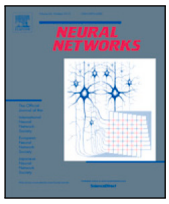
Neural Networks 182 (2025) 106924

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Full Length Article

VC dimension of Graph Neural Networks with Pfaffian activation functions

Giuseppe Alessio D'Inverno *, Monica Bianchini, Franco Scarselli

Department of Information Engineering and Mathematics, University of Siena, Via Roma 56, Siena, 53100, Italy

ARTICLE INFO

Keywords:
VC dimension
Graph Neural Networks
Pfaffian functions

ABSTRACT

Graph Neural Networks (GNNs) have emerged in recent years as a powerful tool to learn tasks across a wide range of graph domains in a data-driven fashion. Based on a message passing mechanism, GNNs have gained increasing popularity due to their intuitive formulation, closely linked to the Weisfeiler–Lehman (WL) test for graph isomorphism, to which they were demonstrated to be equivalent (Morris et al., 2019 and Xu et al., 2019). From a theoretical point of view, GNNs have been shown to be universal approximators, and their generalization capability — related to the Vapnik Chervonekis (VC) dimension (Scarselli et al., 2018) — has recently been investigated for GNNs with piecewise polynomial activation functions (Morris et al., 2023). The aim of our work is to extend this analysis on the VC dimension of GNNs to other commonly used activation functions, such as the sigmoid and hyperbolic tangent, using the framework of Pfaffian function theory. Bounds are provided with respect to the architecture parameters (depth, number of neurons, input size) as well as with respect to the number of colors resulting from the 1–WL test applied on the graph domain. The theoretical analysis is supported by a preliminary experimental study.

1. Introduction

Since Deep Learning (DL) has become a fundamental tool in approaching real-life applications (Fresca, Manzoni, Dedè, & Quarteroni, 2020; Jumper et al., 2021; Lam et al., 2023; Rolnick et al., 2022), the urgency of investigating its theoretical properties has become more evident. Neural networks were then progressively studied analyzing, for example, their expressive power in terms of approximating classes of functions (Daubechies, DeVore, Foucart, Hanin, & Petrova, 2022; Hammer, 2000; Hornik, 1991; Hornik, Stinchcombe, & White, 1989) or showing their limitations in the imitation of neurocognitive tasks (Brugiapaglia, Liu, & Tupper, 2020, 2022; D'Inverno, Brugiapaglia, & Ravanelli, 2024). The *generalization capability* of a learning model, intended as the capacity of correctly performing a specific task on unseen data, has always been a core aspect to evaluate the effectiveness of proposed architectures (Jacot, Gabriel, & Hongler, 2018; Neyshabur, Li, Bhojanapalli, LeCun, & Srebro, 2018). Several metrics and/or methods have been proposed over the years to evaluate such capability (Haussler & Warmuth, 2018; Koltchinskii, 2001). Among them, the *Vapnik Chervonenkis (VC) dimension* (Vapnik & Chervonenkis, 1968) is a metric that measures the capacity of a learning model to *shatter* a set of data points, which means that it can always realize a perfect classifier for any binary labeling of the input data. Intuitively, the greater the VC dimension of the learning model, the more it will fit the data on which it has been trained. However, as it has been shown in Vapnik (2006), a large VC dimension leads to poor generalization, i.e. to a large difference between the error evaluated on the training and on the test set. Therefore, it is important to establish the VC dimension of a model, especially with respect to its hyperparameters, in order to make it capable of generalizing on unseen data.

Graph Neural Networks (GNNs) (Scarselli, Gori, Tsoi, Hagenbuchner, & Monfardini, 2009; Zhou et al., 2020) are machine learning architectures capable of processing graphs that represent patterns (or part of patterns) along with their relationships. GNNs are among the most used deep learning models nowadays, given the impressive performance they have shown in tasks related to structured data (Liu & Zhou, 2022). A great effort has been dedicated to assess their expressive power, mainly related to the study of the so-called Weisfeiler–Lehman (WL) test (Weisfeiler & Leman, 1968) and its variants (Bodnar, Frasca, Otter, et al., 2021; Bodnar, Frasca, Wang, et al., 2021; Morris et al., 2019). Indeed, the standard WL algorithm, which checks whether two graphs are isomorphic by iteratively assigning colors to their nodes, has been proved to be equivalent to GNNs in terms of the capability of distinguishing graphs (Xu, Hu, Leskovec, & Jegelka, 2019). However, little is known about the generalization capabilities of GNNs. In Scarselli, Tsoi, and Hagenbuchner (2018), bounds for the VC dimension have been provided for the original GNN model, namely the first model being introduced. Very recently (Morris, Geerts, Tönshoff, & Grohe, 2023), bounds have been found also for a large class of modern GNNs with

* Corresponding author.
E-mail addresses: dinverno@diism.unisi.it (G.A. D'Inverno), monica.bianchini@unisi.it (M. Bianchini), franco.scarselli@unisi.it (F. Scarselli).
URL: https://www3.diism.unisi.it/dinverno/ (G.A. D'Inverno).

https://doi.org/10.1016/j.neunet.2024.106924
Received 2 April 2024; Received in revised form 19 October 2024; Accepted 12 November 2024
Available online 20 November 2024

Table 1
Summary of the main theorems reported in Section 4.

Theorem 1	General upper bound on the VC dimension for generic GNNs
Theorem 2	Specific upper bound on the VC dimension for each hyperparameter
Theorem 3	General upper bound on the VC dimension for GNNs of type (2)
Theorem 4	Specific upper bound on the VC dimension w.r.t. the colors in the dataset obtained via the 1–WL test

piecewise polynomial activation functions. Nevertheless, message passing GNNs with other common activation functions, such as hyperbolic tangent, sigmoid and arctangent, still lack characterization in terms of VC dimension. This work aims to fill this gap, providing new bounds for modern message passing GNNs with Pfaffian activation functions. *Pfaffian functions* are a large class of differentiable maps, which includes the above mentioned common activation functions, i.e, tanh, logsig, atan, and, more generally, most of the functions used in Engineering having continuous derivatives up to any order. Our main contributions are listed below.

- Section 4 provides upper bounds for message passing GNNs with Pfaffian activation functions with respect to the main hyperparameters, such as the feature dimension, the hidden feature size, the number of message passing layers implemented, and the total number of nodes in the entire training domain (Theorems 1 and 2). To address this issue, we exploit theoretical results in the literature that link the theory of Pfaffian function with the characterization of the VC dimension of GNNs via topological analysis.
- We also study the trend of the VC dimension w.r.t. the colors in the dataset obtained by running the WL test (Theorem 4). Theoretical results suggest that the number of colors have an important effect on the GNN generalization capability. On the one hand, a large total number of colors in the training set improves generalization, since it increases the examples available for learning; on the other hand, a large number of colors in each graph raises the VC dimension and therefore increases the empirical risk value. Table 1 summarizes the main results presented in Section 4;
- Our theoretical findings are assessed by a preliminary experimental study; specifically, in Section 5, we evaluate the gap between the predictive performance on the training and test data.

The manuscript is organized as follows. In Section 2, we offer an overview of work related to the addressed topic. In Section 3, we introduce the main concepts and the notation used throughout the manuscript. In Section 4, we state and discuss our main theoretical results. The preliminary experiments aimed at validating our theoretical results are described in Section 5. Finally, in Section 6, we draw some conclusions, also providing a brief discussion of open problems and future research directions.

2. Related work

In this section we collect the main contributions present in the literature relating to the generalization ability of GNNs, the calculation of the VC dimension and the theory of Pfaffian functions.

Generalization bounds for GNNs —. Several approaches have been exploited to give some insights on the generalization capabilities of GNNs. In Garg, Jegelka, and Jaakkola (2020), new bounds are provided on the *Rademacher complexity* in binary classification tasks; the study is carried out by focusing on the computation trees of the nodes, which are tightly linked to the 1–WL test (D'Inverno, Bianchini, Sampoli, & Scarselli, 2024; Krebs & Verbitsky, 2015). Similarly, in Esser, Chennuru Vankadara, and Ghoshdastidar (2021), generalization bounds for Graph Convolutional Networks (GCNs) are derived, based on the Transductive Rademacher Complexity, which differs from the standard Rademacher Complexity by taking into account unobserved instances. In Verma and Zhang (2019), the stability, and consequently the generalization capabilities of GCNs, are proved to be dependent on the largest eigenvalue of the convolutional filter; therefore, to ensure a better generalization, such eigenvalue should be independent of the graph size. Under the lens of the PAC-learnability framework, the generalization bounds reported in Garg et al. (2020) have been improved in Liao, Urtasun, and Zemel (2020), showing a tighter dependency on the maximum node degree and the spectral norm of the weights. This result aligns with the findings in Verma and Zhang (2019). In Ju, Li, Sharma, and Zhang (2023), sharper bounds on the GNN stability to noise are provided by investigating the correlation between attention and generalization. Specifically, GCNs and Graph Isomorphism Networks (GINs) are considered. The results show a link between the trace of the Hessian of the weight matrices and the stability of GNNs. A correlation between attention and generalization in GCNs and GINs is empirically investigated also in Knyazev, Taylor, and Amer (2019).

VC dimension —. Since it was first introduced in Vapnik and Chervonenkis (1968), the VC dimension has become a widespread metric to assess the generalization capabilities of neural networks. In Vapnik, Levin, and Le Cun (1994), the VC dimension is proven to be tightly related to how the test error correlates, in probability, with the training error. Bounds on the VC dimension have been evaluated for many baseline architectures, such as Multi Layer Perceptrons (MLPs) (Bartlett & Maass, 2003; Sontag et al., 1998), Recurrent Neural Networks (RNNs) (Koiran & Sontag, 1998) and Recursive Neural Networks (Scarselli et al., 2018). In Scarselli et al. (2018), bounds on the VC dimension of the earliest GNN model with Pfaffian activation function are provided as well, while, in Esser et al. (2021), GCNs with linear and ReLU activation functions are considered. Our contribution extends such results to generic GNNs described by Eq. (1) and is particularly related to the work in Morris et al. (2023), where bounds for the VC dimension of modern GNNs are studied, when the activation function is a piecewise linear polynomial function. Bounds are derived also in terms of the number of colors computed by the 1–WL test on the graph domain. However, aside from Scarselli et al. (2018), all the aforementioned works focus solely on specific GNN models with piecewise polynomials activation functions, not considering common activation functions as arctangent, hyperbolic tangent or sigmoid.

Pfaffian functions —. Pfaffian functions have been first introduced in Khovanski (1991) to extend Bezout's classic theorem, which states that the number of complex solutions of a set of polynomial equations can be estimated based on their degree. The theory of Pfaffian functions has been exploited initially in Karpinski and Macintyre (1997) to characterize the bounds of the VC dimension of neural networks. Similarly, in Scarselli et al. (2018), the same approach is used to provide the aforementioned bounds. Pfaffian functions have also proven useful for providing insights into the topological complexity of neural networks and the impact of their depth (Bianchini & Scarselli, 2014).

3. Notation and basic concepts

In this section we introduce the notation used throughout the paper and the main basic concepts necessary to understand its content. Table 2 lists the main notation used in this work.

Graphs —. An *unattributed graph* G can be defined as a pair (V, E), where V is the (finite) set of *nodes* and $E \subseteq V \times V$ is the set of *edges* between nodes. A graph can be defined by its *adjacency matrix* $\mathbf{A}$, where $A_{ij} = 1$ if $e_{ij} = (i, j) \in E$, otherwise $A_{ij} = 0$. The *neighborhood* of a node v is represented by $\mathsf{ne}(v) = \{u \in V | (u, v) \in E\}$. A graph G is said

Table 2
Main notation.

Symbol	Meaning
G	Graph
V	Set of nodes
E	Set of edges
$\mathbf{A}$	Adjacency matrix
$\mathsf{ne}(v)$	Neighborhood of node v
Σ	Color set
HASH_0, HASH	Hashing functions
$\{\!\{\cdot\}\!\}$	Multiset
$\mathbf{h}_v$	Hidden feature of a node v
GNN	Function implementing a GNN
$\mathsf{COMBINE}^{(t+1)}(\cdot,\cdot)$	Combine function at layer $t+1$
$\mathsf{AGGREGATE}^{(t+1)}(\{\!\{\cdot\}\!\})$	Aggregate function at layer $t+1$
READOUT	Readout function of a GNN
σ	Element-wise activation function
VCdim	VC dimension
θ	Set of learnable parameters of a GNN
p	Number of learnable parameters of a GNN
L	Number of layers of a GNN
d	Hidden dimension of a GNN layer
format	Format of a Pfaffian function

to be *undirected* if it is assumed that $(v,u)=(u,v)$ (and therefore its adjacency matrix is symmetric), *directed* otherwise. A graph is said to be *node–attributed* or *labeled* if there exists a map $\boldsymbol{\alpha}: V\to\mathbb{R}^q$ that assigns to every $v\in V$ a *node attribute* (or *label*) $\boldsymbol{\alpha}(v)\in\mathbb{R}^q$. In this case, the graph can be defined as a triple $(V,E,\boldsymbol{\alpha})$.

The 1–WL test —. The *1st order Weisfeiler–Lehman test* (briefly, the *1–WL test*) is a test for graph isomorphism, based on the so–called *color refinement* procedure. Given two graphs $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$, in a finite graph domain $\mathcal{G}$, we perform the following steps.

- At initialization, we assign a color $c^{(0)}(v)$ to each node $v\in V_1\cup V_2$. Formally, in the case of attributed graphs, we can define the color initialization as

  $$c^{(0)}(v)=\mathsf{HASH}_0(\boldsymbol{\alpha}(v)),$$

  where $\mathsf{HASH}_0:\mathbb{R}^q\to\Sigma$ is a function that codes bijectively node attributes to colors. In case of unattributed graphs, the initialization is uniform, and each node v gets the same color $c^{(0)}(v)$.
- For $t>0$, we update the color of each node in parallel on each graph by the following updating scheme

  $$c_v^{(t)}=\mathsf{HASH}(c_v^{(t-1)},\{\!\{c_u^{(t-1)}|u\in\mathsf{ne}[v]\}\!\}),\quad\forall v\in V_1\cup V_2,$$

  where $\{\!\{\cdot\}\!\}$ denotes a multiset and $\mathsf{HASH}:\Sigma\times\Sigma^*\to\Sigma$ is a function mapping bijectively a pair (color, color multiset) to a single color.

To test whether the two graphs G_1 and G_2 are isomorphic or not, the set of colors of the nodes of G_1 and G_2 are compared step by step; if there exist an iteration t such that the colors are different, namely $c^{(t)}_{G_1}:=\{\!\{c^{(t)}(v)\mid v\in V_1\}\!\}$ is different from $c^{(t)}_{G_2}$, the graphs are declared as non-isomorphic. When no difference is detected, the procedure halts as soon as the node partition defined by the colors becomes stable. It has been proven (Kiefer, 2020) that $|V|-1$ iterations are sufficient, and sometimes necessary, to complete the procedure. Moreover, the color refinement procedure can be used also to test whether two nodes are isomorphic or not. Intuitively, two nodes are isomorphic if their neighborhoods (of any order) are equal; such an isomorphism can be tested by comparing the node colors at any step of the 1–WL test. In Krebs and Verbitsky (2015), it has been proven that for node isomorphism up to $2\max(|V_1|,|V_2|)-1$ refinement steps may be required.

We would like to mention two important results that demonstrate the equivalence between GNNs and the 1–WL test in terms of their expressive power. The first result was established in Xu et al. (2019) and characterizes the equivalence of GNNs and the 1–WL test on a graph-level task. This equivalence is based on GNNs with generic message passing layers that satisfy certain conditions. Another characterization is due to Morris et al. (2019) and states the equivalence on a node coloring level, referring to the particular model defined by Eq. (2).

Graph neural networks (GNNs) —. Graph Neural Networks are a class of machine learning models suitable for processing structured data in the form of graphs. At a high level, we can formalize a GNN as a function $\mathsf{GNN}:\mathcal{G}\to\mathbb{R}^r$, where $\mathcal{G}$ is a set of node–attributed graphs and r is the dimension of the output, which depends on the type of task to be carried out; in our setting, we will assume that $r=1$. Intuitively, a GNN learns how to represent the nodes of a graph by vectorial representations, called *hidden features*, giving an encoding of the information stored in the graph The hidden feature $\mathbf{h}_v$ of a node v is, at the beginning, set equal to node attributes, i.e., $\mathbf{h}_v^{(0)}=\boldsymbol{\alpha}(v)$. Then, the features are updated according to the following scheme

$$\mathbf{h}_v^{(t+1)}=\mathsf{COMBINE}^{(t+1)}\big(\mathbf{h}_v^{(t)},\mathsf{AGGREGATE}^{(t+1)}(\{\!\{\mathbf{h}_u^{(t)}|u\in\mathsf{ne}(v)\}\!\})\big),\tag{1}$$

for all $v\in V$ and $t=0,\dots L-1$, where $\mathbf{h}_v^{(t)}$ is the hidden feature of node v at time t, L is the number of layers of the GNN. Here, $\{\mathsf{COMBINE}^{(t)}\}_{t=1,\dots,L}$ and $\{\mathsf{AGGREGATE}^{(t)}\}_{t=1,\dots,L}$ are functions that can be defined by learning from examples. Popular GNN models like GraphSAGE (Hamilton, Ying, & Leskovec, 2017), GCNs (Kipf & Welling, 2016), and Graph Isomorphism Networks (Xu et al., 2019) are based on this updating scheme. The output o is produced by a READOUT function, which, in graph-focused tasks, takes in input the features of all the nodes, i.e. $o=\mathsf{READOUT}(\{\!\{\mathbf{h}_u^{(L)}|u\in V\}\!\})$, while, in node-focused tasks, is calculated on each node, i.e., $o_v=\mathsf{READOUT}(\mathbf{h}_v^{(L)})$, $\forall v\in V$.

For simplicity, in the following we will assume that $\mathsf{COMBINE}^{(1)}$ has $p_{\mathsf{comb}^{(1)}}$ parameters and for every $t=2,\dots,L$ the number of parameters of $\mathsf{COMBINE}^{(t)}$ is the same, and we denote it as p_{comb}. The same holds for $\mathsf{AGGREGATE}^{(t)}$ and READOUT, with the number of parameters denoted respectively as p_{agg} and p_{read}. Thus, the total number of parameters in a GNN defined as in Eq. (1) is $\bar{p}=p_{\mathsf{comb}^{(1)}}+p_{\mathsf{agg}^{(1)}}+(L-1)(p_{\mathsf{comb}}+p_{\mathsf{agg}})+p_{\mathsf{read}}$.

For our analysis, following Morris et al. (2019), we also consider a simpler computational framework, which has been proven to match the expressive power of the Weisfeiler–Lehman test (Morris et al., 2019), and is general enough to be similar to many GNN models. In such a framework, the hidden feature $\mathbf{h}_v^{(t+1)}\in\mathbb{R}^d$ at the message passing iteration $t+1$ is defined as

$$\mathbf{h}_v^{(t+1)}=\boldsymbol{\sigma}\big(\mathbf{W}_{\mathsf{comb}}^{(t+1)}\mathbf{h}_v^{(t)}+\mathbf{W}_{\mathsf{agg}}^{(t+1)}\mathbf{h}_{\mathsf{ne}(v)}^{(t)}+\mathbf{b}^{(t+1)}\big),\tag{2}$$

where $\mathbf{h}_{\mathsf{ne}(v)}^{(t)}=\mathsf{POOL}\big(\{\!\{\mathbf{h}_u^{(t)}|u\in\mathsf{ne}(v)\}\!\}\big)$, $\boldsymbol{\sigma}:\mathbb{R}^d\to\mathbb{R}^d$ is an element–wise activation function, and POOL is the aggregating operator on the features of neighboring nodes,

$$\mathsf{POOL}\big(\{\!\{\mathbf{h}_u^{(t)}|u\in\mathsf{ne}(v)\}\!\}\big)=\sum_{u\in\mathsf{ne}(v)}\mathbf{h}_u^{(t)}.$$

With respect to Eq. (1), we have that $\mathsf{AGGREGATE}^{(t)}(\cdot)=\mathsf{POOL}(\cdot)$ $\forall t=1,\dots,L$, while $\mathsf{COMBINE}^{(t+1)}(\mathbf{h}_v,\mathbf{h}_{\mathsf{ne}(v)})=\boldsymbol{\sigma}\big(\mathbf{W}_{\mathsf{comb}}^{(t+1)}\mathbf{h}_v+\mathbf{W}_{\mathsf{agg}}^{(t+1)}\mathbf{h}_{\mathsf{ne}(v)}+\mathbf{b}^{(t+1)}\big)$. In this case, the READOUT function for graph classification tasks can be defined as

$$\mathsf{READOUT}\Big(\{\!\{\mathbf{h}_v^{(L)}\mid v\in V\}\!\}\Big):=f\Big(\sum_{v\in V}\mathbf{w}\mathbf{h}_v^{(L)}+b\Big).\tag{3}$$

For each node, the hidden state is initialized as $\mathbf{h}_v^{(0)}=\boldsymbol{\alpha}(v)\in\mathbb{R}^q$. The learnable parameters of the GNN can be summarized as

$$\boldsymbol{\theta}:=(\mathbf{W}_{\mathsf{comb}}^{(1)},\mathbf{W}_{\mathsf{agg}}^{(1)},\mathbf{b}^{(1)},\mathbf{W}_{\mathsf{comb}}^{(2)},\mathbf{W}_{\mathsf{agg}}^{(2)},\mathbf{b}^{(2)},\dots,\mathbf{W}_{\mathsf{comb}}^{(L)},\mathbf{W}_{\mathsf{agg}}^{(L)},\mathbf{b}^{(L)},\mathbf{w},b),$$

with $\mathbf{W}_{\mathsf{comb}}^{(1)},\mathbf{W}_{\mathsf{agg}}^{(1)}\in\mathbb{R}^{d\times q}$, $\mathbf{W}_{\mathsf{comb}}^{(t)},\mathbf{W}_{\mathsf{agg}}^{(t)}\in\mathbb{R}^{d\times d}$, for $t=2,\dots,L$, $\mathbf{b}^{(t)}\in\mathbb{R}^{d\times 1}$, for $t=2,\dots,L$, $\mathbf{w}\in\mathbb{R}^{1\times d}$, and $b\in\mathbb{R}$.

VC dimension —. The VC dimension is a measure of complexity of a set of hypotheses, which can be used to bound the empirical error of machine learning models. Formally, a binary classifier $\mathcal{L}$ with parameters θ is said to *shatter* a set of patterns $\{\mathbf{x}_1, \dots, \mathbf{x}_m\} \subseteq \mathbb{R}^q$ if, for any binary labeling of the examples $\{y_i\}_{i=1,\dots,m}$, $y_i \in \{0, 1\}$, there exists θ s.t. the model $\mathcal{L}$ correctly classifies all the patterns, i.e. $\sum_{i=0}^{n} |\mathcal{L}(\theta, \mathbf{x}_i) - y_i| = 0$. The *VC dimension* of the model $\mathcal{L}$ is the dimension of the largest set that $\mathcal{L}$ can shatter.

The VC dimension is linked with the generalization capability of machine learning models. Actually, given a training and a test set for the classifier $\mathcal{L}$, whose patterns are i.i.d. samples extracted from the same distribution, the VC dimension allows to compute a bound, in probability, for the difference between the training and test error (Vapnik & Chervonenkis, 1971):

$$\Pr\left(\mathrm{E}_{\text{test}} \leq \mathrm{E}_{\text{training}} + \sqrt{\frac{1}{N}\left[\mathsf{VCdim}\left(\log\left(\frac{2m}{\mathsf{VCdim}}\right) + 1\right) - \log\left(\frac{\eta}{4}\right)\right]}\right) = 1 - \eta \tag{4}$$

for any $\eta > 0$, where E_{test} is the test error, $\mathrm{E}_{\text{training}}$ is the training error, m is the size of the training dataset and VCdim is the VC dimension of $\mathcal{L}$.

Pfaffian functions —. A Pfaffian chain of order $\ell \geq 0$ and degree $\alpha \geq 1$, in an open domain $U \subseteq \mathbb{R}^n$, is a sequence of analytic functions $f_1, f_2, \dots, f_\ell$ over U, satisfying the differential equations

$$df_j(\mathbf{x}) = \sum_{1 \leq i \leq n} g_{ij}(\mathbf{x}, f_1(\mathbf{x}), \dots, f_j(\mathbf{x}))dx_i, \quad 1 \leq j \leq \ell.$$

Here, $g_{ij}(\mathbf{x}, y_1, \dots, y_j)$ are polynomials in $\mathbf{x} \in U$ and $y_1, \dots, y_j \in \mathbb{R}$ of degree not exceeding α. A function $f(\mathbf{x}) = P(\mathbf{x}, f_1(\mathbf{x}), \dots, f_\ell(\mathbf{x}))$, where $P(\mathbf{x}, y_1, \dots, y_\ell)$ is a polynomial of degree not exceeding β, is called a *Pfaffian function of format* (α, β, ℓ). Pfaffian maps are a large class of functions that include most of the functions with continuous derivatives used in practical applications (Khovanski, 1991). For instance, the logistic sigmoid $\mathsf{logsig} = \frac{1}{1+e^{-x}}$ has derivative $\mathsf{logsig}'(x) = \mathsf{logsig}(x)(1 - \mathsf{logsig}(x))$. Therefore, the chain of logsig is just (logsig), the derivative can be expressed as a polynomial of order 2, and the function itself is a polynomial of order 1 in the functions belonging to the chain, which means that $\mathtt{format}(\mathsf{logsig}) = (2, 1, 1)$. Moreover, the arctangent atan and the hyperbolic tangent tanh are Pfaffian functions, with formats $\mathtt{format}(\mathsf{atan}) = (3, 1, 2)$ and $\mathtt{format}(\mathsf{tanh}) = (2, 1, 1)$, respectively.

4. Theoretical results

In this section we report the main results on the VC dimension of GNNs with Pfaffian activation functions. The proofs can be found in Appendix A.

4.1. Bounds based on the network hyperparameters

Our main result provides a bound on the VC dimension of GNNs in which $\mathsf{COMBINE}^{(t)}$, $\mathsf{AGGREGATE}^{(t)}$ and READOUT are Pfaffian functions. More precisely, we consider a slightly more general version of the GNN model in Eq. (1), where the updating scheme is

$$\mathbf{h}_v^{(t+1)} = \mathsf{COMBINE}^{(t+1)}\left(\mathbf{h}_v^{(t)}, \mathsf{AGGREGATE}^{(t+1)}(\{\!\{\mathbf{h}_u^{(t)} | u \in V\}\!\}, A_v)\right), \tag{5}$$

and A_v is the v–th column of the connectivity matrix, which represents the neighborhood of v. The advantage of the model in Eq. (5) is that it makes explicit the dependence of $\mathsf{AGGREGATE}^{(t)}$ on the graph connectivity. Actually, here we want to underline what the inputs of $\mathsf{AGGREGATE}^{(t)}$ are to clarify and make formally precise the assumptions that those functions are Pfaffian and have a given format.

Our result provides a bound on the VC dimension w.r.t. the total number $\bar{p}$ of parameters, the number of computation units H, the number of layers L, the feature dimension d, the maximum number N of nodes in a graph, and the attribute dimension q. Here, we assume that GNN computation units include the neurons computing the hidden features of each node and the outputs. Therefore, there is a computation unit for each component of a feature, each layer, each node of the input graph and a further computation unit for the READOUT.

Theorem 1. *Let us consider the GNN model described by Eq. (5). If* $\mathsf{COMBINE}^{(t)}$, $\mathsf{AGGREGATE}^{(t)}$ *and* READOUT *are Pfaffian functions with format* $(\alpha_{\mathsf{comb}}, \beta_{\mathsf{comb}}, \ell_{\mathsf{comb}})$, $(\alpha_{\mathsf{agg}}, \beta_{\mathsf{agg}}, \ell_{\mathsf{agg}})$, $(\alpha_{\mathsf{read}}, \beta_{\mathsf{read}}, \ell_{\mathsf{read}})$, *respectively, then the VC dimension satisfies*

$$\mathsf{VCdim}(\mathsf{GNN}) \leq 2 \log B + \bar{p}(16 + 2 \log \bar{s}) \tag{6}$$

where $B \leq 2^{\frac{\bar{\ell}(\bar{\ell}-1)}{2}+1}(\bar{\alpha} + 2\bar{\beta} - 1)^{\bar{p}-1}((2\bar{p} - 1)(\bar{\alpha} + \bar{\beta}) - 2\bar{p} + 2)^{\bar{\ell}}$, $\bar{\alpha} = \max\{\alpha_{\mathsf{agg}} + \beta_{\mathsf{agg}} - 1 + \alpha_{\mathsf{comb}}\beta_{\mathsf{agg}}, \alpha_{\mathsf{read}}\}$, $\bar{\beta} = \max\{\beta_{\mathsf{comb}}, \beta_{\mathsf{read}}\}$,

$\bar{p} = p_{\mathsf{comb}^{(0)}} + p_{\mathsf{agg}^{(0)}} + (L - 1)(p_{\mathsf{comb}} + p_{\mathsf{agg}}) + p_{\mathsf{read}}$, $\bar{\ell} = \bar{p}H$, $H = LNd(\ell_{\mathsf{comb}} + \ell_{\mathsf{agg}}) + \ell_{\mathsf{read}}$ *and* $\bar{s} = LNd + Nq + 1$ *hold. By substituting the definitions in Eq. (6), we obtain*

$$\begin{aligned}\mathsf{VCdim}(\mathsf{GNN}) \leq\ & \bar{p}^2(LNd(\ell_{\mathsf{comb}} + \ell_{\mathsf{agg}}) + \ell_{\mathsf{read}})^2 \\ &+ 2\bar{p}\log(3\gamma) \\ &+ 2\bar{p}\log((4\gamma - 2)\bar{p} + 2 - 2\gamma) \\ &+ \bar{p}(16 + 2\log(LNd + Nq + 1))\end{aligned} \tag{7}$$

where $\bar{\alpha}, \bar{\beta} \leq \gamma$ *for a constant* $\gamma \in \mathbb{R}$.

By inspecting the bound, we observe that the dominant term is $\bar{p}^2H^2 = \bar{p}^2(LNd(\ell_{\mathsf{comb}} + \ell_{\mathsf{agg}}) + \ell_{\mathsf{read}})^2$. Thus, Theorem 1 suggests that the VC dimension is $O(\bar{p}^2L^2N^2d^2)$, w.r.t. the number of parameters $\bar{p}$ of the GNN, the number of layers L, the number N of graph nodes, and the feature dimension d. Notice that those hyperparameters are related by constraints, which should be considered in order to understand how the VC dimension depends on each of them. Therefore, the VC dimension is at most $O(p^4)$ since, as the number p of parameters grows, the number of layers L and/or the feature dimension d also increases.

Interestingly, such a result is similar to those already obtained for feedforward and recurrent neural networks with Pfaffian activation functions. Table 3 compares our result with those available in the literature, highlighting that, even if GNNs have a more complex structure, the growth rate of the VC dimension, depending on the hyperparameters, is the same as the simpler models.

The following theorem provides more details and clarifies how the VC dimension depends on each hyperparameter.

Theorem 2. *Let* $\mathsf{COMBINE}^{(t)}$, $\mathsf{AGGREGATE}^{(t)}$ *and* READOUT *be the Pfaffian functions defined in Theorem 1. If* $p_{\mathsf{comb}}, p_{\mathsf{aggr}}, p_{\mathsf{read}} \in \mathcal{O}(d)$, *then the VC dimension of a GNN defined as in Eq. (1), w.r.t.* $\bar{p}, N, L, d, q$ *satisfies*

$$\begin{aligned}&\mathsf{VCdim}(\mathsf{GNN}) \leq \mathcal{O}(\bar{p}^4) \\ &\mathsf{VCdim}(\mathsf{GNN}) \leq \mathcal{O}(N^2) \\ &\mathsf{VCdim}(\mathsf{GNN}) \leq \mathcal{O}(L^4) \\ &\mathsf{VCdim}(\mathsf{GNN}) \leq \mathcal{O}(d^6) \\ &\mathsf{VCdim}(\mathsf{GNN}) \leq \mathcal{O}(q^2) \quad \square\end{aligned}$$

The proof of Theorem 1 adopts the same reasoning used in Karpinski and Macintyre (1997) to derive a bound on the VC dimension of feedforward neural networks with Pfaffian activation functions, and used in Scarselli et al. (2018) to provide a bound for the first GNN model. Intuitively, the proof is based on the following steps: it is shown that the computation of the GNNs on graphs can be represented by a set of equations defined by Pfaffian functions with format $(\bar{\alpha}, \bar{\beta}, \bar{\ell})$, where $\bar{\alpha}, \bar{\beta}, \bar{\ell}$ are those defined in the theorem; then, the bound is obtained exploiting a result in Karpinski and Macintyre (1997) that associates the VC dimension to the number of connected components in the inverse image of a system of Pfaffian equations. Finally, a result in Gabrielov and Vorobjov (2004) allows to estimate the required number of connected components. Note that our bound and other bounds obtained

Table 3
Upper bounds on the VC dimension of common architectures, where p is the number of network parameters, N the number of nodes in the input graph or sequence, and C the maximum number of colors per graph.

Activation function	Bound	References
Modern GNNs		
Piecewise polynomial	$O(p\log(Cp)+p\log(N))$	Morris et al. (2023)
tanh, logsig or atan	$O(p^4N^2)$	**This work**
tanh, logsig or atan	$O(p^4C^2)$	**This work**
Original GNNs (Scarselli et al., 2009)		
Polynomial	$O(p\log(N))$	Scarselli et al. (2018)
Piecewise polynomial	$O(p^2N\log(N))$	Scarselli et al. (2018)
tanh, logsig or atan	$O(p^4N^2)$	Scarselli et al. (2018)
Positional RecNNs		
Polynomial	$O(pN)$	Hammer (2001)
logsig	$O(p^4N^2)$	Hammer (2001)
Recurrent Neural Networks		
Polynomial	$O(pN)$	Koiran and Sontag (1997)
Piecewise polynomial	$O(p^2N)$	Koiran and Sontag (1997)
tanh or logsig	$O(p^4N^2)$	Koiran and Sontag (1997)
Multilayer Networks		
Binary	$O(p\log p)$	Baum and Haussler (1988), Maass (1994), Sakurai (1995)
Polynomial	$O(p\log p)$	(Goldberg & Jerrum, 1993)
Piecewise polynomial	$O(p^2)$	Goldberg and Jerrum (1993), Koiran and Sontag (1997)
tanh, logsig or atan	$O(p^4)$	Karpinski and Macintyre (1997)

for networks with Pfaffian activation functions are larger than those for networks with simpler activations. As explained in Karpinski and Macintyre (1997) Scarselli et al. (2018), such a difference is likely due to the current limitations of mathematics in this field, which makes tight bounds more difficult to achieve with Pfaffian functions.

We now specifically derive bounds for the VC dimension for the architecture described by Eqs. (2)–(3).

Theorem 3. *Let us consider the GNN model described by Eqs. (2)–(3). If σ is a Pfaffian function in $\mathbf{x}$ with format $(\alpha_\sigma,\beta_\sigma,\ell_\sigma)$, then the VC dimension satisfies*

$$\mathsf{VCdim}\big(\mathsf{GNN}\big) \leq 2\log B + \bar{p}(16+2\log\bar{s}),$$

where $B \leq 2^{\frac{\bar{\ell}(\bar{\ell}-1)}{2}+1}(\bar{\alpha}+2\bar{\beta}-1)^{\bar{p}-1}((2\bar{p}-1)(\bar{\alpha}+\bar{\beta})-2\bar{p}+2)^{\bar{\ell}}$, $\bar{\alpha}=2+3\alpha_\sigma$, $\bar{\beta}=\beta_\sigma$, $\bar{\ell}=\bar{p}H\ell_\sigma$, *and* $\bar{s}=LNd+Nq+1$ *hold.*

In particular, if σ is the logistic sigmoid activation function, we have

$$\mathsf{VCdim}\big(\mathsf{GNN}\big) \leq \bar{p}^2H^2 + 2\bar{p}\log(9) + 2\bar{p}H\log(16\bar{p}) + \bar{p}(16+2\log(\bar{s})). \quad \square$$

The proof of Theorem 3 can be found in Appendix A.

Interestingly, the bounds on the VC dimension that can be derived from Theorem 3, w.r.t the hyperparameters, turn out to be the same derived in Theorem 2. Thus, even if the considered model is simpler, those bounds do not change.

4.2. Bounds based on the number of the 1–WL colors

The developed theory is also easily applied to the case when nodes are grouped according to their colors defined by the Weisfeiler–Lehman algorithm. Intuitively, since GNNs produce the same features on group of nodes with the same color, the computation can be simplified by considering each group as a single entity. As consequence, the bounds on VC dimension can be tightened by using colors in place of nodes. Formally, for a given graph G, let $C_1(G)=\sum_{i=1}^{T}C^t(G)$ be the number of colors generated by the 1–WL test, where $C^t(G)$ is the number of colors at step $t>0$. Moreover, let us assume that $C^t(G)$ is bounded, namely there exists C_1 such that $C_1(G)\leq C_1$ for all the graphs G in the domain $\mathcal{G}$. The following theorem provides a bound on the VC dimension w.r.t. the number of colors produced by the 1–WL test.

Theorem 4. *Let us consider the GNN model described by Eqs. (2)–(3) using the logistic sigmoid* logsig *as the activation function. Assume a subset $\mathcal{S}\subseteq\mathcal{G}$. The VC dimension of the GNN satisfies*

$$\mathsf{VCdim}\big(\mathsf{GNN}(C_1)\big) \leq \mathcal{O}(C_1^2)$$
$$\mathsf{VCdim}\big(\mathsf{GNN}(C_0)\big) \leq \mathcal{O}(\log(C_0)) \quad \square$$

The theorem suggests that the VC dimension depends quadratically on the total number of node colors and logarithmically on the initial number of colors. Actually, a GNN processes all the nodes of a graph at the same time and the GNN architecture is similar to a feedforward network where some computation units are replicated at each node. Thus, the complexity of the GNN grows with the number of nodes and this explains the dependence of the VC dimension on the number of nodes (see Theorem 2). On the other hand, nodes with the same colors cannot be distinguished by the GNN: this means that, in theory, we can use the same computation units for a group of nodes sharing the color. Therefore, to get tighter bounds on the VC dimension, we can consider the number of colors in place of the number of nodes.

Finally, it is worth mentioning that, the presented theorems suggest that GNNs may have a worst generation capability when the domain is composed by graphs with many different colors. This happens because when the number of colors in each graph increases, the VC dimension increases as well. On the other hand, the generalization capability benefits from a large total number of colors in the training set. Actually, generalization depends not only on VC dimension, but, obviously, also on the number of patterns in training set (see Eq. (4)). In GNN graph–focused tasks, graphs play the role of patterns, where we count only the graphs with different colors, as those with the same colors are just copies of the same pattern. A similar reasoning applies to node–focused tasks, by counting the total number of nodes with different colors in the training set.

5. Experimental validation

In this section, we present an experimental validation of our theoretical results. We will show how the VC dimension of GNNs, described in Eqs. (2)–(3), changes as the hyperparameters vary, respecting the bounds found in Theorems 2 and 4.

5.1. Experimental setting

We design two experiments to assess the validity, respectively, of Theorems 2 and 4. In both cases, we train a Graph Neural Network,

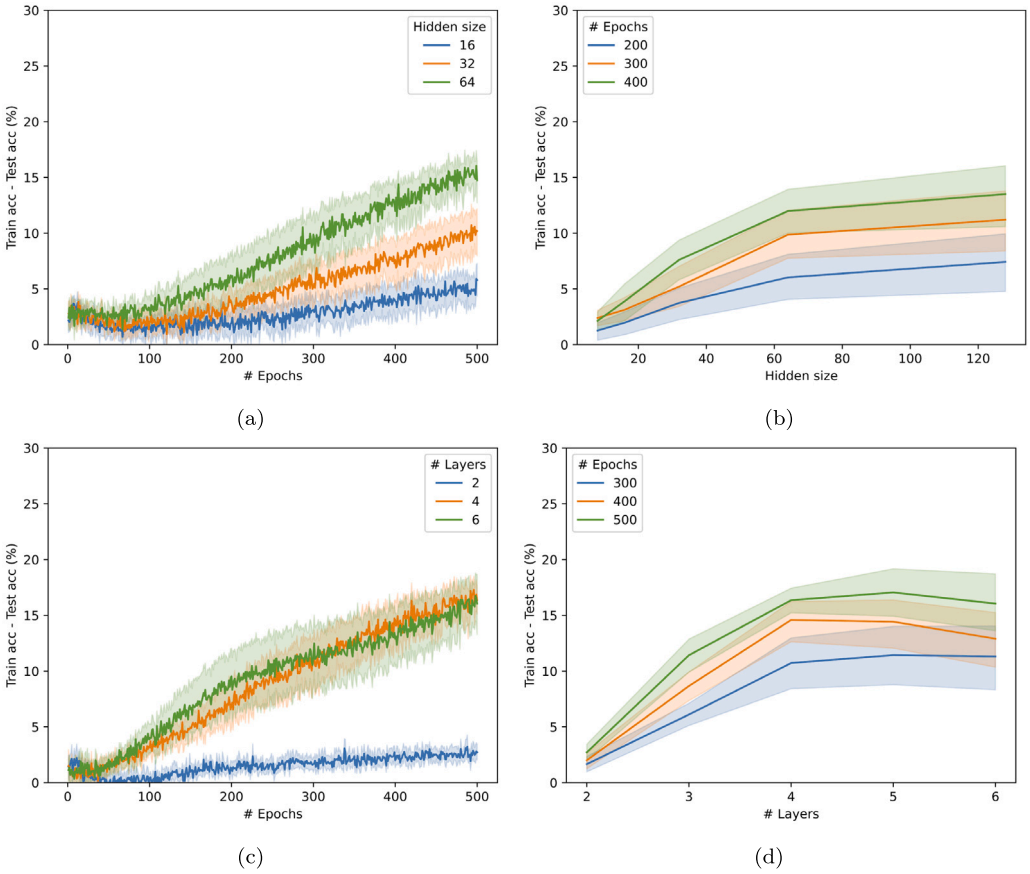


Fig. 1. Results on the task **E1** for GNNs with activation function arctan.

Table 4
Statistics on the benchmark datasets used for **E1**.

Dataset	# Graphs	# Classes	Avg. # Nodes	Avg. # Edges
PROTEINS	1113	2	39.06	72.82
NCI1	4110	2	29.87	32.30
PTC-MR	344	2	14.29	14.69

Table 5
Summary of the parameters for each split of the ordered **NCI1** dataset in task **E2**.

	Split 1	Split 2	Split 3	Split 4
# Nodes	27 667	30 591	31 763	32 673
# Colors	26 243	26 569	24 489	16 348
$\min_G \frac{\#\text{Nodes}(G)}{\#\text{Colors}(G)}$	1.000	1.105	1.208	1.437
$\max_G \frac{\#\text{Nodes}(G)}{\#\text{Colors}(G)}$	1.105	1.208	1.437	8

composed by message passing layers, defined as in Eq. (2), where the activation function σ is arctan or tanh; the final READOUT layer is an affine layer with $\mathbf{W}_{\text{out}} \in \mathbb{R}^{1\times \text{hd}}$, after which a logsig activation function is applied. The expressive power of this model has been proven to be equivalent to the 1–WL test and, therefore, such a model is a good representative of the class of message passing GNNs. The model is trained via Adam optimizer with an initial learning rate $\lambda = 10^{-3}$. The hidden feature size is denoted by hd and the number of layers by l.

E1: We measure the evolution of the difference between the training accuracy and the validation accuracy, diff = training_acc − test_acc, through the training epochs, over three different datasets taken from the TUDataset repository (Morris et al., 2020). In particular, **PROTEINS** (Borgwardt et al., 2005) is a dataset of proteins represented as graphs which contains both enzymes and non-enzymes; **NCI1** (Wale, Watson, & Karypis, 2008) is a dataset of molecules relative to anti-cancer screens where the chemicals are assessed as positive or negative to cell lung cancer; finally, **PTC-MR** (Helma, King, Kramer, & Srinivasan, 2001) is a collection of chemical compounds represented as graphs which report the carcinogenicity for rats. The choice of the datasets has been driven by their binary classification nature. Their statistics are summarized in Table 4. In the experiments, firstly, we fix the hidden feature size to hd = 32 and let the number of layers vary in the range $l \in [2, 3, 4, 5, 6]$, to measure how diff evolves. Subsequently, we fix the number of layers to $l = 3$ and let the hidden feature size vary in the range, hd $\in [8, 16, 32, 64, 128]$, to perform the same task. We train the model for 500 epochs in each run, with the batch size set to 32.

E2: We measure the evolution of the difference between the training accuracy and the validation accuracy, diff, through the training epochs over the dataset **NCI1**, whose graphs are increasingly ordered according to the ratio $\frac{|V(G)|}{C^l(G)}$ and split in four different groups. The intuition here is that, being the number of graph nodes bounded, splitting the ordered dataset as described above, should provide four datasets in which the total number of colors is progressively increasing. The hidden size is fixed at hd = 16, the number of layers is $l = 4$, the batch size is fixed equal to 32. In Table 5, we report the reference values for each split. We train the model for 2000 epochs (the number of epochs is greater than in the **E1** task because **E2** shows greater instability during training).

Each experiment is statistically evaluated over 10 runs. The overall training is performed on an Intel(R) Core(TM) i7-9800X processor running at 3.80 GHz, using 31 GB of RAM and a GeForce GTX 1080

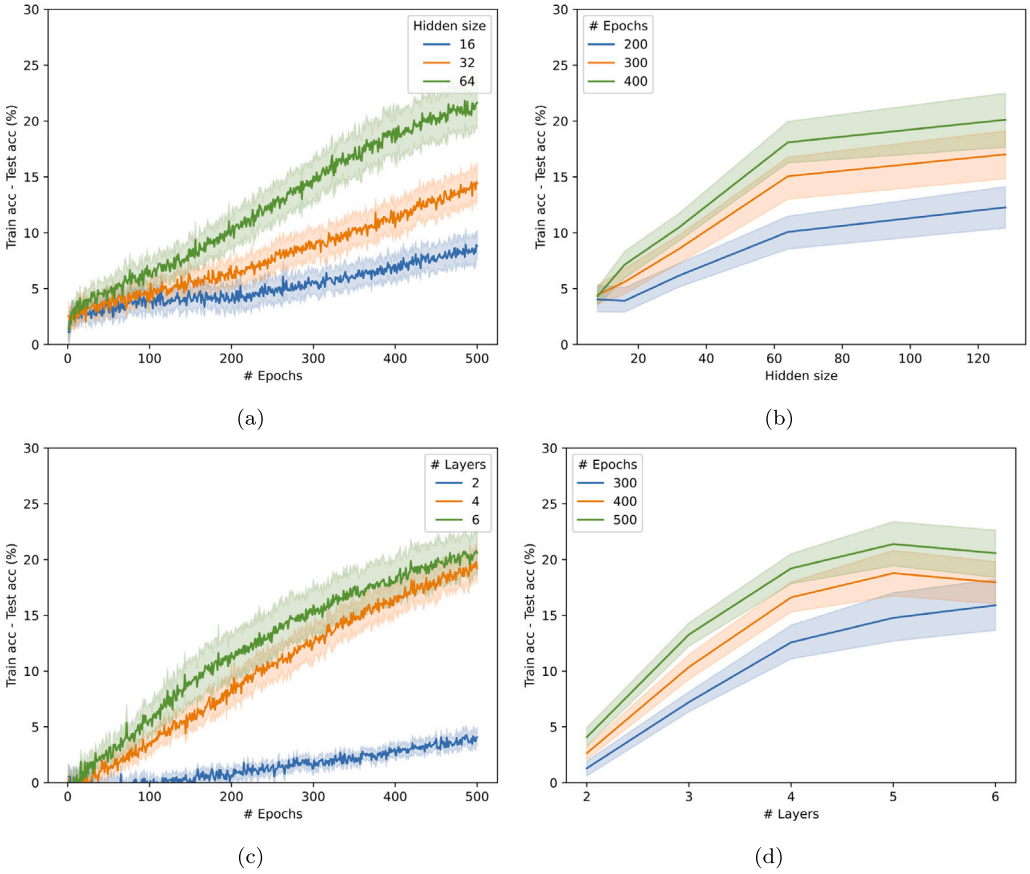


Fig. 2. Results on the task **E1** for GNNs with activation function tanh.

Ti GPU unit. The code developed to run the experiments exploits the Python package Pytorch Geometric and can be found at https://github.com/AleDinve/vc-dim-gnn.

5.2. Experimental results

Task E1 —. Numerical results for the **NCI1** dataset are reported in Figs. 1 and 2, for arctan and tanh activations, respectively, while results for **PROTEINS** and **PTC_MR** are reported in Appendix B. Figures show the mean values of diff as a solid colored line and the 95% confidence interval as a shaded area, to quantify the statistical variability of the results over the 10 runs. In particular, in both Figs. 1 and 2, the evolution of diff is shown as the number of epochs varies, for different values of hd keeping fixed $l = 3$ in (a), and for different values of l keeping fixed hd $= 32$ in (c), respectively. Moreover, the evolution of diff as the hidden size increases, for varying epochs, is shown in (b), while (d) depicts how diff evolves as the number of layers increases.

The behavior of the evolution of diff proves to be consistent with the bounds provided by Theorem 2 with respect to increasing the hidden dimension or the number of layers. Although it is hard to establish a precise function that links the VC dimension to diff, given also the complex nature of Pfaffian functions, we can partially rely on Eq. (4) (which is valid for large sample sets) to argue that our bounds are verified by this experimental setting.

It is interesting to notice that diff tends to increase faster with the hidden size over the dataset **PTC-MR** than over the dataset **PROTEINS**. This could be due to the different average number of nodes in the two datasets; indeed, if we fix the architecture, the GNN could be less prone to overfit (and then to poor generalization capabilities) as the graphs in the dataset become bigger.

Task E2 —. Numerical results for this task on the **NCI1** dataset are reported in Fig. 3, considering the tanh activation function. In particular, the evolution of diff is shown in (a) as the number of epochs varies, for different values of $\frac{V(G)}{C^l(G)}$, keeping fixed $l = 4$ and hd $= 16$; instead, the evolution of diff as the ratio $\frac{V(G)}{C^l(G)}$ increases is depicted in (b), for the number of epochs varying in $\{1000, 1500, 2000\}$.

Similar observations as for the experimental setting **E1** can be drawn here: indeed, the evolution of diff in our experiment is consistent with the bounds presented in Theorem 4, as the ratio between colors and nodes increases.

6. Discussion

In this work we derived new bounds for the VC dimension of modern message passing GNNs with Pfaffian activation functions, closing the gap present in the literature with respect to many common used activation functions; furthermore, we propose a preliminary experimental validation to demonstrate the coherence between theory and practice. Such a characterization allows to compare different GNN models and to find the appropriate tradeoff between the expressivity of those models and their generalization capabilities.

The bounds derived in this work can also be straightforwardly exploited to quantify an upper bound on the *Rademacher complexity* of a function F $\mathcal{R}(F)$; indeed, it holds that $\mathcal{R}(F) \leq C\sqrt{VC(F)}$ for some constant C (Gnecco, Sanguineti, et al., 2008). Nevertheless, other analytical tools should be used to obtain an exhaustive overview on the characterization of the Rademacher Complexity for message passing GNNs; we will further investigate this perspective in future works.

Different research perspectives can be envisaged to improve the results obtained: first, our analysis lacks the derivation of *lower bounds*, which could provide a more precise intuition of the degradation of

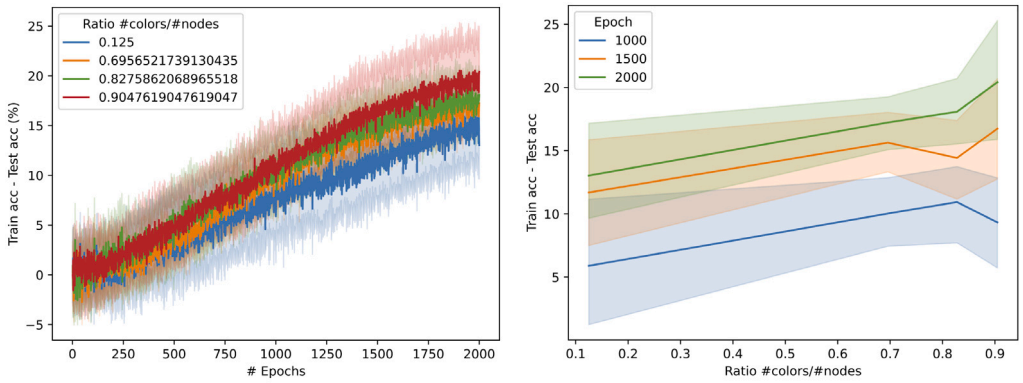


Fig. 3. Results on the task **E2** for GNNs with activation function tanh.

generalization capabilities for GNNs within the chosen architectural framework. In addition, providing a relationship between the VC dimension and the difference between the training and test accuracy would be much more informative; we could establish a quantitative measure with respect to the number of parameters that would allow us to better explain the experimental performance. Furthermore, the proposed analysis on the VC dimension deserves to be extended to other GNN paradigms, such as Graph Transformers (Yun, Jeong, Kim, Kang, & Kim, 2019) and Graph Diffusion Models (Zhang et al., 2023).

CRediT authorship contribution statement

Giuseppe Alessio D'Inverno: Writing – review & editing, Writing – original draft, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Monica Bianchini:** Writing – review & editing, Writing – original draft, Supervision, Project administration. **Franco Scarselli:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Giuseppe Alessio D'Inverno is member of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM). Monica Bianchini is partially supported by the PNRR Project "THE - Tuscany Health Ecosystem", CUP: B83C22003920001.

Appendix A. Proof of the main results

The proof of Theorems 1, 3 and 4 adopts the same reasoning used in Karpinski and Macintyre (1997) and Scarselli et al. (2018) to derive a bound on the VC dimension of feedforward neural networks and the original GNN model, respectively. Before proceeding with the proofs, let us introduce the required notation and some results from Karpinski and Macintyre (1997). These results will provide us with the mathematical tools to represent the computation of GNNs with a set of Pfaffian equations and to bound the VC dimension based on the format of the functions involved in such equations.

A.1. Notation and results from the literature

Representing a set of equations by a logical formula

Formally, we use a theory in which a classifier is described by a logical formula that is constructed by combining Pfaffian equations. Thus, let $\tau_1, \ldots, \tau_{\bar{s}}$ be a set of C^∞ (infinitely differentiable) functions from $\mathbb{R}^{\gamma+p}$ to $\mathbb{R}$. Suppose that $\Phi(\mathbf{y}, \theta), \mathbf{y} \in \mathbb{R}^\gamma, \theta \in \mathbb{R}^p$ is a quantifier-free logical formula constructed using the operators *and* and *or*, and atoms in the form of $\tau_i(\mathbf{y}, \theta) = 0$. Note that, fixed θ, $\Phi(\cdot, \theta)$ takes as input a vector $\mathbf{y}$ and returns a logical value, so that it can be considered as a classifier with input $\mathbf{y}$ and parameters θ. Moreover, the formula can be also used to represent a set of Pfaffian equations, which corresponds to the case when Φ includes only the operator *and* and τ_i are Pfaffian. Actually, later, we will see that $\tau_1, \ldots, \tau_{\bar{s}}$ can be specified so that Φ defines the computation of a GNN.

The VC dimension of Φ

The VC dimension of Φ can be defined in the usual way. Indeed, Φ is said to shatter a set $S = \{\bar{\mathbf{y}}_1, \ldots, \bar{\mathbf{y}}_r\}$ if, for any set of binary assignments $\delta = [\delta_1, \ldots, \delta_r] \in \{0,1\}^r$, there exist parameters $\bar{\theta}$ such that, for any i, $\Phi(\mathbf{y}_i, \bar{\theta})$ is true if $\delta_i = 1$, and false if $\delta_i = 0$. Then, the VC dimension of Φ is defined as the size of the maximum set that Φ can shatter, i.e.,

$$\mathrm{VCdim}(\Phi) = \max_{S \text{ is shattered by } \Phi} |S|.$$

Interestingly, the VC dimension of Φ can be bounded by studying the topological properties of the inverse image, in the parameter domain, of the functions τ_i (Karpinski & Macintyre, 1997). More precisely, let $\bar{\mathbf{y}}_1, \ldots, \bar{\mathbf{y}}_z$ be vectors in $\mathbb{R}^{\bar{\gamma}}$, and $\mathbf{T} : \mathbb{R}^{\bar{p}} \to \mathbb{R}^{\bar{u}}$, $\bar{u} \le \bar{p}$, be defined as

$$\mathbf{T}(\bar{\theta}) = [\bar{\tau}_1(\bar{\theta}), \ldots, \bar{\tau}_{\bar{u}}(\bar{\theta})], \tag{A.1}$$

where $\bar{\tau}_1(\bar{\theta}), \ldots, \bar{\tau}_{\bar{u}}(\bar{\theta})$ are functions of the form of $\tau_i(\bar{\mathbf{y}}_j, \bar{\theta})$, i.e., for each r, $1 \le r \le \bar{u}$, there exist integers i and j such that $\bar{\tau}_r(\bar{\theta}) = \tau_i(\bar{\mathbf{y}}_j, \bar{\theta})$. Let $[\epsilon_1, \ldots, \epsilon_{\bar{u}}]$ be a regular value[1] of $\mathbf{T}$ and assume that there exists a positive integer B that bounds the number of connected components of $\mathbf{T}^{-1}(\epsilon_1, \ldots, \epsilon_{\bar{u}})$ and does not depend on the chosen $\epsilon_1, \ldots, \epsilon_{\bar{u}}$ and on the selected $\bar{\mathbf{y}}_j$. Then, the following proposition holds (Karpinski & Macintyre, 1997).

Theorem 5. *The VC dimension of Φ is bounded as follows:*

$$\mathrm{VCdim}(\Phi) \le 2 \log B + \bar{p}(16 + 2 \log \bar{s}).$$

Therefore, Theorem 5 provides a bound on the VC dimension of Φ that depends on the number $\bar{p}$ of parameters, the total number $\bar{s}$ of

[1] We recall that $[\epsilon_1, \ldots, \epsilon_{\bar{u}}]$ is a regular value of $\mathbf{T}$ if either $\mathbf{T}^{-1}([\epsilon_1, \ldots, \epsilon_{\bar{u}}]) = \emptyset$ or $\mathbf{T}^{-1}([\epsilon_1, \ldots, \epsilon_{\bar{u}}])$ is a $(\bar{p} - \bar{u})$–dimensional C^∞–submanifold of $\mathbb{R}^{\bar{p}}$.

functions τ_i, and the bound B on the number of connected components of $\mathbf{T}^{-1}$.

A bound on the number of connected components

A bound B on the number of connected components of $\mathbf{T}^{-1}$ can be obtained based on known results from the literature. In particular, the following theorem provides a bound in the case of a set of Pfaffian equations.

Theorem 6 (Gabrielov & Vorobjov, 2004). *Consider a system of equations $\bar{q}_1(\theta) = 0, \dots, \bar{q}_k(\theta) = 0$, where $\bar{q}_i$, $1 \le i \le k$, are Pfaffian functions in a domain $G \subseteq \mathbb{R}^{\bar{p}}$, having a common Pfaffian chain of length $\bar{\ell}$ and maximum degrees $(\bar{\alpha}, \bar{\beta})$. Then the number of connected components of the set $\{\theta | \bar{q}_1(\theta) = 0, \dots, \bar{q}_k(\theta) = 0\}$ is bounded by*

$$2^{\frac{\bar{\ell}(\bar{\ell}-1)}{2}+1}(\bar{\alpha}+2\bar{\beta}-1)^{\bar{p}-1}((2\bar{p}-1)(\bar{\alpha}+\bar{\beta})-2\bar{p}+2)^{\bar{\ell}}.$$

A.2. Proof of Theorem 1

First, we prove Theorem 1. The proofs of Theorems 3 and 4 will adopt the same argumentative scheme. As already mentioned, we will follow the reasoning in Karpinski and Macintyre (1997), Scarselli et al. (2018), which consists of three steps. First, it is shown that the computation of GNNs can be represented by a set of equations defined by Pfaffian functions. Then, using the format of such Pfaffian functions, Theorem 6 allows to derive a bound on the number of connected components of the space defined by the equations. Finally, Theorem 5 provides a bound on the VC dimension of GNNs.

Let us define a set of equations $\tau_i(\mathbf{y}, \theta) = 0$ that specifies the computation of the generic GNN model of Eq. (5). Here, θ collects the GNN parameters, while $\mathbf{y}$ contains all the variables necessary to define the GNN calculation, that is, some variables involved in the representation of the GNN input, i.e., the input graph, and other variables used for the representation of the internal features of the GNN.

More precisely, let us assume that $\mathsf{COMBINE}^{(1)}$ has $p_{\mathsf{comb}^{(1)}}$ parameters, $\mathsf{COMBINE}^{(t)}$ has p_{comb} parameters for $2 \le t \le L$, $\mathsf{AGGREGATE}^{(1)}$ has $p_{\mathsf{agg}^{(1)}}$ parameters, $\mathsf{AGGREGATE}^{(t)}$ has p_{agg} parameters for $2 \le t \le L$, and $\mathsf{READOUT}$ has p_{read} parameters.

Then, the dimension of θ is $p_{\mathsf{comb}^{(1)}} + p_{\mathsf{agg}^{(1)}} + (L-1)(p_{\mathsf{comb}} + p_{\mathsf{agg}}) + p_{\mathsf{read}}$. Moreover, for a given graph $\mathbf{G} = (G, \mathbf{L})$ in $\mathcal{G}$, $\mathbf{y}$ contains some vectorial representation of $\mathbf{G}$, namely the Nq graph attributes in $\mathbf{L}_G$, and a vectorial representation of the adjacency matrix $\mathbf{A}$, which requires $N(N-1)/2$ elements. Besides, to define the equations, we use the same trick as in Karpinski and Macintyre (1997) and introduce new variables for each computation unit of the network. These variables belong to the input $\mathbf{y}$ of τ. Formally, we consider a vector of d variables $\mathbf{h}_v^{(k)}$ for each node v and for each layer k. Note that, as we may be interested in defining multiple GNN computations on multiple graphs at the same time, here v implicitly addresses a specific node of some graph in the domain. Finally, a variable $\mathsf{READOUT}$ for each graph contains just a single output of the GNN. Thus, in total, the dimension of $\mathbf{y}$ is $Nq + N(N-1)/2 + NdL + 1$.

Therefore, the computation of the GNN model in (5) is defined by the following set of $LNd + Nq + 1$ equations,

$$\mathbf{h}_v^{(0)} - \mathbf{L}_v = 0, \tag{A.2}$$

$$\mathbf{h}_v^{(t+1)} - \mathsf{COMBINE}^{(t+1)}\left(\mathbf{h}_v^{(t)}, \mathsf{AGGREGATE}^{(t+1)}(\{\!\{\mathbf{h}_u^{(t)} | u \in \mathsf{ne}(v)\}\!\}, \mathbf{A})\right) = 0, \tag{A.3}$$

$$\overline{\mathsf{READOUT}} - \mathsf{READOUT}(\{\!\{\mathbf{h}_v^{(L)} : v \in V\}\!\}) = 0, \tag{A.4}$$

where $\mathbf{A}$ is the variable storing the adjacency matrix of the input graph. We can assume that $\mathbf{A}$ is valid for any finite graph.

The following lemma specifies the format of the Pfaffian functions involved in Eqs. (A.2)–(A.4).

Lemma 7. *Let $\mathsf{COMBINE}^{(t)}$, $\mathsf{AGGREGATE}^{(t)}$ and $\mathsf{READOUT}$ be Pfaffian functions with format, respectively, $(\alpha_{\mathsf{comb}}, \beta_{\mathsf{comb}}, \ell_{\mathsf{comb}})$, $(\alpha_{\mathsf{agg}}, \beta_{\mathsf{agg}}, \ell_{\mathsf{agg}})$, $(\alpha_{\mathsf{read}}, \beta_{\mathsf{read}}, \ell_{\mathsf{read}})$ w.r.t. the variables $\mathbf{y}$ and θ described above, then:*

1. *the left part of Eq. (A.2) is a polynomial of degree 1;*
2. *the left part of Eq. (A.3) is a Pfaffian function with format $(\alpha_{\mathsf{agg}} + \beta_{\mathsf{agg}} - 1 + \alpha_{\mathsf{comb}}\beta_{\mathsf{agg}}, \beta_{\mathsf{comb}}, \ell_{\mathsf{comb}} + \ell_{\mathsf{agg}})$;*
3. *the left part of Eq. (A.4) is a Pfaffian function with format $(\alpha_{\mathsf{read}}, \beta_{\mathsf{read}}, \ell_{\mathsf{read}})$;*
4. *Eqs. (A.2)–(A.4) constitute a system of Pfaffian equations with a maximal format $(\alpha_{\mathsf{system}}, \beta_{\mathsf{system}}, \ell_{\mathsf{system}})$, where $\alpha_{\mathsf{system}} = \max\{\alpha_{\mathsf{agg}} + \beta_{\mathsf{agg}} - 1 + \alpha_{\mathsf{comb}}\beta_{\mathsf{agg}}, \alpha_{\mathsf{read}}\}$, $\beta_{\mathsf{system}} = \max\{\beta_{\mathsf{comb}}, \beta_{\mathsf{read}}\}$ and $\ell_{\mathsf{system}} = LNd(\ell_{\mathsf{comb}} + \ell_{\mathsf{agg}}) + \ell_{\mathsf{read}}$.*

Proof. The first point is straightforwardly evident, while the third is true by definition. The second point can be derived by applying the composition lemma for Pfaffian functions (Gabrielov & Vorobjov, 1995), according to which, if two functions f and g have format $(\alpha_f, \beta_f, \ell_f)$ and $(\alpha_g, \beta_g, \ell_g)$, respectively, then their composition $f \circ g$ has format $(\alpha_g + \beta_g - 1 + \alpha_f\beta_g, \beta_f, \ell_f + \ell_g)$. Finally, the fourth point is obtained by taking the maximum of the format of the involved Pfaffian equations also observing that the common chain is the concatenation of the chains. □

Now we can proceed with the proof of Theorem 1.

Proof. Let $\mathbf{T}$ be defined as in Eq. (A.1), where $\tau_i(\mathbf{y}, \theta) = 0$ are the equations in (A.2), (A.3), (A.4). Combining Theorem 6 with the formats provided by point 3. of Lemma 7, for any input graph and any value of the variables $\mathbf{y}$, the number of connected components of $\mathbf{T}^{-1}$ satisfies

$$B \le 2^{\frac{\bar{\ell}(\bar{\ell}-1)}{2}+1}(\bar{\alpha}+2\bar{\beta}-1)^{\bar{p}-1}((2\bar{p}-1)(\bar{\alpha}+\bar{\beta})-2\bar{p}+2)^{\bar{\ell}}, \tag{A.5}$$

where $\bar{p} = p_{\mathsf{comb}^{(1)}} + p_{\mathsf{agg}^{(1)}} + (L-1)(p_{\mathsf{comb}} + p_{\mathsf{agg}}) + p_{\mathsf{read}}$, $\bar{\alpha} = \alpha_{\mathsf{system}}$, $\bar{\beta} = \beta_{\mathsf{system}}$, $\bar{\ell} = \bar{p}(LNd(\ell_{\mathsf{comb}} + \ell_{\mathsf{agg}}) + \ell_{\mathsf{read}})$.

By Theorem 5, the VC dimension of the GNN described by Eqs. (A.2)–(A.4) is bounded by

$$\mathsf{VCdim}(\mathsf{GNN}) \le 2\log B + \bar{p}(16 + 2\log\bar{s}), \tag{A.6}$$

where $\bar{s} = LNd + Nq + 1$. Thus, substituting Eq. (A.5) in Eq. (A.6), we have:

$$\begin{aligned}
\mathsf{VCdim}(\mathsf{GNN}) &\le 2\log B + \bar{p}(16 + 2\log\bar{s}) \\
&\le 2\log\left(2^{\frac{\bar{\ell}(\bar{\ell}-1)}{2}+1}(\bar{\alpha}+2\bar{\beta}-1)^{\bar{p}-1}((2\bar{p}-1)(\bar{\alpha}+\bar{\beta})-2\bar{p}+2)^{\bar{\ell}}\right) + \\
&\quad + \bar{p}(16 + 2\log\bar{s}) \\
&= \bar{\ell}(\bar{\ell}-1) + 2(\bar{p}-1)\log\left(\bar{\alpha}+2\bar{\beta}-1\right) + 2\bar{\ell}\log\left((2\bar{p}-1)(\bar{\alpha}+\bar{\beta})-2\bar{p}+2\right) + \\
&\quad + \bar{p}(16 + 2\log\bar{s}) + 2,
\end{aligned}$$

obtaining Eq. (6).

If we denote $H = LNd((\ell_{\mathsf{comb}} + \ell_{\mathsf{agg}}) + \ell_{\mathsf{read}})$, we have:

$$\begin{aligned}
\mathsf{VCdim}(\mathsf{GNN}) &\le \bar{p}H(\bar{p}H - 1) + 2(\bar{p}-1)\log(\alpha_{\mathsf{system}} + 2\beta_{\mathsf{system}} - 1) \\
&\quad + 2\bar{p}H\log\left((2\bar{p}-1)(\alpha_{\mathsf{system}} + \beta_{\mathsf{system}}) - 2\bar{p} + 2\right) \\
&\quad + \bar{p}(16 + 2\log(\bar{s})) + 2 \\
&\le \bar{p}^2H^2 + 2\bar{p}\log(3\gamma) \\
&\quad + 2\bar{p}H\log((4\gamma - 2)\bar{p} + 2 - 2\gamma) \\
&\quad + \bar{p}(16 + 2\log(\bar{s})) + 2.
\end{aligned} \tag{A.7}$$

Then, by replacing $\bar{p}$, H and $\bar{s}$, and setting $\gamma = \max\{\bar{\alpha}, \bar{\beta}\}$, it follows that:

$$\begin{aligned}
&\mathsf{VCdim}(\mathsf{GNN}) \\
&\le (p_{\mathsf{comb}^{(1)}} + p_{\mathsf{agg}^{(1)}} + (L-1)(p_{\mathsf{comb}} + p_{\mathsf{agg}}) + p_{\mathsf{read}})^2(LNd(\ell_{\mathsf{comb}} + \ell_{\mathsf{agg}}) + \ell_{\mathsf{read}})^2 \\
&+ 2(p_{\mathsf{comb}^{(1)}} + p_{\mathsf{agg}^{(1)}} + (L-1)(p_{\mathsf{comb}} + p_{\mathsf{agg}}) + p_{\mathsf{read}})\log(3\gamma) \\
&+ 2(p_{\mathsf{comb}^{(1)}} + p_{\mathsf{agg}^{(1)}} + (L-1)(p_{\mathsf{comb}} + p_{\mathsf{agg}}) + p_{\mathsf{read}}).
\end{aligned}$$

$$\cdot \log\left((4\gamma-2)(p_{\mathrm{comb}^{(1)}}+p_{\mathrm{agg}^{(1)}}+(L-1)(p_{\mathrm{comb}}+p_{\mathrm{agg}})+p_{\mathrm{read}})+2-2\gamma\right)$$
$$+\ (p_{\mathrm{comb}^{(1)}}+p_{\mathrm{agg}^{(1)}}+(L-1)(p_{\mathrm{comb}}+p_{\mathrm{agg}})+p_{\mathrm{read}})(16+2\log(LNd+Nq+1)),$$

which leads to Eq. (7) as in the thesis. □

A.3. Proof of Theorem 2

The orders of growth of the VC dimension w.r.t. $\bar{p}, N, L, d, q$ are straightforwardly derived by inspecting Eq. (7) in Theorem 1.

A.4. Proof of Theorem 3

As already mentioned in Appendix A.2, the proof of Theorem 3 follows the same scheme used in proof of Theorem 1. We just recall that we consider a GNN defined by the following updating equation:

$$\mathbf{h}_v^{(t+1)} = \sigma(\mathbf{W}_{\mathrm{comb}}^{(t+1)}\mathbf{h}_v^{(t)} + \mathbf{W}_{\mathrm{agg}}^{(t+1)}\mathbf{h}_{\mathrm{ne}(v)}^{(t+1)} + \mathbf{b}^{(t+1)}), \tag{A.8}$$

where σ is the activation function and

$$\mathbf{h}_{\mathrm{ne}(v)}^{(t+1)} = \sum_{u\in \mathrm{ne}(v)} \mathbf{h}_u^{(t)}. \tag{A.9}$$

The hidden states are initialized as $\mathbf{h}_v^{(0)} = \mathbf{L}_v$. It is easily seen that the total number of parameters is $p = (2d+1)(d(L-1)+q+1)-q$. Indeed, the parameters are:

- $\mathbf{W}_{\mathrm{comb}}^{(1)}, \mathbf{W}_{\mathrm{agg}}^{(1)} \in \mathbb{R}^{d\times q}, \mathbf{b}^{(1)} \in \mathbb{R}^{d\times 1}$, so that we have $2dq + d$ parameters;
- $\mathbf{W}_{\mathrm{comb}}^{(t)}, \mathbf{W}_{\mathrm{agg}}^{(t)} \in \mathbb{R}^{d\times d}, \mathbf{b}^{(t)} \in \mathbb{R}^{d\times 1}$ for $t = 2, \ldots, L$, so that we have $(2d^2+d)(L-1)$ parameters;
- $\mathbf{w} \in \mathbb{R}^{1\times d}, b \in \mathbb{R}$, so that we have $d+1$ parameters.

Summing up, we will have $2dq + d + (2d^2+d)(L-1) + d + 1 = (2d+1)(d(L-1)+q+1) - q$ parameters. By Eq. (A.8), the computation of the GNN is straightforwardly defined by the following set of $LNd + Nq + 1$ equations:

$$\mathbf{h}_v^{(0)} - \mathbf{L}_v = 0, \tag{A.10}$$
$$\mathbf{h}_v^{(t+1)} - \sigma\Big(\mathbf{W}_{\mathrm{comb}}^{(t+1)}\mathbf{h}_v^{(t)} + \sum_u \mathbf{W}_{\mathrm{agg}}^{(t+1)}\mathbf{h}_u^{(t)} m_{v,u} + \mathbf{b}^{(t+1)}\Big) = 0, \tag{A.11}$$
$$\mathsf{READOUT} - \sigma\Big(\sum_{v\in V} \mathbf{w}\mathbf{h}_v^{(L)} + b\Big) = 0, \tag{A.12}$$

where $m_{v,u}$ is a binary value, which is 1 when v and u are connected and 0, otherwise.

Now we state the analogue of Lemma 7 to retrieve the format of Pfaffian functions involved in the above equations.

Lemma 8. *Let σ be a Pfaffian function in $\mathbf{x}$ with format $(\alpha_\sigma, \beta_\sigma, \ell_\sigma)$, then w.r.t. the variables $\mathbf{y}$ and $\mathbf{w}$ described above,*

1. *the left part of Eq. (A.10) is a polynomial of degree 1;*
2. *the left part of Eq. (A.11) is a Pfaffian function having format $(2+3\alpha_\sigma, \beta_\sigma, \ell_\sigma)$;*
3. *the left part of Eq. (A.12) is a Pfaffian function having format $(1+2\alpha_\sigma, \beta_\sigma, \ell_\sigma)$;*
4. *Eqs. (A.10)–(A.12) constitute a system of Pfaffian equations with format $(2+3\alpha_\sigma, \beta_\sigma, H\ell_\sigma)$, where the shared chain is obtained by concatenating the chains of $H = LNd+1$ equations in (A.11),(A.12), including an activation function.*

Proof. The first point is straightforwardly evident. The second and third points can be derived by applying the composition lemma for Pfaffian functions. Actually, the formula inside σ in Eq. (A.11) is a polynomial of degree 3, due to the factors $\mathbf{h}_u^{(t-1)}\mathbf{W}_{\mathrm{agg}}^{(t)} m_{v,u}$, while the formula inside σ in Eq. (A.12) is a polynomial of degree 2, due to the factors $\mathbf{h}_v^{(L)}\mathbf{W}$. Moreover, polynomials are Pfaffian functions with null chains, α equals 0 and β equals their degrees. Thus, the functions inside σ in Eqs. (A.11) and (A.12) have format $(0,3,0)$ and $(0,2,0)$, respectively. Then, the thesis follows by the composition lemma (Gabrielov & Vorobjov, 1995), according to which if two functions f and g have format $(\alpha_f, \beta_f, \ell_f)$ and $(\alpha_g, \beta_g, \ell_g)$, respectively, then their composition $f\circ g$ has format $(\alpha_g + \beta_g - 1 + \alpha_f\beta_g, \beta_f, \ell_f + \ell_g)$. Finally, the fourth point is a consequence of the fact that the equations are independent and the chains can be concatenated. The length of the chain derives directly from the existence of $H = LNd+1$ equations using σ. The degree is obtained from the largest degree of a Pfaffian function, which is the one in Eq. (A.11). □

As in Appendix A.2, it is enough to combine Lemma 8, Theorem 6 and Theorem 5 to obtain the bounds stated in the thesis. The bounds on the VC dimension of the specific GNN with logsig as activation function is easily found since the format of logsig is (2,1,1):

$$\begin{aligned}\mathsf{VCdim}(\mathsf{GNN}) \le\ & ((2d+1)(d(L-1)+q+1)-q)^2(LNd+1)^2 \\ &+ 2((2d+1)(d(L-1)+q+1)-q)\log(9) \\ &+ 2((2d+1)(d(L-1)+q+1)-q)\log(16((2d+1)(d(L-1)+q+1)-q)) \\ &+ ((2d+1)(d(L-1)+q+1)-q)(16+2\log(LNd+Nq+1)).\end{aligned}$$

□

A.5. Proof of Theorem 4

Let us call Basic GNN (BGNN) the model of Eqs. (2)–(3). The proof is based on the introduction of an extended version, which we call EGNN, that can simulate the BGNN. Due to this capability, the EGNN can shatter any set that is shattered by the BGNN so that its VC dimension is greater or equal to the VC dimension of the BGNN. The proof will follow by bounding the VC dimension of the former model.
More precisely, the EGNN exploits the same aggregation mechanism of the BGNN to compute the features, which is described by Eq. (2). On the other hand, the READOUT function is defined as

$$\mathsf{READOUT}\Big(\{\!\{\mathbf{h}_v^{(L)} \mid v\in V\}\!\}\Big) := f\Big(\sum_{v\in V} \mathbf{w}\mathbf{h}_v^{(L)} c_v + b\Big), \tag{A.13}$$

where c_v are additional real inputs used to weight each node feature in the READOUT function. The simulation is based on the following steps.

(1) Each input graph G of the BGNN is transformed to another graph G', where all the nodes having the same 1–WL color are merged into a single node and the edges are merged consequently;
(2) The EGNN is applied to G' and each c_v is set equal to the number of nodes that have been merged to obtain node v.

Note that a GNN cannot distinguish nodes with the same color as the computation is the same on all these nodes. Thus, the BGNN and the EGNN produce the same features on nodes sharing color. As a consequence, also the READOUTs of the two models have the same output, when the c_v are equal to the number of nodes within each color cluster. Given these assumptions, the number of equations describing the Pfaffian variety associated to the EGNN is reduced to $s_c = C_1 d + C_0 q + 1$, which can be used in place of $\bar{s}$ in Theorem 5. Moreover, also the chains of the Pfaffian functions in merged equations can be merged and we have that H can be replaced by $H_c = C_1 d + 1$. Finally, the length of the chain ℓ of Theorem 6 is replaced by $\bar{\ell}_c = \bar{p} H_c \ell_\sigma$.

With the above changes, we can replace the variables in Eq. (A.7) as in Appendix A.2, obtaining:

$$\begin{aligned}\mathsf{VCdim}(\mathsf{GNN}) &\le \bar{p}H_c(\bar{p}H_c - 1) + 2\bar{p}\log(9) \\ &\quad + 2\bar{p}H_c\log(16\bar{p}-7) \\ &\quad + \bar{p}(16 + 2\log(\bar{s}_c)) + 2 \\ &\le \bar{p}^2(C_1 d+1)^2 + 2\bar{p}\log(9) \\ &\quad + 2\bar{p}(C_1 d+1)\log(16\bar{p}-7) \\ &\quad + \bar{p}(16 + 2\log(C_1 d + C_0 q + 1)).\end{aligned}$$

and the thesis holds. □

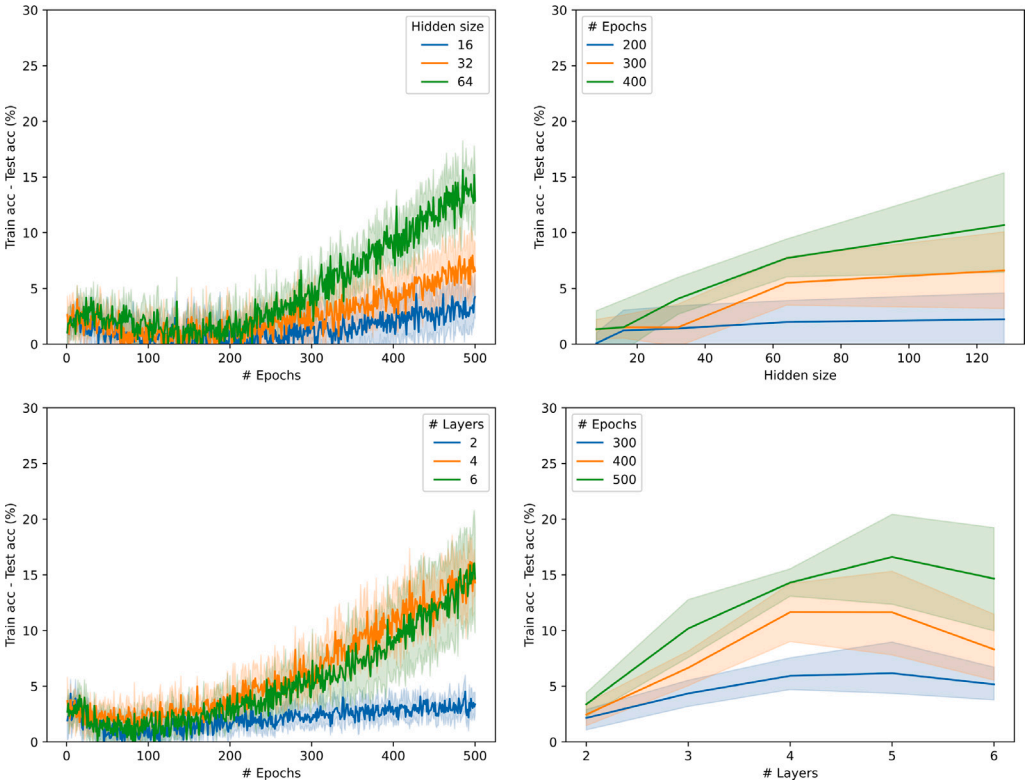


Fig. B.4. Results on the task **E1** for GNNs with activation function atan over the dataset **PROTEINS**.

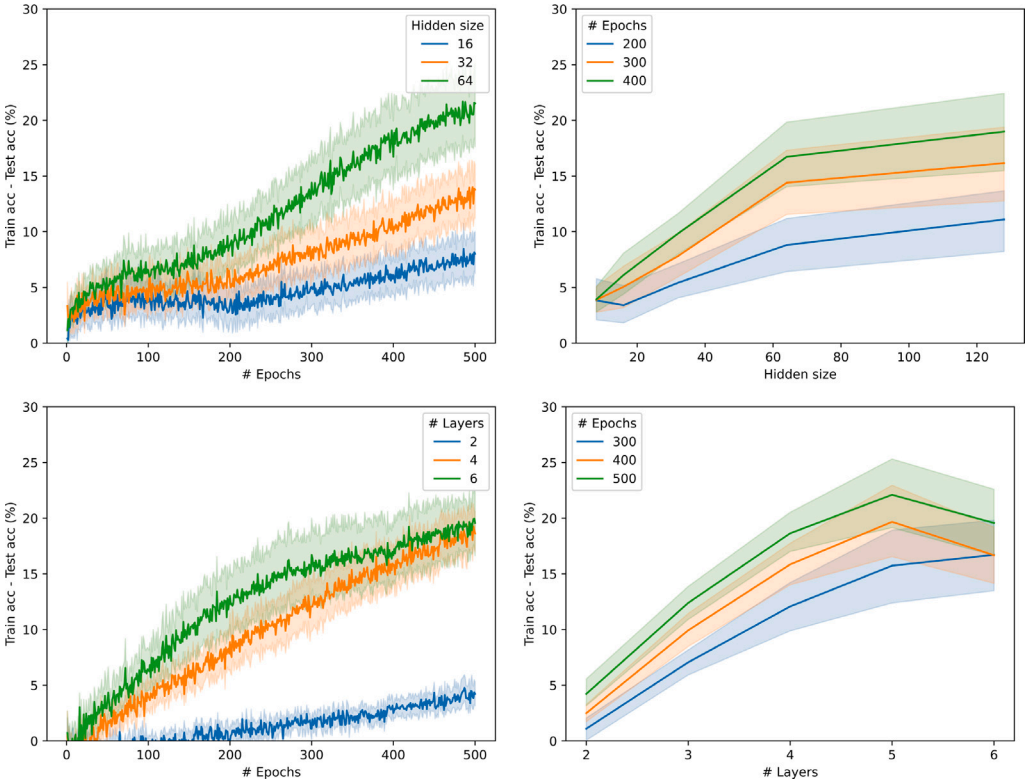


Fig. B.5. Results on the task **E1** for GNNs with activation function atan over the dataset **PTC-MR**.

Appendix B. Experiments on other datasets

In this appendix, we report the additional results on the experiment E1, regarding the evolution of the difference between the training and the test set, for GNNs with activation function $f \in \{\arctan, \tanh\}$, over a dataset $D \in \{$**PROTEINS**, **PTC-MR**$\}$. Each figure shows the evolution of diff through the epochs, for certain values of hd, keeping fixed $l = 3$, and for certain values of l, keeping fixed hd $= 32$; for each figure, the

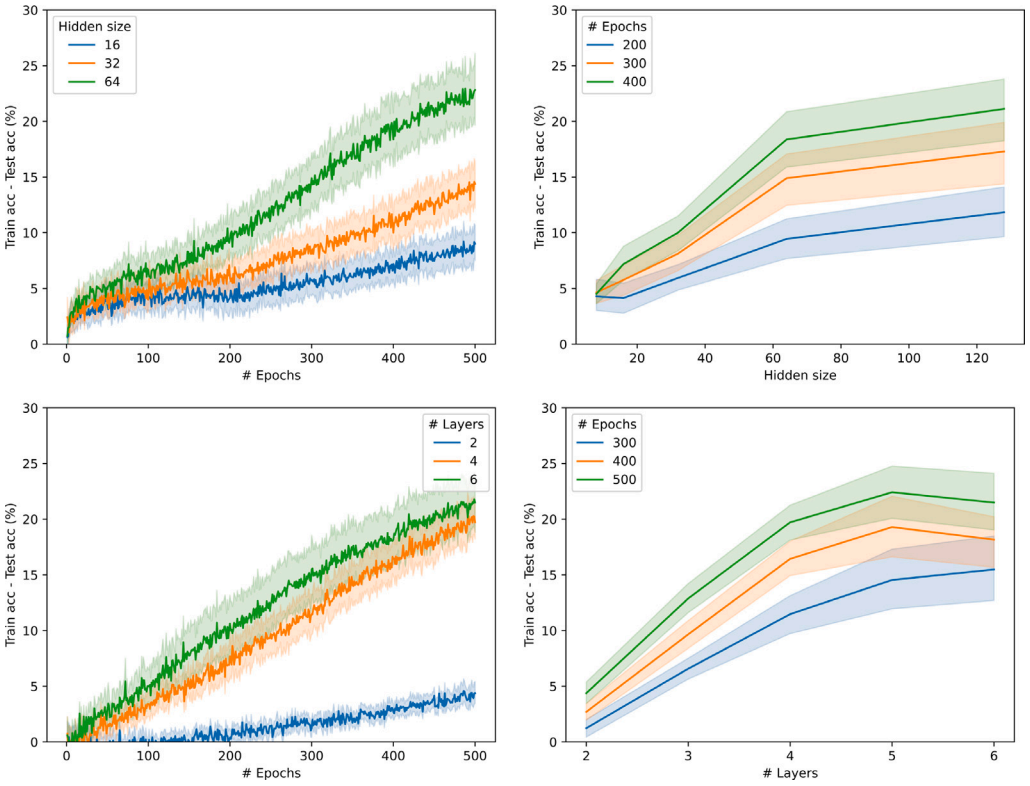


Fig. B.6. Results on the task **E1** for GNNs with activation function tanh over the dataset **PROTEINS**.

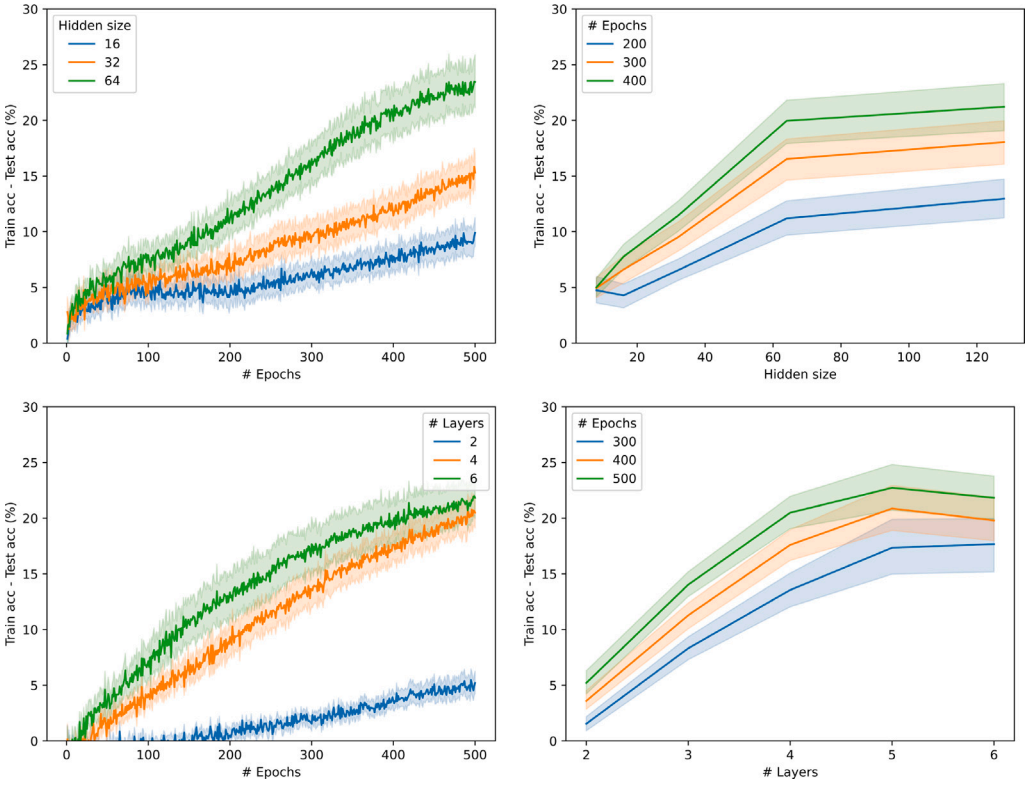


Fig. B.7. Results on the task **E1** for GNNs with activation function tanh over the dataset **PTC-MR**.

picture on the left shows how diff evolves as the hidden size increases, while the picture on the right shows how diff evolves as the hidden size, or the number of layers, increases (see Figs. B.4–B.7). The experiments reported here confirm the same conclusion drawn in Section 5.

Data availability

I have shared my code within the manuscript.

References

Bartlett, P. L., & Maass, W. (2003). Vapnik-Chervonenkis dimension of neural nets. In *The handbook of brain theory and neural networks*, (pp. 1188–1192). Citeseer.

Baum, E., & Haussler, D. (1988). What size net gives valid generalization? In *Advances in neural information processing systems*: *vol. 1*.

Bianchini, M., & Scarselli, F. (2014). On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(8), 1553–1565.

Bodnar, C., Frasca, F., Otter, N., Wang, Y., Lio, P., Montufar, G. F., et al. (2021). Weisfeiler and lehman go cellular: Cw networks. *Advances in Neural Information Processing Systems*, *34*, 2625–2640.

Bodnar, C., Frasca, F., Wang, Y., Otter, N., Montufar, G. F., Lio, P., et al. (2021). Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks. In *International conference on machine learning* (pp. 1026–1037). PMLR.

Borgwardt, K. M., Ong, C. S., Schönauer, S., Vishwanathan, S., Smola, A. J., & Kriegel, H.-P. (2005). Protein function prediction via graph kernels. *Bioinformatics*, *21*(suppl_1), i47–i56.

Brugiapaglia, S., Liu, M., & Tupper, P. (2020). Generalizing outside the training set: When can neural networks learn identity effects? arXiv preprint arXiv:2005.04330.

Brugiapaglia, S., Liu, M., & Tupper, P. (2022). Invariance, encodings, and generalization: learning identity effects with neural networks. *Neural Computation*, *34*(8), 1756–1789.

Daubechies, I., DeVore, R., Foucart, S., Hanin, B., & Petrova, G. (2022). Nonlinear approximation and (deep) ReLU networks. *Constructive Approximation*, *55*(1), 127–172.

D'Inverno, G. A., Bianchini, M., Sampoli, M. L., & Scarselli, F. (2024). On the approximation capability of GNNs in node classification/regression tasks. *Soft Computing*, *28*(13), 8527–8547.

D'Inverno, G. A., Brugiapaglia, S., & Ravanelli, M. (2024). Generalization limits of graph neural networks in identity effects learning. *Neural Networks*, Article 106793.

Esser, P., Chennuru Vankadara, L., & Ghoshdastidar, D. (2021). Learning theory can (sometimes) explain generalisation in graph neural networks. *Advances in Neural Information Processing Systems*, *34*, 27043–27056.

Fresca, S., Manzoni, A., Dedè, L., & Quarteroni, A. (2020). Deep learning-based reduced order models in cardiac electrophysiology. *PLoS One*, *15*(10), Article e0239416.

Gabrielov, A., & Vorobjov, N. (1995). Complexity of stratification of semi-pfaffian sets. *Discrete & Computational Geometry*, *14*(1), 71–91.

Gabrielov, A., & Vorobjov, N. (2004). Complexity of computations with Pfaffian and Noetherian functions. *Normal Forms, Bifurcations and Finiteness Problems in Differential Equations*, *137*, 211–250.

Garg, V., Jegelka, S., & Jaakkola, T. (2020). Generalization and representational limits of graph neural networks. In *International conference on machine learning* (pp. 3419–3430). PMLR.

Gnecco, G., Sanguineti, M., et al. (2008). Approximation error bounds via Rademacher complexity. *Applied Mathematical Sciences*, *2*, 153–176.

Goldberg, P., & Jerrum, M. (1993). Bounding the vapnik-chervonenkis dimension of concept classes parameterized by real numbers. In *Proceedings of the sixth annual conference on computational learning theory* (pp. 361–369).

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, *30*.

Hammer, B. (2000). On the approximation capability of recurrent neural networks. *Neurocomputing*, *31*(1–4), 107–123.

Hammer, B. (2001). On the generalization ability of recurrent networks. In *Artificial neural networks—ICANN 2001: international conference Vienna, Austria, August 21–25, 2001 proceedings 11* (pp. 731–736). Springer.

Haussler, D., & Warmuth, M. (2018). The probably approximately correct (PAC) and other learning models. *The Mathematics of Generalization*, 17–36.

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

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*(2), 251–257.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Jacot, A., Gabriel, F., & Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*: *vol. 31*.

Ju, H., Li, D., Sharma, A., & Zhang, H. R. (2023). Generalization in graph neural networks: Improved PAC-Bayesian bounds on graph diffusion. In *International conference on artificial intelligence and statistics* (pp. 6314–6341). PMLR.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., et al. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*(7873), 583–589.

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

Khovanski, A. G. (1991). *Fewnomials*: *vol. 88*, American Mathematical Soc..

Kiefer, S. (2020). The Weisfeiler-Leman algorithm: an exploration of its power. *ACM SIGLOG News*, *7*(3), 5–27.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Knyazev, B., Taylor, G. W., & Amer, M. (2019). Understanding attention and generalization in graph neural networks. In *Advances in neural information processing systems*: *vol. 32*.

Koiran, P., & Sontag, E. D. (1997). Neural networks with quadratic VC dimension. *Journal of Computer and System Sciences*, *54*(1), 190–198.

Koiran, P., & Sontag, E. D. (1998). Vapnik-Chervonenkis dimension of recurrent neural networks. *Discrete Applied Mathematics*, *86*(1), 63–79.

Koltchinskii, V. (2001). Rademacher penalties and structural risk minimization. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, *47*(5), 1902–1914.

Krebs, A., & Verbitsky, O. (2015). Universal covers, color refinement, and two-variable counting logic: Lower bounds for the depth. In *2015 30th annual ACM/IEEE symposium on logic in computer science* (pp. 689–700). IEEE.

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., et al. (2023). Learning skillful medium-range global weather forecasting. *Science*, eadi2336.

Liao, R., Urtasun, R., & Zemel, R. (2020). A PAC-bayesian approach to generalization bounds for graph neural networks. arXiv preprint arXiv:2012.07690.

Liu, Z., & Zhou, J. (2022). *Introduction to graph neural networks*. Springer Nature.

Maass, W. (1994). Neural nets with superlinear VC-dimension. *Neural Computation*, *6*(5), 877–884.

Morris, C., Geerts, F., Tönshoff, J., & Grohe, M. (2023). WL meet VC. In *Proceedings of the 40th International Conference on Machine Learning*.

Morris, C., Kriege, N. M., Bause, F., Kersting, K., Mutzel, P., & Neumann, M. (2020). TUDataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 workshop on graph representation learning and beyond (gRL+ 2020)*. arXiv: 2007.08663. URL www.graphlearning.io.

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., et al. (2019). Weisfeiler and leman go neural: Higher-order graph neural networks. In *AAAI conference on artificial intelligence* (pp. 4602–4609).

Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., & Srebro, N. (2018). Towards understanding the role of over-parametrization in generalization of neural networks. arXiv preprint arXiv:1805.12076.

Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., et al. (2022). Tackling climate change with machine learning. *ACM Computing Surveys*, *55*(2), 1–96.

Sakurai, A. (1995). On the VC-dimension of depth four threshold circuits and the complexity of Boolean-valued functions. *Theoretical Computer Science*, *137*(1), 109–127.

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

Scarselli, F., Tsoi, A. C., & Hagenbuchner, M. (2018). The Vapnik-Chervonenkis dimension of graph and recursive neural networks. *Neural Networks*, *108*, 248–259.

Sontag, E. D., et al. (1998). VC dimension of neural networks. *NATO ASI Series F Computer and Systems Sciences*, *168*, 69–96.

Vapnik, V. (2006). *Estimation of dependences based on empirical data*. Springer Science & Business Media.

Vapnik, V. N., & Chervonenkis, A. Y. (1968). On the uniform convergence of relative frequencies of events to their probabilities. In *Doklady akademii nauk USsR*: *vol. 181*, (no. 4), (pp. 781–787).

Vapnik, V., & Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, *16*(2), 264–280.

Vapnik, V., Levin, E., & Le Cun, Y. (1994). Measuring the VC-dimension of a learning machine. *Neural Computation*, *6*(5), 851–876.

Verma, S., & Zhang, Z.-L. (2019). Stability and generalization of graph convolutional neural networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1539–1548).

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

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

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

Yun, S., Jeong, M., Kim, R., Kang, J., & Kim, H. J. (2019). Graph transformer networks. In *Advances in neural information processing systems*: *vol. 32*.

Zhang, M., Qamar, M., Kang, T., Jung, Y., Zhang, C., Bae, S.-H., et al. (2023). A survey on graph diffusion models: Generative ai in science for molecule, protein and material. arXiv preprint arXiv:2304.01565.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., et al. (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*, 57–81.