Weisfeiler–Leman at the margin

Billy J. Franks[*]
University of Kaiserslautern-Landau

Christopher Morris[*]
RWTH Aachen University

Ameya Velingker
Google Research

Floris Geerts
University of Antwerp

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 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.

1 Introduction

Graph-structured data are common in fields such as chemo- and bioinformatics [55, 100, 114], combinatorial optimization [25], image analysis [97], and social-network analysis [32], highlighting the need for effective machine learning methods for graphs. Current approaches include *graph kernels* [20, 63] and *message-passing graph neural networks* (MPNNs) [44, 93]. Notably, 1-WL [112] and its MPNN counterparts [78, 115] have recently improved vertex- and graph-level learning [80]. However, 1-WL's limitations in distinguishing non-isomorphic graphs [7, 24] have led to more expressive extensions [80]. For instance, Bouritsas et al. [21] enhanced 1-WL and MPNNs by incorporating subgraph information, showing that this approach improves graph discrimination and predictive performance compared to 1-WL and k-WL [24]. *Yet, the reasons behind these performance improvements remain unclear.* Recent work [82] using 1-WL to analyze the VC dimension of MPNNs does not clarify why increased expressive power correlates with better generalization performance. Specifically, while higher VC dimension reflects that more non-isomorphic graphs can be differentiated by 1-WL, it also worsens generalization performance. This issue is similarly relevant for 1-WL-based kernels. See Appendix A for a discussion of related work.

Here, based on Alon et al. [3]'s theory of partial concepts, we derive *tight* upper and lower bounds for the VC dimension of the 1-WL-based kernels, corresponding MPNNs, and more expressive architectures, parameterized by the *margin* separating the data. Our theory establishes the first link between increased expressive power and improved generalization performance. In addition, building on Ji and Telgarsky [53], we show that gradient flow pushes the MPNN's weights toward the maximum margin solution.

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

[*]Equal contribution.

B. J. Franks et al., Weisfeiler–Leman at the margin (Extended Abstract). Presented at the Third Learning on Graphs Conference (LoG 2024), Virtual Event, November 26–29, 2024.

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 $\boldsymbol{x} \in \mathbb{R}^d$.

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. [3], 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 [106] 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 $c \in \mathbb{H}$ such that $c(x_i) = \tau_i$, for all $i \in [m]$. In essence, Alon et al. [3] 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* [27], 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 $\mathbb{H}_{r,\lambda}(\mathbb{R}^d)$ as follows

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

Alon et al. [3] 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 [5].

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 and Leman [112]. 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 $\mathsf{RELABEL}$ 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_{t+1} 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. [95], 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., $\phi_{\mathsf{WL}}^{(T)}(G) \coloneqq \big[\phi_0(G), \ldots, \phi_T(G)\big]$, where $[\ldots]$ denote column-wise vector concatenation. This results in the kernel $k_{\mathsf{WL}}^{(T)}(G, H) \coloneqq \langle \phi_{\mathsf{WL}}^{(T)}(G), \phi_{\mathsf{WL}}^{(T)}(H)\rangle$, where $\langle \cdot, \cdot \rangle$ denotes the standard inner product. We further define the *normalized* 1-WL *feature vector* $\overline{\phi_{\mathsf{WL}}^{(T)}}(G) \coloneqq {\phi_{\mathsf{WL}}^{(T)}(G)}/{\|\phi_{\mathsf{WL}}^{(T)}(G)\|}$ obtained by normalizing the 1-WL feature vector to unit length.

Weisfeiler–Leman optimal assignment kernel. Based on the 1-WL, Kriege et al. [61] 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. [61] 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 $\phi_{\mathsf{WLOA}}^{(T)}$ for the set of n-order graphs. From the theory developed in Kriege et al. [61], 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 [7, 24]. However, there exists a large set of more expressive extensions of the 1-WL, which have been successfully leveraged as kernel or neural architectures [80]. 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-$\mathsf{WL}_{\mathcal{F}}$. It is a simplified variant of the algorithms defined in Bouritsas et al. [21], 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 $C_t^{1,\mathcal{F}} \colon V(G) \to \mathbb{N}$, where $C_0^{1,\mathcal{F}}(v) \coloneqq \ell_{\mathcal{F}}(v)$ and $C_t^{1,\mathcal{F}}(v) \coloneqq$

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

for $v \in V(G)$. Hence, the 1-$\mathsf{WL}_{\mathcal{F}}$ only differs from the 1-WL at the initialization step. In Proposition 11, we show that the 1-$\mathsf{WL}_{\mathcal{F}}$ is more expressive than the 1-WL. We can also define a 1-WLOA variant of the 1-$\mathsf{WL}_{\mathcal{F}}$, which we denote by 1-$\mathsf{WLOA}_{\mathcal{F}}$. See Appendix C for how to derive kernels based on the 1-$\mathsf{WL}_{\mathcal{F}}$. See Appendix D for a formal definition of MPNNs and more expressive variants.

3 Weisfeiler–Leman at the margin: When more expressivity matters

Here, we prove lower and upper bounds on the VC dimension of 1-WL-based kernels, MPNNs, and their more expressive generalizations. 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 ${r^2}/{\lambda^2}$ [3, 14]. For the lower bound, the following lemma, implicit in Alon et al. [3], states sufficient conditions for $\mathbb{S}$ such that the VC dimension of $\mathbb{H}_{r,\lambda}(\mathbb{S})$ is also lower-bounded by ${r^2}/{\lambda^2}$.

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 ${}^{r^2}\!/_{\lambda^2}$. 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.

We first consider the class of graph embeddings obtained by the 1-WL feature map after $T \geq 0$ iterations, i.e., $\mathcal{E}_{\mathsf{WL}}(n, d_T) \coloneqq \{G \mapsto \phi^{(T)}_{\mathsf{WL}}(G) \mid G \in \mathcal{G}_n\}$ and its normalized counterpart $\bar{\mathcal{E}}_{\mathsf{WL}}(n, d_T) \coloneqq \{G \mapsto \overline{\phi^{(T)}_{\mathsf{WL}}}(G) \mid G \in \mathcal{G}_n\}$, 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 $\mathsf{VC\text{-}dim}(\mathbb{H}_{1,\lambda}(\bar{\mathcal{E}}_{\mathsf{WL}}(n, d_T))) \in \Theta({}^{1}\!/_{\lambda^2})$.

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-$\mathsf{WLOA}$, and 1-$\mathsf{WLOA}_{\mathcal{F}}$. The only difference is that $\|\phi^{(t)}_{\mathsf{WL}}(G_i)\| \neq \|\phi^{(t)}_{\mathsf{WLOA}}(G_i)\|$ 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 $\sqrt{Tn}$ 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. See Appendix E.4 for analogous results for MPNN and more expressive variants. In addition, in Appendix F, we show that gradient flow pushes the MPNN's weights toward the maximum margin solution.

In the full paper, we derive conditions under which 1-$\mathsf{WLOA}_{\mathcal{F}}$ *leads to better generalization performance than the* 1-WLOA*. We also report on empirical results, validating our derived bounds in practice.*

4 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. 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. In addition, we introduced variations of expressive 1-WL-based kernels and neural architectures with provable generalization properties. 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.

References

[1] A. Aamand, J. Y. Chen, P. Indyk, S. Narayanan, R. Rubinfeld, N. Schiefer, S. Silwal, and T. Wagner. Exponentially improving the complexity of simulating the Weisfeiler-Lehman test with graph neural networks. *ArXiv preprint*, 2022. 11

[2] R. Abboud, İ. İ. Ceylan, M. Grohe, and T. Lukasiewicz. The surprising power of graph neural networks with random node initialization. In *Joint Conference on Artificial Intelligence*, pages 2112–2118, 2021. 11

[3] N. Alon, S. Hanneke, R. Holzman, and S. Moran. A theory of PAC learnability of partial concept classes. In *Annual Symposium on Foundations of Computer Science*, pages 658–671, 2021. 1, 2, 3, 12, 18

[4] T. Amir, S. J. Gortler, I. Avni, R. Ravina, and N. Dym. Neural injective functions for multisets, measures and graphs via a finite witness theorem. *ArXiv preprint*, 2023. 11

[5] M. Anthony and P. L. Bartlett. *Neural Network Learning - Theoretical Foundations*. Cambridge University Press, 2002. 2

[6] S. Arora, N. Cohen, and E. Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, pages 244–253, 2018. 25

[7] V. Arvind, J. Köbler, G. Rattan, and O. Verbitsky. On the power of color refinement. In *International Symposium on Fundamentals of Computation Theory*, pages 339–350, 2015. 1, 3

[8] W. Azizian and M. Lelarge. Characterizing the expressive power of invariant and equivariant graph neural networks. In *International Conference on Learning Representations*, 2021. 11

[9] M. Balcilar, P. Héroux, B. Gaüzère, P. Vasseur, S. Adam, and P. Honeine. Breaking the limits of message passing graph neural networks. In *International Conference on Machine Learning*, pages 599–608, 2021. 11

[10] A. Baranwal, K. Fountoulakis, and A. Jagannath. Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization. In *International Conference on Machine Learning*, 2021. 12

[11] P. Barceló, E. V. Kostylev, M. Monet, J. Pérez, J. L. Reutter, and J. P. Silva. The logical expressiveness of graph neural networks. In *International Conference on Learning Representations*, 2020. 11

[12] P. Barceló, F. Geerts, J. L. Reutter, and M. Ryschkov. Graph neural networks with local graph parameters. In *Advances in Neural Information Processing Systems*, pages 25280–25293, 2021. 11

[13] P. Barceló, M. Galkin, C. Morris, and M. A. R. Orth. Weisfeiler and Leman go relational. In *Learning of Graphs Conference*, 2022. 11

[14] P. Bartlett and J. Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. *Advances in Kernel methods—support vector learning*, pages 43–54, 1999. 3, 17

[15] I. I. Baskin, V. A. Palyulin, and N. S. Zefirov. 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. 11

[16] D. Beaini, S. Passaro, V. Létourneau, W. L. Hamilton, G. Corso, and P. Lió. Directional graph networks. In *International Conference on Machine Learning*, pages 748–758, 2021. 11

[17] B. Bevilacqua, F. Frasca, D. Lim, B. Srinivasan, C. Cai, G. Balamurugan, M. M. Bronstein, and H. Maron. Equivariant subgraph aggregation networks. In *International Conference on Learning Representations*, 2022. 11

[18] C. Bodnar, F. Frasca, N. Otter, Y. G. Wang, P. Liò, G. Montúfar, and M. M. Bronstein. Weisfeiler and Lehman go cellular: CW networks. In *Advances in Neural Information Processing Systems*, pages 2625–2640, 2021. 11

[19] C. Bodnar, F. Frasca, Y. Wang, N. Otter, G. F. Montúfar, P. Lió, and M. M. Bronstein. Weisfeiler and Lehman go topological: Message passing simplicial networks. In *International Conference on Machine Learning*, pages 1026–1037, 2021. 11

[20] K. M. Borgwardt, M. E. Ghisu, F. Llinares-López, L. O'Bray, and B. Rieck. Graph kernels: State-of-the-art and future challenges. *Foundations and Trends in Machine Learning*, 13(5–6), 2020. 1, 11, 12

[21] G. Bouritsas, F. Frasca, S. Zafeiriou, and M. M. Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *ArXiv preprint*, 2020. 1, 3, 11

[22] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and deep locally connected networks on graphs. In *International Conference on Learning Representation*, 2014. 11

[23] J. Böker, R. Levie, N. Huang, S. Villar, and C. Morris. Fine-grained expressivity of graph neural networks. In *Advances in Neural Information Processing Systems*, 2023. 11

[24] J. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identifications. *Combinatorica*, 12(4):389–410, 1992. 1, 3

[25] Q. Cappart, D. Chételat, E. B. Khalil, A. Lodi, C. Morris, and P. Veličković. Combinatorial optimization and reasoning with graph neural networks. In *Joint Conference on Artificial Intelligence*, pages 4348–4355, 2021. 1

[26] Z. Chen, S. Villar, L. Chen, and J. Bruna. On the equivalence between graph isomorphism testing and function approximation with gnns. In *Advances in Neural Information Processing Systems*, pages 15868–15876, 2019. 11

[27] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. 2, 12

[28] L. Cotta, C. Morris, and B. Ribeiro. Reconstruction for powerful graph representations. In *Advances in Neural Information Processing Systems*, pages 1713–1726, 2021. 11

[29] G. Dasoulas, L. D. Santos, K. Scaman, and A. Virmaux. Coloring graph neural networks for node disambiguation. In *International Joint Conference on Artificial Intelligence*, pages 2126–2132, 2020. 11

[30] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3837–3845, 2016. 11

[31] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems*, pages 2224–2232, 2015. 11

[32] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010. 1

[33] R. El-Yaniv and D. Pechyony. Transductive rademacher complexity and its applications. In *Annual Conference on Learning Theory*, pages 157–171, 2007. 12

[34] P. M. Esser, L. C. Vankadara, and D. Ghoshdastidar. Learning theory can (sometimes) explain generalisation in graph neural networks. In *Advances in Neural Information Processing Systems*, pages 27043–27056, 2021. 12

[35] J. Feng, Y. Chen, F. Li, A. Sarkar, and M. Zhang. How powerful are k-hop message passing graph neural networks. In *Advances in Neural Information Processing Systems*, 2022. 11

[36] B. Finkelshtein, X. Huang, M. Bronstein, and İ. İ. Ceylan. Cooperative graph neural networks. *ArXiv preprint*, 2023. 11

[37] B. J. Franks, M. Anders, M. Kloft, and P. Schweitzer. A systematic approach to universal random features in graph neural networks. *Transactions on Machine Learning Research*, 2023. 11

[38] F. Frasca, B. Bevilacqua, M. M. Bronstein, and H. Maron. Understanding and extending subgraph GNNs by rethinking their symmetries. *ArXiv preprint*, 2022. 11

[39] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro. Convolutional neural network architectures for signals supported on graphs. *IEEE Transactions on Signal Processing*, 67(4):1034–1049, 2019. 11

[40] V. K. Garg, S. Jegelka, and T. S. Jaakkola. Generalization and representational limits of graph neural networks. In *International Conference on Machine Learning*, pages 3419–3430, 2020. 12

[41] F. Geerts. The expressive power of kth-order invariant graph networks. *ArXiv preprint*, 2020. 11

[42] F. Geerts and J. L. Reutter. Expressiveness and approximation properties of graph neural networks. In *International Conference on Learning Representations*, 2022. 11

[43] F. Geerts, F. Mazowiecki, and G. A. Pérez. Let's agree to degree: Comparing graph convolutional networks in the message-passing framework. In *International Conference on Machine Learning*, pages 3640–3649, 2021. 11

[44] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272, 2017. 1, 11, 12, 13

[45] O. Goldreich. Introduction to testing graph properties. In *Property Testing*. Springer, 2010. 12

[46] C. Goller and A. Küchler. Learning task-dependent distributed representations by backpropagation through structure. In *International Conference on Neural Networks*, pages 347–352, 1996. 11

[47] M. Grohe. The descriptive complexity of graph neural networks. *ArXiv preprint*, 2023. 11

[48] A. Grønlund, L. Kamma, and K. G. Larsen. Near-tight margin-based generalization bounds for support vector machines. In *International Conference on Machine Learning*, pages 3779–3788, 2020. 12

[49] W. L. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017. 11

[50] B. Hammer. Generalization ability of folding networks. *IEEE Trans. Knowl. Data Eng.*, (2): 196–206, 2001. 12

[51] M. Horn, E. D. Brouwer, M. Moor, Y. Moreau, B. Rieck, and K. M. Borgwardt. Topological graph neural networks. In *International Conference on Learning Representations*, 2022. 11

[52] Y. Huang, X. Peng, J. Ma, and M. Zhang. Boosting the cycle counting power of graph neural networks with I^2-GNNs. *ArXiv preprint*, 2022. 11

[53] Z. Ji and M. Telgarsky. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2019. 1, 22, 24, 25, 26, 27, 28, 29

[54] H. Ju, D. Li, A. Sharma, and H. R. Zhang. Generalization in graph neural networks: Improved pac-bayesian bounds on graph diffusion. *ArXiv preprint*, 2023. 12

[55] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 2021. 1

[56] M. Karpinski and A. Macintyre. Polynomial bounds for VC dimension of sigmoidal and general Pfaffian neural networks. *Journal of Computer and System Sciences*, 54(1):169–176, 1997. 11

[57] J. Kim, T. D. Nguyen, S. Min, S. Cho, M. Lee, H. Lee, and S. Hong. Pure transformers are powerful graph learners. *ArXiv preprint*, 2022. 11

[58] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 13

[59] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. 11, 12

[60] D. B. Kireev. Chemnet: A novel neural network based method for graph/property mapping. *Journal of Chemical Information and Computer Sciences*, 35(2):175–180, 1995. 11

[61] N. M. Kriege, P. Giscard, and R. C. Wilson. On valid optimal assignment kernels and applications to graph classification. In *Advances in Neural Information Processing Systems*, pages 1615–1623, 2016. 3, 11

[62] N. M. Kriege, C. Morris, A. Rey, and C. Sohler. A property testing framework for the theoretical expressivity of graph kernels. In *International Joint Conference on Artificial Intelligence*, pages 2348–2354, 2018. 11, 12

[63] N. M. Kriege, F. D. Johansson, and C. Morris. A survey on graph kernels. *Applied Network Science*, 5(1):6, 2020. 1, 11, 12

[64] R. Levie. A graphon-signal analysis of graph neural networks. In *Advances in Neural Information Processing Systems*, 2023. 12

[65] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1): 97–109, 2019. 11

[66] P. Li, Y. Wang, H. Wang, and J. Leskovec. Distance encoding: Design provably more powerful neural networks for graph representation learning. In *Advances in Neural Information Processing Systems*, 2020. 11

[67] R. Liao, R. Urtasun, and R. S. Zemel. A PAC-Bayesian approach to generalization bounds for graph neural networks. In *International Conference on Learning Representations*, 2021. 12

[68] T. Maehara and H. NT. A simple proof of the universality of invariant/equivariant graph neural networks. *ArXiv preprint*, 2019. 11

[69] H. Maron, H. Ben-Hamu, H. Serviansky, and Y. Lipman. Provably powerful graph networks. In *Advances in Neural Information Processing Systems*, pages 2153–2164, 2019. 11

[70] K. Martinkus, P. A. Papp, B. Schesch, and R. Wattenhofer. Agent-based graph neural networks. *ArXiv preprint*, 2022. 11

[71] S. Maskey, Y. Lee, R. Levie, and G. Kutyniok. Generalization analysis of message passing neural networks on large random graphs. In *Advances in Neural Information Processing Systems*, 2022. 12

[72] C. Merkwirth and T. Lengauer. Automatic generation of complementary descriptors with molecular graph networks. *Journal of Chemical Information and Modeling*, 45(5):1159–1168, 2005. 11

[73] A. Micheli. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009. 11

[74] A. Micheli and A. S. Sestito. 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*, pages 10–17, 2005. 11

[75] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT Press, 2012. 12

[76] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5425–5434, 2017. 11

[77] C. Morris, K. Kersting, and P. Mutzel. Glocalized Weisfeiler-Lehman kernels: Global-local feature maps of graphs. In *IEEE International Conference on Data Mining*, pages 327–336, 2017. 11

[78] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe. Weisfeiler and Leman go neural: Higher-order graph neural networks. In *AAAI Conference on Artificial Intelligence*, pages 4602–4609, 2019. 1, 11, 13, 14, 20, 21

[79] C. Morris, G. Rattan, and P. Mutzel. Weisfeiler and Leman go sparse: Towards higher-order graph embeddings. In *Advances in Neural Information Processing Systems*, 2020. 11

[80] C. Morris, Y. L., H. Maron, B. Rieck, N. M. Kriege, M. Grohe, M. Fey, and K. Borgwardt. Weisfeiler and Leman go machine learning: The story so far. *ArXiv preprint*, 2021. 1, 3, 11, 13

[81] C. Morris, G. Rattan, S. Kiefer, and S. Ravanbakhsh. SpeqNets: Sparsity-aware permutation-equivariant graph networks. In *International Conference on Machine Learning*, pages 16017–16042, 2022. 11

[82] C. Morris, F. Geerts, J. Tönshoff, and M. Grohe. WL meet VC. In *International Conference on Machine Learning*, pages 25275–25302, 2023. 1, 12, 20

[83] L. Müller, M. Galkin, C. Morris, and L. Rampásek. Attending to graph transformers. *ArXiv preprint*, 2023. 11

[84] R. L. Murphy, B. Srinivasan, V. A. Rao, and B. Ribeiro. Relational pooling for graph representations. In *International Conference on Machine Learning*, pages 4663–4673, 2019. 11

[85] H. Nguyen and T. Maehara. Graph homomorphism convolution. In *International Conference on Machine Learning*, pages 7306–7316, 2020. 11

[86] P. A. Papp and R. Wattenhofer. A theoretical comparison of graph neural network extensions. In *International Conference on Machine Learning*, pages 17323–17345, 2022. 11

[87] P. A. Papp, L. F. K. Martinkus, and R. Wattenhofer. DropGNN: Random dropouts increase the expressiveness of graph neural networks. In *Advances in Neural Information Processing Systems*, 2021. 11

[88] O. Puny, D. Lim, B. T. Kiani, H. Maron, and Y. Lipman. Equivariant polynomials for graph neural networks. *ArXiv preprint*, 2023. 11

[89] C. Qian, G. Rattan, F. Geerts, C. Morris, and M. Niepert. Ordered subgraph aggregation networks. In *Advances in Neural Information Processing Systems*, 2022. 11

[90] C. Qian, A. Manolache, K. Ahmed, Z. Zeng, G. V. den Broeck, M. Niepert, and C. Morris. Probabilistically rewired message-passing neural networks. *ArXiv preprint*, 2023. 11

[91] E. Rosenbluth, J. Tönshoff, and M. Grohe. Some might say all you need is sum. *ArXiv preprint*, 2023. 11

[92] R. Sato, M. Yamada, and H. Kashima. Random features strengthen graph neural networks. In *SIAM International Conference on Data Mining*, pages 333–341, 2021. 11

[93] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. 1, 11, 13

[94] F. Scarselli, A. C. Tsoi, and M. Hagenbuchner. The Vapnik-Chervonenkis dimension of graph and recursive neural networks. *Neural Networks*, pages 248–259, 2018. 11, 12

[95] N. Shervashidze, S. V. N. Vishwanathan, T. H. Petri, K. Mehlhorn, and K. M. Borgwardt. Efficient graphlet kernels for large graph comparison. In *International Conference on Artificial Intelligence and Statistics*, pages 488–495, 2009. 2

[96] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, pages 2539–2561, 2011. 11

[97] M. Simonovsky and N. Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 29–38, 2017. 1

[98] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19:70:1–70:57, 2018. 28

[99] A. Sperduti and A. Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–35, 1997. 11

[100] J. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N. Donghia, C. MacNair, S. French, L. Carfrae, Z. Bloom-Ackerman, V. Tran, A. Chiappino-Pepe, A. Badran, I. Andrews, E. Chory, G. Church, E. Brown, T. Jaakkola, R. Barzilay, and J. Collins. A deep learning approach to antibiotic discovery. *Cell*, pages 688–702.e13, 2020. 1

[101] R. Talak, S. Hu, L. Peng, and L. Carlone. Neural trees for learning on graphs. *ArXiv preprint*, 2021. 11

[102] E. H. Thiede, W. Zhou, and R. Kondor. Autobahn: Automorphism-based graph neural nets. In *Advances in Neural Information Processing Systems*, pages 29922–29934, 2021. 11

[103] I. O. Tolstikhin and D. Lopez-Paz. Minimax lower bounds for realizable transductive classification. *ArXiv preprint*, 2016. 12

[104] J. Tönshoff, M. Ritzert, H. Wolf, and M. Grohe. Graph learning with 1D convolutions on random walks. *ArXiv preprint*, 2021. 11

[105] V. Vapnik. *Statistical learning theory*. Wiley, 1998. 12

[106] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995. 2, 11

[107] V. N. Vapnik and A. Chervonenkis. A note on one class of perceptrons. *Avtomatika i Telemekhanika*, 24(6):937–945, 1964. 12

[108] A. Velingker, A. K. Sinop, I. Ktena, P. Velickovic, and S. Gollapudi. Affinity-aware graph networks. *ArXiv preprint*, 2022. 11

[109] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. 11

[110] S. Verma and Z. Zhang. Stability and generalization of graph convolutional neural networks. In *International Conference on Knowledge Discovery & Data Mining*, pages 1539–1548, 2019. 12

[111] C. Vignac, A. Loukas, and P. Frossard. Building powerful and equivariant graph neural networks with structural message-passing. In *Advances in Neural Information Processing Systems*, 2020. 11

[112] B. Weisfeiler and A. Leman. 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`. 1, 2

[113] A. Wijesinghe and Q. Wang. A new perspective on "how graph neural networks go beyond weisfeiler-lehman?". In *International Conference on Learning Representations*, 2022. 11

[114] F. Wong, E. J. Zheng, J. A. Valeri, N. M. Donghia, M. N. Anahtar, S. Omori, A. Li, A. Cubillos-Ruiz, A. Krishnan, W. Jin, A. L. Manson, J. Friedrichs, R. Helbig, B. Hajian, D. K. Fiejtek, F. F. Wagner, H. H. Soutter, A. M. Earl, J. M. Stokes, L. D. Renner, and J. J. Collins. Discovery of a structural class of antibiotics with explainable deep learning. *Nature*, 2023. 1

[115] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. 1, 11, 13

[116] P. Yanardag and S. V. N. Vishwanathan. A structural smoothing framework for robust graph comparison. In *Advances in Neural Information Processing Systems*, pages 2134–2142, 2015. 11

[117] P. Yanardag and S. V. N. Vishwanathan. Deep graph kernels. In *International Conference on Knowledge Discovery and Data Mining*, pages 1365–1374, 2015. 11

[118] G. Yehudai, E. Fetaya, E. A. Meirom, G. Chechik, and H. Maron. From local structures to size generalization in graph neural networks. In *International Conference on Machine Learning*, pages 11975–11986, 2021. 12

[119] J. You, J. Gomes-Selman, R. Ying, and J. Leskovec. Identity-aware graph neural networks. In *AAAI Conference on Artificial Intelligence*, pages 10737–10745, 2021. 11

[120] B. Zhang, G. Feng, Y. Du, D. He, and L. Wang. A complete expressiveness hierarchy for subgraph gnns via subgraph weisfeiler-lehman tests. *ArXiv preprint*, 2023. 11

[121] B. Zhang, S. Luo, L. Wang, and D. He. Rethinking the expressive power of GNNs via graph biconnectivity. *ArXiv preprint*, 2023. 11

[122] M. Zhang and P. Li. Nested graph neural networks. In *Advances in Neural Information Processing Systems*, pages 15734–15747, 2021. 11

[123] L. Zhao, W. Jin, L. Akoglu, and N. Shah. From stars to subgraphs: Uplifting any GNN with local structure awareness. In *International Conference on Learning Representations*, 2022. 11

A Related work

In the following, we discuss relevant related work.

A.1 Graph kernels based on the 1-WL

Shervashidze et al. [96] were the first to utilize the 1-WL as a graph kernel. Later, Morris et al. [77, 79, 81] generalized this to variants of the k-WL. Moreover, Kriege et al. [61] derived the *Weisfeiler-Leman optimal assignment kernel*, using the 1-WL to compute optimal assignments between vertices of two given graphs. Yanardag and Vishwanathan [116] successfully employed the Weisfeiler–Leman kernels within frameworks for smoothed [116] and deep graph kernels [117]. For a theoretical investigation of graph kernels based on the 1-WL, see [62]. See also [80] for an overview of the Weisfeiler–Leman algorithm in machine learning and Borgwardt et al. [20], Kriege et al. [63] for a detailed review of graph kernels.

A.2 MPNNs

Recently, MPNNs [44, 93] emerged as the most prominent graph representation learning architecture. Notable instances of this architecture include, e.g., Duvenaud et al. [31], Hamilton et al. [49], and Veličković et al. [109], which can be subsumed under the message-passing framework introduced in Gilmer et al. [44]. In parallel, approaches based on spectral information were introduced in, e.g., Bruna et al. [22], Defferrard et al. [30], Gama et al. [39], Kipf and Welling [59], Levie et al. [65], and Monti et al. [76]—all of which descend from early work in Baskin et al. [15], Goller and Küchler [46], Kireev [60], Merkwirth and Lengauer [72], Micheli [73], Micheli and Sestito [74], Scarselli et al. [93], and Sperduti and Starita [99]. Rcently, connections between MPNNs and Weisfeiler–Leman-type algorithms have been shown [11, 43, 78, 115]. Specifically, Morris et al. [78] and Xu et al. [115] showed that the 1-WL limits the expressive power of any possible MPNN architecture in distinguishing non-isomorphic graphs. [21] showed how to make MPNNs more expressive by incorporating subgraph information.

A.3 Expressive power of MPNNs

Recently, connections between MPNNs and Weisfeiler–Leman-type algorithms have been shown [11, 43, 78, 115]. Specifically, Morris et al. [78] and Xu et al. [115] 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 and Lelarge [8], Geerts [41], Maron et al. [69], Morris et al. [78, 79, 81], and connected to the permutation-equivariant function approximation over graphs, see, e.g., Azizian and Lelarge [8], Chen et al. [26], Geerts and Reutter [42], Maehara and NT [68]. Furthermore, Aamand et al. [1], Amir et al. [4] 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 [84, 111], using random features [2, 29, 92] or individualization-refinement algorithms [37], affinity measures [108], equivariant graph polynomials [88], homomorphism and subgraph counts [12, 21, 85], spectral information [9], simplicial [19] and cellular complexes [18], persistent homology [51], random walks [70, 104], graph decompositions [101], relational [13], distance [66] and directional information [16], graph rewiring [90] and adaptive message passing [36], subgraph information [17, 28, 35, 38, 52, 80, 86, 87, 89, 102, 113, 119, 120, 122, 123], and biconnectivity [121]. See Morris et al. [80] for an in-depth survey on this topic. Geerts and Reutter [42] 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. [57] showed that transformer architectures [83] can simulate the 2-WL. Grohe [47] showed tight connections between MPNNs' expressivity and circuit complexity. Moreover, Rosenbluth et al. [91] investigated the expressive power of different aggregation functions beyond sum aggregation. Finally, Böker et al. [23] 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. [94] used classical techniques from learning theory [56] to show that MPNNs' VC dimension [106] 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 [50]. We note here that Scarselli et al. [94] analyzed a different type of MPNN not aligned with modern MPNN architectures [44]. Garg et al. [40] showed that the empirical Rademacher complexity (see, e.g., Mohri et al. [75]) 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. [67] refined these results via a PAC-Bayesian approach, further refined in Ju et al. [54]. Maskey et al. [71] used random graphs models to show that MPNNs' generalization ability depends on the (average) number of vertices in the resulting graphs. In addition, Levie [64] 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 and Zhang [110] studied the generalization abilities of 1-layer MPNNs in a transductive setting based on algorithmic stability. Similarly, Esser et al. [34] used stochastic block models to study the transductive Rademacher complexity [33, 103] of standard MPNNs. For semi-supervised node classification, [10] 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 [59]. Most recently, [82] 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. [62] leveraged results from graph property testing [45] to study the sample complexity of learning to distinguish various graph properties, e.g., planarity or triangle freeness, using graph kernels [20, 63]. Finally, [118] showed negative results for MPNNs' generalization ability to larger graphs.

Margin theory and VC dimension. Using the margin as a regularization mechanism dates back to Vapnik and Chervonenkis [107]. Later, the concept of margin was successfully applied to *support vector machines* (SVMs) [27, 105] and connected to VC dimension theory; see Mohri et al. [75] for an overview. Grønlund et al. [48] derived the so-far tightest generalization bounds for SVMs. Alon et al. [3] 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.

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* $\|\boldsymbol{x}\| \coloneqq \sqrt{x_1^2 + \dots + x_d^2}$ 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 $G[S] \coloneqq (V(G) \cap S, E(G) \cap S^2)$.

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 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 $C^{1,\mathcal{F}}_t \to \Sigma_t$ for each iteration $t \leq T$. Let Σ_t denote the *range* of $C^{1,\mathcal{F}}_t$, i.e., $\Sigma_t \coloneqq \{c \mid \exists\, v \in V(G) \colon C^{1,\mathcal{F}}_t(v) = c\}$. 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.,

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

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.

D 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. [93] and Gilmer et al. [44], 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

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

to compute a single vectorial representation based on learned vertex features after iteration L. Again, $\mathsf{READOUT}$ 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 and Ba [58], 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 [78, 115], a large set of more expressive extensions of MPNNs [80] exists. Here, we introduce the *MPNN$_{\mathcal{F}}$ architecture*, an MPNN variant of the 1-$\mathsf{WL}_{\mathcal{F}}$; 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-$\mathsf{WL}_{\mathcal{F}}$, e.g., one-hot encodings of $\ell_{\mathcal{F}}$. Following Morris et al. [78], it is straightforward to derive an MPNN$_{\mathcal{F}}$ architecture that has the same expressive power as the 1-$\mathsf{WL}_{\mathcal{F}}$ 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 D.1 for details. We stress that simple MPNNs are already expressive enough to be equivalent to the 1-WL in distinguishing non-isomorphic graphs [78]. The class $\mathsf{MPNN}_{\mathsf{mlp},\mathcal{F}}(d, L)$ is defined similarly, based on MPNN$_{\mathcal{F}}$s.

D.1 Simple MPNNs

Here, we provide more details on the simple MPNNs mentioned in Appendix D. 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 Appendix D, 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.[2] 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. [78] 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 D.1 for details. We stress that simple MPNNs are already expressive enough to be equivalent to the 1-WL in distinguishing non-isomorphic graphs [78]. The class $\mathsf{MPNN}_{\mathsf{mlp},\mathcal{F}}(d, L)$ is defined similarly, based on MPNN$_{\mathcal{F}}$s.

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, i.e.,

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

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:

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

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

.

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

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

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 6. 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

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

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

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

and

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

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

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

where $\tilde{\boldsymbol{x}}_{\boldsymbol{\alpha}}^{+}$ and $\tilde{\boldsymbol{x}}_{\boldsymbol{\alpha}}^{-}$ are derived from applying the optimization (Equation (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

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

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 $\boldsymbol{x}_i \coloneqq (\boldsymbol{x}_i^1, \boldsymbol{x}_i^2)$ with $\boldsymbol{x}_i^1 \in \mathbb{R}^{d_1}$ and $\boldsymbol{x}_i^2 \in \mathbb{R}^{d_2}$.

Proposition 7. If $X \coloneqq \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ is a sample, such that

1. $(\boldsymbol{x}_1^1, y_1), \ldots, (\boldsymbol{x}_n^1, y_n)$ is (r_1, γ_1)-separable and
2. $(\boldsymbol{x}_1^2, y_1), \ldots, (\boldsymbol{x}_n^2, y_n)$ 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 $\boldsymbol{x}_i^+ \coloneqq \boldsymbol{x}_i$, $(\boldsymbol{x}_i^1)^+ \coloneqq (\boldsymbol{x}_i^1, 0)$, $(\boldsymbol{x}_i^2)^+ \coloneqq (0, \boldsymbol{x}_i^2)$ for $i \in I^+$, and $\boldsymbol{x}_i^- \coloneqq \boldsymbol{x}_i$, $(\boldsymbol{x}_i^1)^- \coloneqq (\boldsymbol{x}_i^1, 0)$, and $(\boldsymbol{x}_i^2)^- \coloneqq (0, \boldsymbol{x}_i^2)$ for $i \in I^-$. We collect $\boldsymbol{x}_i^+$, $\boldsymbol{x}_i^-$, $(\boldsymbol{x}_i^1)^+$, $(\boldsymbol{x}_i^2)^+$, $(\boldsymbol{x}_i^1)^-$, and $(\boldsymbol{x}_i^2)^-$ 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

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

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

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

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 8. 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 9. 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 10. 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 $v, w \in V(G)$,

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

The base case, $t = 0$, is clear since 1-$\mathsf{WL}_{\mathcal{F}}$ refines the single color class induced by C^1_0. For the induction, assume that Equation (5) holds and assume that, $C^{1,\mathcal{F}}_{t+1}(v) = C^{1,\mathcal{F}}_{t+1}(w)$ holds. Hence, $C^1_t(v) = C^1_t(w)$ and

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

holds. Hence, there is a *color-preserving bijection* $\varphi \colon N(v) \to N(w)$ between the above two multisets, i.e., $C^{1,\mathcal{F}}_t(a) = C^{1,\mathcal{F}}_t(\varphi(a))$, for $a \in N(v)$. Hence, by Equation (5), $C^1_t(a) = C^1_t(\varphi(a))$, for $a \in N(v)$. Consequently, it holds that $C^1_{t+1}(v) = C^1_{t+1}(w)$, 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 11. 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 12 (Theorem 1.6 in [14])**.** 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 13 (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. [3], we show that the vectors $\boldsymbol{b}_1, \dots, \boldsymbol{b}_m$ 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

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

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 13 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 14. 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 $G_1, \dots, G_m$ 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 13. Indeed, we can simply consider G_i to be an $(i-1)$-regular graph of order n; see Lemma 8. 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 $(0, \dots, \underbrace{n}_{\text{pos } i}, \dots, 0)$ in the unnormalized case, and $\frac{1}{\sqrt{1+1n}}(0, \dots, \underbrace{n}_{\text{pos } i}, \dots, 0)$ in the normalized case. It is readily verified that $\boldsymbol{b}_i^{(2)} \coloneqq (0, \dots, \underbrace{n}_{\text{pos } i}, \dots, 0)$ and $\boldsymbol{b}_i^{(1)}$ being the remaining initial colors are vectors satisfying the assumptions of Lemma 13 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 $\|\boldsymbol{b}_i\| = \sqrt{T+1}n \coloneqq r'$. 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 13 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 15. 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 14. 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 8. 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-$\mathsf{WL}_{\mathcal{F}}$-case, the graphs in $\mathcal{F}$ change the initial colors. Because of our regular graph construction from Lemma 8, 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 16. 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 14 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 17. 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 Corollary 3 and Theorem 16. □

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

Corollary 18 (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 19 (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 20 (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 21 (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 22. 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

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

E.4 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 23. 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 MPNN$_{\mathcal{F}}$ architectures by defining $\mathcal{E}_{\mathsf{MPNN},\mathcal{F}}(n,d,T)$ analogously to the above.

Corollary 24. 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.

We now lift the above results for the 1-WL kernel to MPNNs. To prove Proposition 23, 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 25. 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 [82, 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. [78, 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 $\boldsymbol{f}_v^{(t)}$ in $\mathbb{R}^{1\times n}$ for $v \in V(G)$. That is, for vertices v and w in $V(G)$ it holds that

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

We note, by the construction outlined in the proof of Morris et al. [78, 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. [78, 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

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

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 24.

Corollary 26. 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 $G \in S$,

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

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. [78, Theorem 2] also works for vertex-labeled graphs, the proof technique for Proposition 25 can be straightforwardly lifted to the 1-$\mathsf{WL}_{\mathcal{F}}$. □

We can also extend Proposition 25 to the 1-WLOA and 1-$\mathsf{WLOA}_{\mathcal{F}}$, i.e., derive an MPNN architecture that can compute 1-WLOA's and 1-$\mathsf{WLOA}_{\mathcal{F}}$'s feature vectors. By that, we can extend Proposition 4 and Corollary 5 to their corresponding MPNN versions.

Proposition 27. 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 25, 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 $G \in S$,

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

We now show how to transform $\phi^{(T)}_{\text{1-}\mathsf{WL}}(G)$ into $\phi^{(T)}_{\mathsf{WLOA}}(G)$. We show the transformation for a single iteration $t \leq T$, i.e., transforming $\phi_{t,\text{1-}\mathsf{WL}}(G)$ into $\phi_{t,\text{1-}\mathsf{WLOA}}(G)$. 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, $\phi_{t,\text{1-}\mathsf{WL}}(G)$ has C entries. Hence, the number of components for $\phi_{t,\mathsf{WLOA}}(G)$ is at most Cn. By multiplying $\phi_{t,\text{1-}\mathsf{WL}}(G)$ 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 $\phi_{t,\text{1-}\mathsf{WL}}(G)$ 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 $\boldsymbol{r}' = \phi_{t,\text{1-}\mathsf{WLOA}}(G)$, implying the result □

In a similar way as for Corollary 26, we can lift the above result to the 1-$\mathsf{WLOA}_{\mathcal{F}}$.

Corollary 28. 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 $G \in S$,

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

F Large margins and gradient flow

Proposition 23 and Corollary 24 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 and Telgarsky [53], 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 Appendix D 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 $\boldsymbol{A}'(G) \coloneqq \boldsymbol{A}(G) + \boldsymbol{I}_n$, $\boldsymbol{I}_n \in \mathbb{R}^{n \times n}$ 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; 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 29. 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

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

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

Alignment theorems. 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 $y_i \cdot \bar{\boldsymbol{u}}^\top \boldsymbol{X}_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i} > 0$ for all i. Furthermore, the maximum margin is given by

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

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

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

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 31. Suppose Assumptions 29 and 30 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

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

- Also,

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

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

Assumption 32. 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 33 (Convergence to the maximum margin solution)**.** Suppose Assumptions 29 and 32 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.

We present the proofs of our main results from Appendix F, i.e., Theorem 41 and Theorem 44. 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 and Telgarsky [53].

F.0.1 Setup

Recall that we consider *linear* L-layer MPNNs following Appendix D 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 29. 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

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

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:

$$\frac{d\mathcal{R}(\boldsymbol{W}(t))}{dt} = \left\langle \nabla \mathcal{R}(\boldsymbol{W}(t)), \frac{d\boldsymbol{W}(t)}{dt} \right\rangle = -\|\nabla \mathcal{R}(\boldsymbol{W})\|_2^2 = -\sum_{j=1}^{L} \left\| \frac{\partial \mathcal{R}}{\partial \boldsymbol{W}^{(j)}} \right\|_F^2, \tag{6}$$

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 30. 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 and Telgarsky [53], 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 and Telgarsky [53] 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. [6] applied to our setting—our $\mathcal{R}_1$ and $\mathcal{R}$ correspond to L^1 and L^N, respectively, in the aforementioned work.

Lemma 34 (Theorem 1 in Arora et al. [6])**.** $(\boldsymbol{W}^{(j+1)})^\top(t)\boldsymbol{W}^{(j+1)}(t) - \boldsymbol{W}^{(j)}(t)(\boldsymbol{W}^{(j)})^\top(t)$ is a constant function of t.

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

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

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

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

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

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

Adding the above equation to its transpose, we obtain

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

Note that this is equivalent to

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

which implies that $(\boldsymbol{W}^{(j+1)})^\top(t)\boldsymbol{W}^{(j+1)}(t) - \boldsymbol{W}^{(j+1)}(t)(\boldsymbol{W}^{(j+1)})^\top(t)$ 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.,

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

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

Proof. The lemma is the same as the first part of Lemma 2.3 in [53]. Therefore, we defer to the proof there. □

Our main interest in Lemma 35 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 36. Under gradient flow subject to Assumptions 29 and 30, $\{t \geq 0 \colon \boldsymbol{W}(t) \in B(R)\}$ has finite measure.

Proof. The corollary corresponds to the second part of Ji and Telgarsky [53]. We reproduce the proof here. Note that since $d\mathcal{R}(\boldsymbol{W}(t))/dt = -\|\nabla \mathcal{R}(\boldsymbol{W}(t))\|_F^2 \leq 0$ for all $t \geq 0$ (see Equation (6)),

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

where the final implication holds due to Lemma 35. Since $\mathcal{R}(\boldsymbol{W}(0))$ is finite, this implies that $\{t \geq 0 \colon \boldsymbol{W}(t) \in B(R)\}$ has finite measure. □

We now define the following notation for convenience:

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

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

Proof. A proof appears in [53]; 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 38. For all $1 \leq j \leq L$, we have

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

Proof. Once again, the proof appears in [53] (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 [53]; 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 [53]). Therefore, we provide the proof below.

Lemma 39. Suppose $\min_{1 \leq j \leq L} \|\boldsymbol{W}^{(j)}(t)\|_F \to \infty$ as $t \to \infty$. For any $1 \leq j \leq L$, we have,

- $\boldsymbol{W}^{(j)}(t)/\|\boldsymbol{W}^{(j)}(t)\|_F \to \boldsymbol{u}_j(t)\boldsymbol{v}_j(t)^\top$ 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 37 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, $\boldsymbol{W}^{(j)}(t)/\|\boldsymbol{W}^{(j)}(t)\|_F \to \boldsymbol{u}_j(t)\boldsymbol{v}_j(t)^\top$, which establishes the first part.

For the second part, note that by Lemma 38 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

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

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 and Telgarsky [53]; therefore, we defer to the proofs there.

Theorem 40 (Parts 1 and 2 of Theorem 2.2 in [53])**.** 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 [53]. □

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

Theorem 41. Suppose Assumptions 29 and 30 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

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

- Also,

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

Proof. Note that by Theorem 40, we have that $\|\boldsymbol{W}^{(j)}\|_F \to \infty$ for every j. Thus, the first part of Lemma 39 implies the first part of the theorem. Note that setting $j = 1$ in the second part of Lemma 39 implies the second part of the theorem, completing the proof. □

F.0.3 Margin

We now state results on the margin.

Lemma 42. Suppose the data set $\{(\boldsymbol{X}_i, y_i)\}_{i=1}^k$ and G_i on n_i nodes are sampled according to Assumption 32. Let $S \subset \{1, 2, \ldots, k\}$ be the set of indices for support vectors. Then,

$$\min_{\substack{\|\boldsymbol{\xi}\|_2=1 \\ \langle\boldsymbol{\xi},\bar{\boldsymbol{u}}\rangle=0}} \max_{i\in S} \left\langle \boldsymbol{\xi}, Z_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i} \right\rangle > 0 \tag{7}$$

with probability 1 over the sampling.

Proof. First, we note that there are $s \leq d$ 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

$$\sum_{i\in S} \alpha_i Z_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i} = \bar{\boldsymbol{u}}. \tag{8}$$

This follows from Soudry et al. [98] (see Lemma 12 in Appendix B), which was also used by Ji and Telgarsky [53]).

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

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

Then, note that

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

This implies that $\left\langle \boldsymbol{\xi}, Z_i \boldsymbol{A}'(G_i)^L \mathbf{1}_{n_i} \right\rangle = 0$ 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 43. Suppose Assumption 32 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 (8)).

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 (7).

Observe that

$$\begin{aligned}\langle \boldsymbol{w}^{\perp}, \nabla \mathcal{R}(\boldsymbol{w}^{\top})\rangle &= \frac{1}{k}\sum_{i=1}^{k} \ell'(\langle \boldsymbol{w}, \boldsymbol{v}_i\rangle)\cdot \langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_i\rangle \\ &= -\frac{1}{k}\sum_{i=1}^{k} \exp(-\langle \boldsymbol{w}, \boldsymbol{v}_i\rangle)\cdot \langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_i^{\perp}\rangle \\ &= -\frac{1}{k}\exp(-\langle \boldsymbol{w}, \boldsymbol{v}_{j'}\rangle)\cdot \langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_{j'}^{\perp}\rangle - \frac{1}{k}\sum_{\substack{1\le i\le k \\ \langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_i^{\perp}\rangle \ge 0}} \exp(-\langle \boldsymbol{w}, \boldsymbol{v}_i\rangle)\cdot \langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_i^{\perp}\rangle.\end{aligned} \tag{9}$$

The first term on the righthand side of (9) can be bounded as follows:

$$\begin{aligned}-\frac{1}{k}\exp(-\langle \boldsymbol{w}, \boldsymbol{v}_{j'}\rangle)\cdot \langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_{j'}^{\perp}\rangle &= -\frac{1}{k}\exp\left(-\langle \boldsymbol{w}, \boldsymbol{v}_{j'}^{\parallel}\rangle - \langle \boldsymbol{w}, \boldsymbol{v}_{j'}^{\perp}\rangle\right)\cdot \langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_{j'}^{\perp}\rangle \\ &= -\frac{1}{k}\exp\left(-\langle \boldsymbol{w}^{\parallel}, \boldsymbol{v}_{j'}^{\parallel}\rangle\right)\exp\left(-\langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_{j'}^{\perp}\rangle\right)\cdot \langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_{j'}^{\perp}\rangle \\ &\ge \frac{1}{k}\exp(-\langle \boldsymbol{w}, \gamma \boldsymbol{u}\rangle)\exp(\alpha\|\boldsymbol{w}^{\perp}\|)\cdot \alpha\|\boldsymbol{w}^{\perp}\|.\end{aligned} \tag{10}$$

For the second term in (9), we have

$$\begin{aligned}\sum_{\substack{1\le i\le k \\ \langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_i^{\perp}\rangle \ge 0}} -\frac{1}{k}\exp(-\langle \boldsymbol{w}, \boldsymbol{v}_i\rangle)\cdot \langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_i^{\perp}\rangle &= \sum_{\substack{1\le i\le k \\ \langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_i^{\perp}\rangle \ge 0}} -\frac{1}{k}\exp(-\langle \boldsymbol{w}, \gamma\bar{\boldsymbol{u}}\rangle)\exp(-\langle \boldsymbol{w}, \boldsymbol{v}_i - \gamma\bar{\boldsymbol{u}}\rangle)\cdot \langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_i^{\perp}\rangle \\ &\ge \sum_{\substack{1\le i\le k \\ \langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_i^{\perp}\rangle \ge 0}} -\frac{1}{k}\exp(-\langle \boldsymbol{w}, \gamma\bar{\boldsymbol{u}}\rangle)\exp(-\langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_i^{\perp}\rangle)\cdot \langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_i^{\perp}\rangle \\ &\ge \sum_{\substack{1\le i\le k \\ \langle \boldsymbol{w}^{\perp}, \boldsymbol{v}_i^{\perp}\rangle \ge 0}} \frac{1}{k}\exp(-\langle \boldsymbol{w}, \gamma\bar{\boldsymbol{u}}\rangle)(-e^{-1}) \\ &\ge \exp(-\langle \boldsymbol{w}, \gamma\bar{\boldsymbol{u}}\rangle)(-e^{-1}),\end{aligned} \tag{11}$$

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

By plugging (10) and (11) into (9), we obtain

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

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

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

Theorem 44 (Convergence to the maximum margin solution)**.** Suppose Assumptions 29 and 32 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 [53], except that one uses Assumption 32 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 43 as in [53]. □