Proof. The proof follows that of Theorem 2.8 in (Ji & Telgarsky, 2019), except that one uses Assumption 17 along with the transformations $x_i \mapsto \boldsymbol{X}_i \boldsymbol{A}'(G)^L \mathbf{1}_n$ and $z_i \mapsto \boldsymbol{Z}_i \boldsymbol{A}'(G)^L \mathbf{1}_n$, where the relevant support vectors are of the form $\boldsymbol{Z}_i \boldsymbol{A}'(G)^L \mathbf{1}_n$. The proof follows similarly from Lemma 59 as in (Ji & Telgarsky, 2019). □

G. Additional experimental details, data, and results

Here, we outline additional details on the experiments and state more experimental results.

G.1. Datasets

We used the well-known graph classification benchmark datasets from Morris et al. (2020a); see Table 2 for dataset statistics and properties.[2] Specifically, we used the ENZYMES (Schomburg et al., 2004; Borgwardt & Kriegel, 2005), MUTAG (Debnath et al., 1991; Kriege & Mutzel, 2012), PROTEINS (Dobson & Doig, 2003; Borgwardt & Kriegel, 2005), PTC_FM, and PTC_MR (Helma et al., 2001) datasets. *To concentrate purely on the graph structure, we omitted potential vertex and edge labels.* Moreover, we created two sets of synthetic datasets. First, we created synthetic datasets to verify Proposition 9. We followed the construction outlined in the proof of Proposition 9 to create 1 000 graphs on 16, 32, 64, and 128 vertices. Secondly, we created 1 000 Erdős–Rényi graphs with 20 vertices each using edge probabilities 0.05, 0.1, 0.2, and 0.3, respectively. Here, we set a graph's class to the number of subgraphs isomorphic to either C_3, C_4, C_5, or K_4, resulting in sixteen different datasets.

G.2. Graph kernel and MPNNs architectures

We implemented the (normalized) 1-WL, 1-WLOA, 1-WL$_{\mathcal{F}}$, and the 1-WLOA$_{\mathcal{F}}$ in Python. For the MPNN experiments, we used the GIN layer (Xu et al., 2019) using reLU activation functions and fixed the feature dimension to 64. We used mean pooling and a two-layer MLP using a dropout of 0.5 after the first layer for all experiments for the final classification. For the MPNN$_{\mathcal{F}}$ architectures, we encoded the initial label function $l_{\mathcal{F}}$ as a one-hot encoding.

G.3. Experimental protocol and model configuration

For the graph kernel experiments, for the 1-WLOA variants, we computed the (cosine) normalized Gram matrix for each kernel and computed the classification accuracies using the C-SVM implementation of LIBSVM (Chang & Lin, 2011). Here, a large C enforces linear separability with a large margin. We computed the ℓ_2 normalized feature vectors for the other kernels and computed the classification accuracies using the linear SVM implementation of LIBLINEAR (Fan et al., 2008). In both cases, we used 10-fold cross-validation. We repeated each 10-fold cross-validation ten times with different random folds and report average training and testing accuracies and standard deviations. We additionally report the margin on the

[2] All datasets are publicly available at `www.graphlearning.io`.

Table 2: Dataset statistics and properties.

Dataset	Properties			
	Number of graphs	Number of classes/targets	∅ Number of nodes	∅ Number of edges
ENZYMES	600	6	32.6	62.1
MUTAG	188	2	17.9	19.8
PROTEINS	1 113	2	39.1	72.8
PTC_FM	349	2	14.1	14.5
PTC_MR	344	2	14.3	14.7

Table 3: Mean train, test accuracies, and margins of kernel architectures on ER graphs for different levels of sparsity and different subgraphs. NLS—Not linearly separable. NB—Only one class in the dataset.

Subgraph	Algorithm	Edge probability			
		0.05	0.1	0.2	0.3
C_3	1-WL	94.2 ± 1.7 88.6 ± 0.7 NLS	96.3 ± 4.6 42.12 ± 1.7 0.013 ± 0.001	44.1 ± 12.4 11.2 $\pm$ NLS	38.4 ± 5.0 6.99 ± 1.1 NLS
	1-WLOA	100.0 ± 0.0 87.9 ± 0.1 DNC	100.0 ± 0.0 44.6 ± 1.1 DNC	100.0 ± 0.0 11.5 ± 0.9 DNC	100.0 ± 0.0 5.2 ± 0.3 DNC
	1-WL$_{\mathcal{F}}$	100.0 ± 0.0 **100.0** ± 0.0 0.037 ± 0.0	100.0 ± 0.0 **99.7** ± 0.1 0.009 < 0.001	100.0 ± 0.0 **100.0** ± 0.4 0.002 < 0.001	99.1 ± 0.2 **64.7** ± 0.8 0.001 < 0.001
	1-WLOA$_{\mathcal{F}}$	100.0 ± 0.0 98.6 ± 0.0 DNC	100.0 ± 0.0 93.8 ± 0.3 DNC	100.0 ± 0.0 42.0 ± 1.0 DNC	100.0 ± 0.0 7.4 ± 0.2 DNC
C_4	1-WL	95.1 ± 1.0 92.3 ± 0.2 NLS	89.7 ± 3.9 46.8 ± 0.8 NLS	48.7 ± 10.1 5.4 ± 0.5 NLS	46.6 ± 10.8 2.2 ± 0.4 NLS
	1-WLOA	100.0 ± 0.0 92.6 ± 0.0 DNC	100.0 ± 0.0 49.6 ± 0.8 DNC	100.0 ± 0.0 5.13 ± 0.6 DNC	100.0 ± 0.0 1.7 ± 0.3 DNC
	1-WL$_{\mathcal{F}}$	100.0 ± 0.0 **99.9** ± 0.1 0.037 ± 0.001	100.0 ± 0.0 **98.2** ± 0.2 0.009 < 0.001	100.0 0.0 **78.9** ± 0.6 0.002 < 0.001	100.0 ± 0.0 **7.3** ± 0.4 0.002 < 0.001
	1-WLOA$_{\mathcal{F}}$	100.0 ± 0.0 99.3 < 0.1 DNC	100.0 ± 0.0 93.7 ± 0.2 DNC	100.0 $\pm$ 22.4 ± 0.9 DNC	100.0 ± 0.0 2.8 ± 0.6 DNC
C_5	1-WL	97.2 ± 0.5 95.8 ± 0.3 NLS	69.3 ± 6.6 53.5 ± 0.6 NLS	65.1 ± 14.9 4.3 ± 0.7 NLS	64.8 ± 9.9 1.26 ± 0.2 NLS
	1-WLOA	100.0 ± 0.0 95.9 ± 0.0 DNC	100.0 ± 0.0 54.2 ± 0.3 DNC	100.0 ± 0.0 4.7 ± 0.5 DNC	100.0 ± 0.0 1.4 ± 0.5 DNC
	1-WL$_{\mathcal{F}}$	100.0 ± 0.0 **99.9** ± 0.0 0.058 ± 0.001	100.0 ± 0.0 **98.4** ± 0.2 0.012 < 0.001	100.0 ± 0.0 **68.8** ± 0.5 0.002 < 0.001	100.0 ± 0.0 **4.2** ± 0.5 0.003 < 0.001
	1-WLOA$_{\mathcal{F}}$	100.0 ± 0.0 99.5 ± 0.0 DNC	100.0 ± 0.0 91.8 ± 0.4 DNC	100.0 ± 0.0 20.0 ± 0.0 DNC	100.0 ± 0.0 2.6 ± 0.2 DNC
K_4	1-WL	NB	99.4 < 0.1 99.4 ± 0.0 NLS	78.0 ± 0.6 77.7 ± 0.3 NLS	68.7 ± 9.9 17.1 ± 0.9 NLS
	1-WLOA	NB	100.0 ± 0.0 99.4 ± 0.0 DNC	100.0 ± 0.0 77.8 ± 0.3 DNC	100.0 ± 0.0 20.6 ± 0.9 DNC
	1-WL$_{\mathcal{F}}$	NB	100.0 ± 0.0 **100.0** ± 0.0 0.122 ± 0.0	100.0 ± 0.0 **99.9** ± 0.1 0.022 < 0.001	100.0 ± 0.0 **94.8** ± 0.4 0.004 < 0.001
	1-WLOA$_{\mathcal{F}}$	NB	100.0 ± 0.0 99.4 ± 0.0 DNC	100.0 ± 0.0 97.8 ± 0.1 DNC	100.0 ± 0.0 74.6 ± 0.6 DNC

training splits for the linear SVM experiments. For the kernelized SVM, this was not possible. For the experiments on the TUDATSETS, following the evaluation method proposed in Morris et al. (2020a), the C-parameter and numbers of iterations were selected from $\{10^{-3}, 10^{-2}, \ldots, 10^2, 10^3\}$ and $\{1, \ldots, 5\}$, respectively, using a validation set sampled uniformly at random from the training fold (using 10 % of the training fold). For 1-WL$_{\mathcal{F}}$ and MPNN$_{\mathcal{F}}$, we used cycles and complete graphs on three to six vertices, respectively.

For the synthetic datasets, we set the C-parameter to 10^{10} to enforce linear separability and choose the number of iterations as with the TUDATSETS. All kernel experiments were conducted on a workstation with 512 GB of RAM using a single CPU core.

For the MPNN experiments, we also followed the evaluation method proposed in Morris et al. (2020a), choosing the number of layers from $\{1, \ldots, 5\}$, using a validation set sampled uniformly at random from the training fold (using 10 % of the training fold). We used an initial learning rate of 0.01 across all experiments with an exponential learning rate decay with patience of 5, a batch size of 128, and set the maximum number of epochs to 200. All MPNN experiments were conducted on a workstation with 512 GB of RAM using a single core and one NVIDIA Tesla A100s with 80 GB of GPU memory.

G.4. Results and discussion

In the following we answer questions **Q1** to **Q4**.

Q1 (*"Does adding expressive power make datasets more linearly separable?"*) See Tables 1, 3 and 6. Table 1 confirms Proposition 9, i.e., the 1-WL and 1-WLOA kernels do not achieve accuracies better than random and cannot linearly separate the training data. The 1-WL$_{\mathcal{F}}$ and 1-WLOA$_{\mathcal{F}}$ kernel linearly separate the data while achieving perfect test accuracies. In addition, Table 3 also confirms this for the ER graphs, i.e., for all datasets, the 1-WL and 1-WLOA kernels cannot separate the training data while the 1-WL$_{\mathcal{F}}$ and 1-WLOA$_{\mathcal{F}}$ can. Moreover, the subgraph-based kernels achieve the overall best predictive performance over all datasets, e.g., on the dataset using edge probability 0.2 and $\mathcal{F} = \{C_5\}$ the test accuracies of the 1-WL$_{\mathcal{F}}$ improves over the 1-WL by more than 57 %. Similar effects can be observed for the MPNN architectures; see Table 6.

Table 4: Mean train and test accuracies of the MPNN architectures on TUDATASETS datasets for different subgraphs.

$\mathcal{F}$	Algorithm	Dataset				
		ENZYMES	MUTAG	PROTEINS	PTC_FM	PTC_MR
—	MPNN	25.5 ± 1.9 20.9 ± 1.5	84.5 ± 1.2 82.4 ± 1.9	68.4 ± 0.8 67.1 ± 0.8	61.4 ± 1.8 59.2 ± 2.4	56.7 ± 2.3 53.7 ± 3.0
C_3	MPNN$_{\mathcal{F}}$	61.3 ± 4.1 29.2 ± 1.5	84.2 ± 1.9 81.8 ± 2.6	72.7 ± 0.8 67.8 ± 0.8	61.9 ± 2.4 59.7 ± 2.4	57.3 ± 0.9 **56.3** ± 2.5
C_3-C_4	MPNN$_{\mathcal{F}}$	61.8 ± 3.0 28.9 ± 1.5	84.5 ± 2.4 82.1 ± 2.6	72.5 ± 0.8 **69.0** ± 0.9	62.0 ± 0.8 59.4 ± 1.9	57.5 ± 0.8 54.3 ± 1.3
C_3-C_5	MPNN$_{\mathcal{F}}$	61.6 ± 2.8 28.5 ± 2.2	83.9 ± 2.2 81.8 ± 2.4	72.6 ± 0.8 68.6 ± 0.9	62.0 ± 2.0 58.8 ± 2.5	56.7 ± 1.4 54.9 ± 1.8
C_3-C_6	MPNN$_{\mathcal{F}}$	61.8 ± 6.0 28.8 ± 1.6	83.4 ± 2.4 82.2 ± 2.6	72.7 ± 0.6 68.4 ± 0.5	62.4 ± 0.9 **60.5** ± 2.4	57.2 ± 1.3 54.0 ± 2.0
K_3	MPNN$_{\mathcal{F}}$	60.1 ± 4.5 28.8 ± 1.0	84.8 ± 1.6 84.2 ± 1.7	72.6 ± 0.6 67.7 ± 0.7	62.1 ± 1.6 59.0 ± 2.3	57.2 ± 0.8 54.3 ± 1.8
K_3-K_4	MPNN$_{\mathcal{F}}$	61.8 ± 5.5 **29.5** ± 1.6	83.2 ± 2.6 80.9 ± 2.4	73.2 ± 0.8 68.1 ± 0.6	62.3 ± 2.0 59.1 ± 3.2	57.0 ± 1.4 53.7 ± 2.6
K_3-K_5	MPNN$_{\mathcal{F}}$	60.6 ± 3.9 29.2 ± 1.3	84.9 ± 1.4 **82.9** ± 2.1	72.9 ± 0.9 68.0 ± 1.1	61.7 ± 2.1 57.8 ± 2.8	58.6 ± 1.3 54.4 ± 1.8
K_3-K_6	MPNN$_{\mathcal{F}}$	60.3 ± 4.7 28.8 ± 1.4	84.6 ± 1.3 82.4 ± 2.5	73.4 ± 0.7 68.5 ± 0.9	62.2 ± 1.3 59.3 ± 1.6	57.7 ± 1.0 54.4 ± 2.8

Table 5: Mean train, test accuracies, and margins of the kernel architectures on TUDATASETS datasets for different subgraphs. DNC—Did not compute due to implicit kernel.

$\mathcal{F}$	Algorithm	Dataset				
		ENZYMES	MUTAG	PROTEINS	PTC_FM	PTC_MR
—	1-WL	90.4 ± 4.4 34.1 ± 1.7 0.023 ± 0.002	88.9 ± 1.4 83.7 ± 2.1 0.073 ± 0.044	79.9 ± 1.8 68.1 ± 1.1 0.090 ± 0.0189	70.1 ± 3.4 55.7 ± 2.7 0.196 ± 0.104	67.0 ± 2.2 54.2 ± 2.2 0.296 ± 0.194
—	1-WLOA	100.0 0.0 ± 0.0 32.3 ± 1.7 DNC	99.3 ± 1.4 82.6 ± 2.0 DNC	96.6 ± 2.7 **73.9** ± 0.7 DNC	91.7 ± 4.2 58.2 ± 1.5 DNC	95.5 ± 4.4 55.7 ± 1.5 DNC
C_3	1-WL$_{\mathcal{F}}$	97.0 ± 1.8 37.9 ± 1.8 0.021 ± 0.002	88.1 ± 2.4 83.3 ± 2.0 0.134 ± 0.082	89.1 ± 2.3 65.3 ± 1.1 0.078 ± 0.090	72.1 ± 3.1 57.0 ± 2.2 0.167 ± 0.070	70.9 ± 3.6 54.7 ± 2.3 0.143 ± 0.044
C_3-C_4	1-WL$_{\mathcal{F}}$	97.5 ± 1.1 **40.6** ± 1.7 0.020 ± 0.001	88.6 ± 1.2 **84.7** ± 1.9 0.065 ± 0.044	90.5 ± 2.1 65.2 ± 1.3 0.056 ± 0.014	72.5 ± 4.2 56.3 ± 1.4 0.188 ± 0.113	70.1 ± 4.9 55.5 ± 3.2 0.121 ± 0.037
C_3-C_5	1-WL$_{\mathcal{F}}$	97.1 ± 1.1 38.0 ± 1.5 0.022 ± 0.001	89.9 ± 1.8 83.0 ± 2.1 0.072 ± 0.066	91.6 ± 2.1 63.6 ± 1.1 0.052 ± 0.018	71.4 ± 3.5 56.6 ± 1.2 0.229 ± 0.187	68.8 ± 2.9 55.5 ± 1.7 0.138 ± 0.067
C_3-C_6	1-WL$_{\mathcal{F}}$	96.5 ± 1.5 38.7 ± 1.4 0.021 ± 0.001	92.2 ± 1.4 83.5 ± 2.2 0.090 ± 0.039	92.1 ± 2.4 64.9 ± 0.9 0.050 ± 0.019	74.8 ± 2.7 57.2 ± 2.8 0.193 ± 0.167	73.2 ± 4.2 56.5 ± 1.9 0.171 ± 0.123
K_3	1-WL$_{\mathcal{F}}$	96.4 ± 2.5 37.6 ± 0.9 0.021 ± 0.001	89.5 ± 1.8 84.0 ± 1.8 0.086 ± 0.064	87.2 ± 2.4 64.9 ± 1.6 0.059 ± 0.021	71.7 ± 4.1 57.0 ± 2.1 0.150 ± 0.081	67.8 ± 3.2 54.7 ± 3.1 0.162 ± 0.068
K_3-K_4	1-WL$_{\mathcal{F}}$	96.8 ± 3.0 36.8 ± 1.4 0.020 ± 0.002	88.3 ± 2.0 **84.7** ± 1.8 0.100 ± 0.064	88.9 ± 2.6 64.7 ± 1.2 0.062 ± 0.018	70.6 ± 3.1 56.0 ± 2.3 0.151 ± 0.065	68.6 ± 4.2 55.6 ± 2.2 0.135 ± 0.049
K_3-K_5	1-WL$_{\mathcal{F}}$	96.5 ± 2.2 36.6 ± 1.8 0.021 ± 0.001	88.2 ± 2.2 82.7 ± 2.6 0.098 ± 0.072	89.7 ± 2.8 64.5 ± 0.9 0.047 ± 0.012	72.9 ± 5.7 57.2 ± 1.2 0.182 ± 0.077	67.6 ± 4.4 54.3 ± 1.3 0.145 ± 0.055
K_3-K_6	1-WL$_{\mathcal{F}}$	95.8 ± 2.0 37.7 ± 1.5 0.021 ± 0.001	88.7 ± 1.8 84.3 ± 1.1 0.078 ± 0.049	88.9 ± 3.0 63.6 ± 1.4 0.055 ± 0.022	71.0 ± 2.7 56.0 ± 2.4 0.147 ± 0.047	69.6 ± 3.7 55.0 ± 2.1 0.133 ± 0.038
C_3	1-WLOA$_{\mathcal{F}}$	100.0 ± 0.0 36.7 ± 1.8 DNC	99.3 ± 1.4 83.4 ± 2.7 DNC	100.0 ± 0.0 67.6 ± 0.9 DNC	91.7 ± 4.2 **59.6** ± 0.5 DNC	93.7 ± 5.7 56.1 ± 1.4 DNC
C_3-C_4	1-WLOA$_{\mathcal{F}}$	100.0 ± 0.0 36.1 ± 1.8 DNC	99.6 ± 1.1 84.3 ± 1.9 DNC	99.6 ± 1.1 66.5 ± 0.4 DNC	88.0 ± 4.7 59.5 ± 1.2 DNC	94.6 ± 3.4 55.5 ± 1.7 DNC
C_3-C_5	1-WLOA$_{\mathcal{F}}$	100.0 ± 0.0 35.0 ± 1.7 DNC	99.7 ± 1.0 82.2 ± 1.3 DNC	100.0 ± 0.0 65.9 ± 0.5 DNC	89.7 ± 4.6 58.6 ± 1.2 DNC	92.8 4.7 ± 4.7 55.4 ± 1.0 DNC
C_3-C_6	1-WLOA$_{\mathcal{F}}$	100.0 ± 0.0 35.8 ± 1.8 DNC	98.3 ± 3.2 83.1 ± 2.6 DNC	99.6 ± 1.2 66.2 ± 0.6 DNC	91.7 ± 5.3 58.7 ± 2.6 DNC	89.2 ± 5.4 55.9 ± 1.5 DNC
K_3	1-WLOA$_{\mathcal{F}}$	100.0 ± 0.0 37.0 ± 1.6 DNC	100.0 ± 0.0 83.2 ± 1.7 DNC	99.2 ± 1.5 67.2 ± 0.5 DNC	92.5 ± 5.9 59.0 ± 1.7 DNC	92.7 ± 3.6 57.0 ± 1.8 DNC
K_3-K_4	1-WLOA$_{\mathcal{F}}$	100.0 ± 0.0 37.7 ± 1.4 DNC	99.3 ± 1.5 82.1 ± 1.6 DNC	100.0 ± 0.0 66.7 ± 0.7 DNC	92.5 ± 5.9 59.0 ± 1.7 DNC	94.1 ± 5.0 57.6 ± 1.7 DNC
K_3-K_5	1-WLOA$_{\mathcal{F}}$	100.0 ± 0.0 37.1 ± 1.4 DNC	100.0 ± 0.0 82.5 ± 1.8 DNC	98.9 ± 1.7 66.7 ± 1.1 DNC	93.3 ± 4.3 **59.6** ± 1.0 DNC	94.6 ± 2.7 **57.1** ± 1.3 DNC
K_3-K_6	1-WLOA$_{\mathcal{F}}$	100.0 ± 0.0 38.3 ± 1.6 DNC	99.6 ± 1.1 83.4 ± 1.4 DNC	98. ± 1.8 5 66.8 ± 0.6 DNC	94.6 ± 4.6 58.9 ± 1.1 DNC	93.2 ± 6.1 56.4 ± 2.5 DNC

Table 6: Mean test accuracies of MPNN architectures on ER graphs for different levels of sparsity and different subgraphs. NB—Only one class in the dataset.

Subgraph	Algorithm	Probability			
		0.05	0.1	0.2	0.3
C_3	MPNN	90.7 ± 0.5	50.7 ± 1.0	20.1 ± 0.7	9.5 ± 0.6
	MPNN$_{\mathcal{F}}$	**98.8** ± 0.3	**96.1** ± 0.3	**72.5** ± 1.3	**32.9** ± 1.0
C_4	MPNN	90.7 ± 0.7	51.3 ± 1.4	20.1 ± 0.8	9.9 ± 0.9
	MPNN$_{\mathcal{F}}$	**99.1** ± 0.2	**96.2** ± 0.4	**73.3** ± 0.9	**33.1** ± 1.3
C_5	MPNN	90.6 ± 0.5	51.3 ± 1.2	20.1 ± 0.6	9.9 ± 0.6
	MPNN$_{\mathcal{F}}$	**98.9** ± 0.2	**96.2** ± 0.4	**73.6** ± 1.0	**33.8** ± 1.0
K_4	MPNN	NB	51.0 ± 1.2	20.1 ± 1.0	10.1 ± 1.3
	MPNN$_{\mathcal{F}}$	NB	**96.2** ± 0.5	**72.9** ± 0.9	**33.6** ± 1.2

Q2 (*"Can the increased predictive performance of a more expressive variant of the* 1-*WL algorithm be explained by an increased margin?")* See Tables 3 and 5 and Figure 1. On the TUDATASETS, an increased margin often leads to less difference between train and test accuracy; see Table 5. For example, on the PROTEINS datasets, the 1-$\mathsf{WL}_{\mathcal{F}}$, for all $\mathcal{F}$, leads to a larger difference, while its margin is always *strictly smaller* than 1-WL's margin. Hence, the empirical results align with our theory, i.e., a smaller margin worsens the generalization error. Similar effects can be observed for all other datasets, except MUTAG. On the ER dataset, comparing the 1-WL and 1-$\mathsf{WL}_{\mathcal{F}}$, for all $\mathcal{F}$, we can clearly confirm the theoretical results. That is, the 1-WL cannot separate any dataset with a positive margin, while the 1-$\mathsf{WL}_{\mathcal{F}}$ can, and we observe a decreased difference between 1-$\mathsf{WL}_{\mathcal{F}}$'s train and test accuracies compared to the 1-WL. Analyzing the 1-$\mathsf{WL}_{\mathcal{F}}$ further, for all $\mathcal{F}$, a decreasing margin always results in an increased difference between test and train accuracies. For example, for $\mathcal{F} = \{C_4\}$ and $p = 0.05$, the 1-$\mathsf{WL}_{\mathcal{F}}$ achieves a margin of 0.037 with a difference of 0.1 %, for $p = 0.1$, it achieves a margin of 0.009 with a difference of 1.8 %, for $p = 0.1$, it achieves a margin of 0.002 with a difference of 31.2 %, and, for $p = 0.3$, it achieve a margin of 0.003 with difference of 95.8 %, confirming the theoretical results. Moreover, see Figure 1 of visual illustration of this observation.

Q3 (*"Does the* 1-$\mathsf{WLOA}_{\mathcal{F}}$ *lead to increased predictive performance?")* See Tables 1, 3 and 5. Table 5 shows that the 1-$\mathsf{WLOA}_{\mathcal{F}}$ kernel performs similarly to the 1-$\mathsf{WL}_{\mathcal{F}}$, while sometimes achieving better accuracies, e.g., on the PTC_FM and PTC_MR datasets. Table 1 confirms this observation for the empirical verification of Proposition 9, i.e., the 1-$\mathsf{WLOA}_{\mathcal{F}}$ achieve perfect accuracy scores. The results are less clear for the ER datasets. On some datasets, e.g., edge probability 0.05, the 1-$\mathsf{WLOA}_{\mathcal{F}}$ performs similarly to the 1-$\mathsf{WL}_{\mathcal{F}}$ architecture. However, the algorithm leads to significantly worse predictive performance on other datasets. We speculate this is due to numerical problems of the kernelized SVM implementation.

Q4 (*"Do the results lift to MPNNs?")* See Tables 1, 4 and 6. Table 1 shows that Proposition 9 also lifts to MPNNs, i.e., like the 1-$\mathsf{WL}_{\mathcal{F}}$ kernel, the MPNN$_{\mathcal{F}}$ architecture achieves perfect prediction accuracies while the standard MPNNs does not perform better than random. In addition, on the TUDatasets, the MPNN$_{\mathcal{F}}$ architecture clearly outperforms the standard MPNN over all datasets; see Table 4. For example, on the ENZYMES dataset, the MPNN$_{\mathcal{F}}$ architecture beats the MPNN by at least 7 %, for all subgraph choices. This observation holds across all datasets. Moreover, the MPNN$_{\mathcal{F}}$ architectures also achieve better predictive performance on all ER datasets, see Table 6, compared to MPNNs. For example on the dataset using edge probability 0.2 and $\mathcal{F} = \{C_5\}$, the test accuracies of the MPNN$_{\mathcal{F}}$ architecture improves over the MPNN by more than 52 %.