A query language perspective on graph learning

A Query Language Perspective on Graph Learning[*]

Floris Geerts

Abstract

A key component of graph and relational learning methods is the computation of vector representations of the input graphs or relations. The starting point of this tutorial is that we model this computation as queries, mapping relational objects into the realm of real vector spaces. We then revisit recent works in the machine learning community on the expressive power of graph learning methods from this unifying query language perspective. Here, we consider the expressive power related to the discrimination of inputs and to the approximation power of functions. Finally, we argue that the bridge between graph learning and query languages opens many interesting avenues for further research.

1 Introduction

In recent years, the development of machine learning methods for graph (and relational) data has gained considerable attention. See e.g., the recent graph learning books (Hamilton, 2020; Ma and Tang, 2021; Wu et al., 2022). Underlying most graph learning methods is the computation of graph, vertex or – more generally – p-vertex embeddings in some real vector space. The rationale behind this is that by mapping discrete graph objects to vectors in some continuous space, the whole arsenal of continuous optimization and learning techniques becomes available.

Crucial is that the embedding methods are graph isomorphism invariant, hereby ensuring that the methods only rely on intrinsic graph properties and not on the chosen graph representation. It is precisely this invariance requirement that prevents the use of classical machine learning methods and has led, for example, to the development of graph neural networks.

We define embeddings more formally, as follows. Let $\mathcal{G}$ be the set of all graphs and $\mathcal{V}$ be the set of all vertices. Furthermore, let the output space $\mathcal{Y}$ be a real vector space. Then, *graph embeddings* are functions of the form $\xi : \mathcal{G} \to \mathcal{Y}$, *vertex embeddings* are of the form $\xi : \mathcal{G} \to (\mathcal{V} \to \mathcal{Y})$ and *p-vertex embeddings* are of the form $\xi : \mathcal{G} \to (\mathcal{V}^p \to \mathcal{Y})$. Graph and vertex embeddings are thus 0-vertex and 1-vertex embeddings, respectively. In what follows, for a graph G in $\mathcal{G}$ and a p-vertex tuple $\mathbf{v}$ in G, we write $\xi(G, \mathbf{v})$ rather than $\xi(G)(\mathbf{v})$. *Invariance* of embeddings requires that for any graphs G and H in $\mathcal{G}$ and any isomorphism π from G to H, and all p-vertex tuples $\mathbf{v}$ in G, $\xi(G, \mathbf{v}) = \xi\big(\pi(G), \pi(\mathbf{v})\big)$ holds. Invariance of p-vertex embeddings, for $p \geq 1$, is often called *equivariance* instead.

To get a more concrete idea of how graph learning works, we next describe the simple, yet popular, *empirical risk minimization* (ERM) approach (see e.g., Shalev-Shwartz and Ben-David (2014); Mohri et al. (2018) for more background). Following the exposition in Jegelka

[*] Published version: https://doi.org/10.1145/3584372.3589936

(2022), graph learning can be seen to correspond to learning a *partially known target embedding* $\Xi : \mathcal{G} \to (\mathcal{V}^p \to \mathcal{Y})$. Here, the partial knowledge is typically given in the form of some *training data* $\mathcal{T} := \big\{(G_1, \mathbf{v}_1, y_1), \ldots, (G_m, \mathbf{v}_m, y_m)\big\}$ consisting of graphs G_i in $\mathcal{G}$, p-vertex tuples $\mathbf{v}_i$ in G_i and their *correct values* $y_i = \Xi(G_i, \mathbf{v}_i)$ in $\mathcal{Y}$, for $i = 1, \ldots, m$. The training data is assumed to be a representative sample. Finally, in the ERM approach, learning the embedding Ξ given training data $\mathcal{T}$ is achieved by *approximating* Ξ using p-vertex embeddings from some "hypothesis" class $\mathcal{L}$. In particular "learning" Ξ is achieved by returning

$$\hat{\xi} \in \operatorname*{argmin}_{\xi \in \mathcal{L}} \frac{1}{m} \sum_{i=1}^{m} \ell\big(G_i, \mathbf{v}_i, y_i, \xi(G_i, \mathbf{v}_i)\big) \equiv \operatorname*{argmin}_{\xi \in \mathcal{L}} \hat{\mathcal{R}}_{\mathcal{T}}(\xi),$$

where $\ell : \mathcal{G} \times \mathcal{V}^p \times \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a *loss* or *risk* function and the $(G_i, \mathbf{v}_i, y_i)$'s are the examples in the training data $\mathcal{T}$. The function $\hat{\mathcal{R}}_{\mathcal{T}}(\xi)$ is called the *empirical risk* of ξ (relative to $\mathcal{T}$). Machine learning systems solve the ERM problem for various loss functions and classes of embeddings.

Many different classes of embedding methods have been proposed, using handcrafted graph features, graph kernel methods, spectral graph information, and graph neural networks architectures. In the graph learning community, the emphasis is primarily on *experimentally* comparing and validating embedding methods.

Here, we want to *theoretically* analyze classes of embeddings and show how database theoreticians can contribute to this area. We first define the notion of *discriminative power* as a tool for doing so. We then provide a *unifying embedding language* – inspired by relational aggregate query languages – which allows to (i) classify embedding methods in fragments of the embedding language; and to (ii) derive insights on the discriminative power of the methods based on the power of the corresponding embedding language fragments. The tutorial is largely based on the ICLR'22 paper (Geerts and Reutter, 2022).

2 Discriminative Power

A natural way of measuring what classes $\mathcal{L}$ of embeddings can do is based on their ability to assign different vectors to different inputs. For example, if $\mathcal{L}$ solely consists of constant embeddings, then all inputs will be embedded in the same way by embeddings in $\mathcal{L}$. The class $\mathcal{L}$ has thus no discriminative power. Hence, unless the target embedding Ξ to be learned is constant as well, the ERM approach will provide a bad approximation of Ξ based on embeddings in $\mathcal{L}$.

At the other end of the spectrum, if $\mathcal{L}$ can assign different vectors to any pair of non-isomorphic inputs, then $\mathcal{L}$ has maximal discriminative power due to the invariance assumption on embeddings. This necessarily implies that $\mathcal{L}$ is powerful enough to solve the graph isomorphism problem. When the target embedding Ξ to be learned, however, embeds certain non-isomorphic inputs in the same way, the use of such complex $\mathcal{L}$ for learning Ξ may be too excessive. And indeed, most practical classes of embeddings will have a discriminative power in between these two extremes.

Let $\mathcal{L}$ be a class of p-vertex embeddings. The *discriminative power of* $\mathcal{L}$ is formally defined using the equivalence relation

$$\varrho_p(\mathcal{L}) := \big\{(G, \mathbf{v}, H, \mathbf{w}) \in (\mathcal{G} \times \mathcal{V}^p) \times (\mathcal{G} \times \mathcal{V}^p) \;\big|\; \forall \xi \in \mathcal{L} : \xi(G, \mathbf{v}) = \xi(H, \mathbf{w})\big\}.$$

In other words, $\varrho_p(\mathcal{L})$ reflects which inputs cannot be discriminated by embeddings in $\mathcal{L}$. For the constant embedding class $\mathcal{L}$ mentioned above, $\varrho_p(\mathcal{L})$ consists of all possible input pairs. For $\mathcal{L}$ of maximal discriminative power, $\varrho_p(\mathcal{L})$ only contains pairs of isomorphic inputs. In a nutshell, the larger $\varrho_p(\mathcal{L})$ the weaker the class $\mathcal{L}$ is in discriminating different inputs.

It is now easy to compare entirely different classes $\mathcal{L}$ and $\mathcal{L}'$ of embeddings by simply comparing how $\varrho_p(\mathcal{L})$ and $\varrho_p(\mathcal{L}')$ relate. Indeed, if $\varrho_p(\mathcal{L}) \subseteq \varrho_p(\mathcal{L}')$ then $\mathcal{L}$ is at least as powerful as $\mathcal{L}'$, or equivalently, the power of $\mathcal{L}'$ is bounded by that of $\mathcal{L}$. Similarly, $\varrho_p(\mathcal{L}) \subsetneq \varrho_p(\mathcal{L}')$ implies that $\mathcal{L}$ has more power than $\mathcal{L}'$, and $\varrho_p(\mathcal{L}) = \varrho_p(\mathcal{L}')$ implies that both classes have the same power.

Not only does the notion of discriminative power provide the means of comparing different classes of embeddings, it also arises in the context of approximation and generalization properties.

2.1 Approximation Properties

One the pillars of deep learning is the Universality Theorem of neural networks (Cybenko, 1989; Hornik, 1991; Leshno et al., 1993). Roughly stated, it says that the class of functions represented by neural networks is dense in the class of all continuous functions. Density implies that any continuous function can be approximated to arbitrary precision by a neural network (on compact domains). This property is sometimes used as a theoretical justification for using neural networks.

A similar question can be posed in the context of classes $\mathcal{L}$ of embeddings. That is, which target embeddings Ξ can be approximated to arbitrary precision by embeddings in $\mathcal{L}$? Clearly, understanding the approximation capabilities and limitations of embeddings in $\mathcal{L}$ gives insight in what target embeddings Ξ could in principle be learned by ERM.

Various works have addressed this question in the graph context (Abboud et al., 2021; Azizian and Lelarge, 2021; Chen et al., 2019; Dasoulas et al., 2020; Geerts and Reutter, 2022; Keriven and Peyré, 2019; Maehara and NT, 2019; Maron et al., 2019b). We here focus on the connection between the discriminative power and approximation properties of classes of embeddings. For more details we refer to Azizian and Lelarge (2021); Geerts and Reutter (2022).

For simplicity, we focus on graph embeddings of the form $\mathcal{G} \to \mathbb{R}$ but everything can be generalized to p-vertex embeddings and arbitrary target spaces $\mathcal{Y} = \mathbb{R}^d$ for some $d \in \mathbb{N}$ (Azizian and Lelarge, 2021; Geerts and Reutter, 2022). To define a sound notion of approximation we assume in this subsection that $\mathcal{G}$ is turned into a topological space $\hat{\mathcal{G}}$ and that the target embedding $\Xi : \hat{\mathcal{G}} \to \mathbb{R}$ to be learned is *continuous*. There are various ways of endowing the graph space $\mathcal{G}$ with a topology. For example, in the absence of real-valued features one can use the discrete topology. Alternatively, one can use the adjacency matrix (or tensor) representation of graphs and use standard topologies on such matrix spaces. Or, one can use any distance function between graphs to define a topology. We fix any such topology.

Let us denote by $\mathcal{C}(\hat{\mathcal{G}}, \mathbb{R})$ the class of all continuous graph embeddings $\xi : \hat{\mathcal{G}} \to \mathbb{R}$. The space $\mathcal{C}(\hat{\mathcal{G}}, \mathbb{R})$ is a real vector space since it is closed under linear combinations. If $\hat{\mathcal{G}}$ is moreover assumed to be compact, $\mathcal{C}(\hat{\mathcal{G}}, \mathbb{R})$ can be turned into a *normed space*. The typical norm used is the so-called *sup norm* defined by $\xi \to \|\xi\| := \sup\{|\xi(G)| \in \mathbb{R} \mid G \in \hat{\mathcal{G}}\}$. This norm basically measures the maximum value attained by $\xi(G)$ when G ranges over the compact space $\hat{\mathcal{G}}$. When $\hat{\mathcal{G}}$ is compact, this norm is indeed well-defined.

We can finally define which graph embeddings can be approximated by a class $\mathcal{L}$ of graph embeddings. More precisely, we define the *(uniform) closure of* $\mathcal{L}$, denoted by $\overline{\mathcal{L}}$, as follows

$$\overline{\mathcal{L}} \coloneqq \big\{\Xi \in \mathcal{C}(\hat{\mathcal{G}}, \mathbb{R}) \bigm| \forall \epsilon > 0, \exists \xi_\epsilon \in \mathcal{L} : \|\Xi - \xi_\epsilon\| < \epsilon\big\}.$$

Intuitively, $\overline{\mathcal{L}}$ contains all graph embeddings that can be approximated by embeddings in $\mathcal{L}$. Clearly, linking back to the problem of learning an unknown target embedding $\Xi : \mathcal{G} \to \mathbb{R}$, it is desirable that $\Xi \in \overline{\mathcal{L}}$. The following is a restatement of a generalized Stone-Weierstrass Theorem (Prolla, 1994; Timofte, 2005) in the context of graph embeddings.

Theorem 1. *If the class* $\mathcal{L} \subseteq \mathcal{C}(\hat{\mathcal{G}}, \mathbb{R})$ *of graph embeddings is closed under linear combinations and products, and contains a non-zero constant embedding, then*

$$\overline{\mathcal{L}} = \big\{\Xi \in \mathcal{C}(\hat{\mathcal{G}}; \mathbb{R}) \bigm| \varrho_0(\mathcal{L}) \subseteq \varrho_0(\Xi)\big\}.$$

In other words, if $\mathcal{L}$ satisfies the conditions of the theorem, then only target embeddings Ξ can be approximated whose discriminative power is as most that of the class $\mathcal{L}$. Furthermore, to achieve universality $\mathcal{L}$ should have maximum discriminative power. Indeed, in that case $\overline{\mathcal{L}}$ consists precisely of all invariant continuous functions in $\mathcal{C}(\hat{\mathcal{G}}, \mathbb{R})$. We also remark that the discriminative power of $\mathcal{L}$ does not change when closing it under linear combinations and products. Hence, knowing $\varrho_0(\mathcal{L})$ provides all necessary information to understand what kind of target functions can be approximated.

2.2 Generalization Properties

Another motivation for studying the discriminative power of classes $\mathcal{L}$ of embeddings stems from its relationship to generalization properties of embeddings. Let us first recall what generalization means.

As can be found in any machine learning textbook (see e.g., Shalev-Shwartz and Ben-David (2014); Mohri et al. (2018)), instead of ERM one actually wants to learn p-vertex embeddings that minimize the generalization error (rather than empirical risk). The idea is that, intuitively, good embeddings in $\mathcal{L}$ are those that perform well relative to the loss function *on expectation* over all examples.

That is, one considers a probability distribution $\mathcal{P}$ over the examples $(G, \mathbf{v}, y)$ in $\mathcal{G} \times \mathcal{V}^p \times \mathcal{Y}$ and one would like to find elements $\xi^\star$ in $\mathcal{L}$ satisfying

$$\xi^\star \in \operatorname*{argmin}_{\xi \in \mathcal{L}} \mathbb{E}_{(G,\mathbf{v},y)\sim\mathcal{P}}\big[\ell\big(G, \mathbf{v}, y, \xi(G, \mathbf{v})\big)\big] \equiv \operatorname*{argmin}_{\xi \in \mathcal{L}} \mathcal{R}(\xi),$$

where $\mathcal{R}(\xi)$ denotes the *generalization error* or *risk* of ξ. The problem is that $\mathcal{P}$ is unknown or not available. Hence, as mentioned, in practice learners only can measure the empirical risk $\hat{\mathcal{R}}_{\mathcal{T}}(\xi)$ based on some sample training data $\mathcal{T}$ and perform ERM. It is of great interest to bound the difference between risk and empirical risk.

In the context of graph embeddings (and graph neural networks in particular) recent works considered bounding the generalization error using Rademacher complexity, the Vapnik-Chervonenkis (VC) dimension and Graph Neural Tangent Kernels (GNTK) (Esser et al., 2021; Garg et al., 2020; Liao et al., 2021; Maskey et al., 2022; Morris et al., 2023; Scarselli et al., 2018; Verma and Zhang, 2019; Du et al., 2019). We here describe the approach using VC dimension and its relation to the discriminative power (Morris et al., 2023). The starting point is the classical result (Shalev-Shwartz and Ben-David, 2014; Mohri et al., 2018) which tells that for any $\xi \in \mathcal{L}$, with high probability

$$\mathcal{R}(\xi) - \hat{\mathcal{R}}_{\mathcal{T}}(\xi) \leq \mathcal{O}\left(\sqrt{\frac{\log(m/\mathsf{VC}(\mathcal{L}))}{m/\mathsf{VC}(\mathcal{L})}}\right), \tag{1}$$

where m is the training data size and $\mathsf{VC}(\mathcal{L})$ is the VC dimension of $\mathcal{L}$ (Vapnik, 1995). For completeness, let us recall the notion of VC dimension. We assume for simplicity that the embeddings in $\mathcal{L}$ map inputs to $\{0,1\} \subset \mathbb{R}$. Furthermore, let $\mathcal{G}'$ be a subset of $\mathcal{G}$. Then, a collection $(G_1, \mathbf{v}_1), \ldots, (G_s, \mathbf{v}_s)$ of inputs in $\mathcal{G}' \times \mathcal{V}^p$ is said to be *shattered* by $\mathcal{L}$ if, for any boolean vector $\boldsymbol{\tau} \in \{0,1\}^s$, there exists an embedding $\xi_{\boldsymbol{\tau}}$ in $\mathcal{L}$ such that $\xi_{\boldsymbol{\tau}}(G_i, \mathbf{v}_i) = \tau_i$ for all $i = 1, \ldots, s$. The VC dimension of $\mathcal{L}$ on $\mathcal{G}'$ is then defined as

$$\mathsf{VC}_{\mathcal{G}'}(\mathcal{L}) \coloneqq \max\bigl\{s \bigm| \exists (G_1, \mathbf{v}_1), \ldots, (G_s, \mathbf{v}_s) \in \mathcal{G}' \times \mathcal{V}^p \text{ which can be shattered by } \mathcal{L}\bigr\}.$$

Hence, when the VC dimension of $\mathcal{L}$ is finite, the equation (1) allows to relate the size of the training set with generalization bounds.

We next relate the VC dimension of $\mathcal{L}$ on $\mathcal{G}'$ to the discriminative power of $\mathcal{L}$ on $\mathcal{G}'$. Suppose that $\mathcal{L}$ can shatter $(G_1, \mathbf{v}_1), \ldots, (G_s, \mathbf{v}_s)$. This implies that any pair of inputs $(G_i, \mathbf{v}_i)$ and $(G_j, \mathbf{v}_j)$ can be discriminated by $\mathcal{L}$, or in other words that $(G_i, \mathbf{v}_i, G_j, \mathbf{v}_j) \notin \varrho_p(\mathcal{L})$. Let us consider the number of equivalence classes in the quotient $(\mathcal{G}' \times \mathcal{V}^p)/_{\varrho_p(\mathcal{L})}$. Then clearly, this forms an upper bound on the number of inputs in $\mathcal{G}' \times \mathcal{V}^p$ that can be shattered. In other words:

Proposition 1 (Morris et al. (2023))**.** $\mathsf{VC}_{\mathcal{G}'}(\mathcal{L}) \leq |(\mathcal{G}' \times \mathcal{V}^p)/_{\varrho_p(\mathcal{L})}|$.

Hence, understanding $\varrho_p(\mathcal{L})$ provides insights into the VC dimension of $\mathcal{L}$ which in turn can be used to relate training data sizes m to good generalization properties.

2.3 Looking Ahead

We have seen that understanding the discriminative power $\varrho_p(\mathcal{L})$ is important for graph learning. So far, however, $\varrho_p(\mathcal{L})$ is nothing else but a definition. We will see shortly, however, that we can often relate $\varrho_p(\mathcal{L})$ to other well-studied equivalence relations. In particular, we will tie $\varrho_p(\mathcal{L})$ to distinguishability by Weisfeiler-Leman algorithms and finite variable fragments of first-order logic with counting quantifiers.

3 Embedding Methods

Those readers looking for some specific examples of embedding methods will be disappointed by now. And despite the title of this section, we also do not give concrete examples here. Instead, we will mention different methods without giving any details. The aim is to just show that there are many different methods out there.

For example, "early" methods in graph learning consist of kernel methods. For our purposes, a graph kernel is defined in terms of the inner product of graph embeddings. A plethora of graph kernels exist, e.g., based on shortest-paths (Borgwardt et al., 2005), random walks (Gärtner et al., 2003; Kang et al., 2012; Kashima et al., 2003; Sugiyama and Borgwardt, 2015), small subgraphs (Shervashidze et al., 2009; Kriege and Mutzel, 2012), local neighborhood information (Costa and Grave, 2010; Morris et al., 2017; Shervashidze et al., 2011), Laplacian information (Kondor and Pan, 2016), and matchings (Fröhlich et al., 2005; Kriege et al., 2016; Nikolentzos et al., 2017). We refer to Borgwardt et al. (2020) and Kriege et al. (2020) for thorough surveys.

Another important class of embedding methods consists of so-called graph neural networks (GNNs) (Gilmer et al., 2017; Scarselli et al., 2009). Most GNNs compute 1-vertex embeddings and graph embeddings. Notable examples of GNNs include those based on message-passing (Gilmer et al., 2017) such as e.g., (Duvenaud et al., 2015; Hamilton et al., 2017; Veličković et al., 2018; Xu et al., 2019; Morris et al., 2019). One also has GNNs based on spectral graph information such as, e.g., (Bruna et al., 2014; Defferrard et al., 2016; Gama et al., 2019; Kipf and Welling, 2017; Levie et al., 2019; Monti et al., 2017; Balcilar et al., 2021b). Some GNN architectures can employ vertex identifiers (Murphy et al., 2019; Vignac et al., 2020), use random features (Abboud et al., 2021; Dasoulas et al., 2020; Sato et al., 2021), equivariant graph polynomials (Puny et al., 2023), homomorphism and subgraph counts (Barceló et al., 2021; Bouritsas et al., 2020; Nguyen and Maehara, 2020), simplicial (Bodnar et al., 2021b) and cellular complexes (Bodnar et al., 2021a), persistent homology (Horn et al., 2022), random walks (Tönshoff et al., 2021; Martinkus et al., 2022), graph decompositions (Talak et al., 2021), relational (Barceló et al., 2022), distance (Li et al., 2020) and directional information (Beaini et al., 2021), subgraph information (Bevilacqua et al., 2022; Cotta et al., 2021; Feng et al., 2022; Frasca et al., 2022; Huang et al., 2023; Morris et al., 2021; Papp et al., 2021; Papp and Wattenhofer, 2022; Qian et al., 2022; Thiede et al., 2021; Wijesinghe and Wang, 2022; You et al., 2021; Zhang and Li, 2021; Zhao et al., 2022; Zhang et al., 2023a), and biconnectivity (Zhang et al., 2023b). Examples of graph neural network architectures using higher-order p-vertex embeddings for $p \geq 2$ are e.g., (Azizian and Lelarge, 2021; Geerts and Reutter, 2022; Maron et al., 2019a; Morris et al., 2019, 2020, 2022).

4 Weisfeiler-Leman and Logic

What kick-started the exploration of the discriminative power of embeddings were the two seminal papers (Morris et al., 2019; Xu et al., 2019). In those papers, the power of so-called Message-Passing Neural Networks (MPNN) (Gilmer et al., 2017) and that of the one-dimensional Weisfeiler-Leman algorithm ($\mathsf{1\text{-}WL}$) were shown to be the same. That is, they showed

$$\varrho_{0/1}(\mathsf{MPNN}) = \varrho_{0/1}(\mathsf{1\text{-}WL}),$$

where $\varrho_{0/1}(\mathsf{1\text{-}WL})$ indicates which graphs cannot be discriminated by $\mathsf{1\text{-}WL}$. This connection is important because $\varrho_{0/1}(\mathsf{1\text{-}WL})$ is studied extensively in the theoretical computer science community. In particular, $\varrho_0(\mathsf{1\text{-}WL})$ is known to coincide with $\varrho_0(\mathsf{C}_2)$, with C_2 the two-variable fragment of first-order logic with counting quantifiers (Cai et al., 1992). Also, $\varrho_1(\mathsf{1\text{-}WL}) = \varrho_1(\mathsf{GC}_2)$, where GC_2 is the guarded fragment of C_2 (Grohe, 2021). Inspired by this connection, more expressive higher-order GNNs (k-GNN) were proposed (Morris et al., 2019) satisfying

$$\varrho_{0/1}(k\text{-}\mathsf{GNN}) = \varrho_{0/1}(k\text{-}\mathsf{WL}),$$

where $\varrho_{0/1}(k\text{-}\mathsf{WL})$ refers to the discriminative power of the k-dimensional Weisfeiler-Leman algorithm. This in turn is equal to $\varrho_{0/1}(\mathsf{C}_{k+1})$ with C_{k+1} the $(k+1)$-variable fragment of first-order logic with counting quantifiers (Grohe, 2021). The use of k-WL and logic to assess the discriminative power of embedding methods has by now become standard (Barceló et al., 2020b; Bevilacqua et al., 2022; Geerts et al., 2021a; Azizian and Lelarge, 2021; Maron et al., 2019a; Morris et al., 2019, 2020, 2022; Qian et al., 2022; Aamand et al., 2022). We refer to the surveys (Grohe, 2020, 2021; Morris et al., 2021) for details. *In all these works, however, the analysis of the discriminative power of embedding methods requires proofs, geared at the specifics of the architecture. We next show that a unified, query language-based approach comes in handy.*

5 Graph Embedding Language (GEL)

When inspecting the different embedding methods listed in Section 3 it is not always immediately clear how they compare to each other and what their discriminative power is. One of the main reasons is the lack of a unifying language to describe the methods. When looking at those methods from a database perspective, we feel the urge to create a formal specification language for graph embedding methods. Of course, the fact that embedding methods operate on real numbers and perform linear algebra and other continuous operations scares off database theoreticians. There is, however, nothing to be afraid off.

Indeed, query languages for linear algebra have been considered before (Khamis et al., 2016; Brijder et al., 2019, 2018; Geerts, 2019, 2021; Geerts et al., 2021b,c; Barceló et al., 2022; Hutchison et al., 2017; Khamis et al., 2020). Their expressive power was studied precisely for understanding machine learning processes. Furthermore, the discriminative power of some of these languages is also related to 1-WL and 2-WL (Geerts, 2019, 2021). And these insights have been use to analyze the discriminative power of simple GNNs (Balcilar et al., 2021a). We here present a more general approach, proposed in Geerts and Reutter (2022).

More precisely, we define a general *graph embedding language*, referred to as GEL,[1] in which graph neural networks (and other graph learning methods) can be expressed. The language is very similar to the aggregate query languages considered for studying the expressive power of SQL (Libkin, 2003; Hella et al., 2001). This comes at no surprise since many of the linear algebra query languages have close ties to aggregate query languages (Brijder et al., 2019; Geerts et al., 2021b; Barceló et al., 2022), as also observed in Geerts et al. (2021c).

The syntax of gel We necessarily will need to support computations over real vectors. To this aim we parameterize GEL with a set Ω consisting of

- *functions* of the form $F : \mathbb{R}^d \to \mathbb{R}^{d'}$ for some $d, d' \in \mathbb{N}$; and
- *aggregate functions* of the form $\Theta : (2^{\mathbb{R}^d} \to \mathbb{N}) \to \mathbb{R}^{d'}$ for some $d, d' \in \mathbb{N}$, taking as input a multiset of vectors in $\mathbb{R}^d$ and returning a vector in $\mathbb{R}^{d'}$.

We inductively define expressions in $\text{GEL}(\Omega)$ and associate with each $\text{GEL}(\Omega)$ expression e its *dimension* $\mathsf{dim}(e)$ and its set $\mathsf{fv}(e)$ of *free variables*. Intuitively, an expression of dimension d generates a vector embedding in $\mathbb{R}^d$. As set of variables we take $X := \{x_1, x_2, x_3, \ldots\}$. We first treat the inductive cases corresponding to function and aggregation application. Let $e_1 \ldots, e_\ell$ be $\text{GEL}(\Omega)$ expressions of respective dimensions $d_1, \ldots, d_\ell$.

function application: If $F : \mathbb{R}^d \to \mathbb{R}^{d'}$ is a function in Ω and $d = d_1 + \cdots + d_\ell$, then

$$F(e_1, \ldots, e_\ell)$$

[1] The language does not have a well-established name. It is related to $\mathsf{sumMATLANG}$ Geerts et al. (2021b) and was called Tensor Language (or higher-order MPNNs) in Geerts and Reutter (2022).

is a $\mathrm{GEL}(\Omega)$ expression of dimension d' and $\mathsf{fv}\big(F(e_1,\ldots,e_\ell)\big)$ is $\mathsf{fv}(e_1)\cup\cdots\cup\mathsf{fv}(e_\ell)$.

aggregation: If $\Theta : (2^{\mathbb{R}^d} \to \mathbb{N}) \to \mathbb{R}^{d'}$ is an aggregate function in Ω and $d = d_1$, then

$$\Theta_{\bar{x}}[e_1|e_2],$$

where $\bar{x} = (x_1,\ldots,x_k)$ is a tuple of variables such that $\{x_1,\ldots,x_k\} \subseteq \mathsf{fv}(e_1)\cup\mathsf{fv}(e_2)$, is a $\mathrm{GEL}(\Omega)$ expression of dimension d' and $\mathsf{fv}\big(\Theta_{\bar{x}}[e_1|e_2]\big)$ is $\big(\mathsf{fv}(e_1)\cup\mathsf{fv}(e_2)\big)\setminus\{x_1,\ldots,x_k\}$.

The role of expression e_2 in $\Theta_{\bar{x}}[e_1|e_2]$ as a multiset selector will become clear when defining the semantics. Having dealt with the inductive cases, it remains to kick-start the creation of $\mathrm{GEL}(\Omega)$ expressions. Therefore we consider base cases corresponding to graph information and equality constraints between variables:

vertex label: $L_f(x_i)$ is a $\mathrm{GEL}(\Omega)$ expression with $\mathsf{fv}\big(L_f(x_i)\big)$ being $\{x_i\}$, for $f \in \mathbb{N}$.

edge: $E(x_i,x_j)$ is a $\mathrm{GEL}(\Omega)$ expression with $\mathsf{fv}\big(E(x_i,x_j)\big)$ being $\{x_i,x_j\}$.

dis/equality: $\mathbb{1}[x_i \,\mathsf{op}\, x_j]$ with $\mathsf{op} \in \{=,\neq\}$ is a $\mathrm{GEL}(\Omega)$ expression with $\mathsf{fv}\big(\mathbb{1}[x_i \,\mathsf{op}\, x_j]\big)$ being $\{x_i,x_j\}$.

All these base cases have dimension one. This concludes the definition of $\mathrm{GEL}(\Omega)$. We can extend this definition to accommodate for additional edge labels, if desired.

The semantics of gel In this section we let $\mathcal{G}$ consist of graphs specified by $G = \big([n],\mathbf{A},\mathbf{F}\big)$ with $[n] := \{1,2,\ldots,n\}$, for some $n \in \mathbb{N}$, the set of vertices, $\mathbf{A} \in \mathbb{R}^{n\times n}$ an adjacency matrix, and $\mathbf{F} \in \mathbb{R}^{n\times d_0}$ a matrix representing d_0-dimensional vertex features.

Let e be a $\mathrm{GEL}(\Omega)$ expression of dimension d and free variables $\mathsf{fv}(e) = \{x_i \mid i \in I\}$ for some (finite) index set $I \subseteq \mathbb{N}$. We define $\mathsf{max}(e)$ as the maximal index in I. We next give $\mathrm{GEL}(\Omega)$ expressions e a semantics in terms of the embedding ξ_e that they compute. That is, given input graph G and a p-vertex tuple $\mathbf{v} \in [n]^p$ with $p \geq \mathsf{max}(e)$, we define for $f = 1,\ldots,d_0$,

$$\begin{aligned}
\xi_{L_f(x_i)}(G,\mathbf{v}) &:= \mathbf{F}_{v_i,f} \in \mathbb{R}\\
\xi_{E(x_i,x_j)}(G,\mathbf{v}) &:= \mathbf{A}_{v_i,v_j} \in \mathbb{R}\\
\xi_{\mathbb{1}[x_i \,\mathsf{op}\, x_j]}(G,\mathbf{v}) &:= \begin{cases} 1 & \text{if } v_i \,\mathsf{op}\, v_j\\ 0 & \text{otherwise}\end{cases} \in \mathbb{R}\\
\xi_{F(e_1,\ldots,e_\ell)}(G,\mathbf{v}) &:= F\big(\xi_{e_1}(G,\mathbf{v}),\ldots,\xi_{e_\ell}(G,\mathbf{v})\big) \in \mathbb{R}^{d'},
\end{aligned}$$

for $e_1,\ldots,e_\ell$ of dimensions $d_1,\ldots,d_\ell$ such that $d = d_1+\cdots+d_\ell$ and $F : \mathbb{R}^d \to \mathbb{R}^{d'}$ in Ω. Finally, let $\Theta : (2^{\mathbb{R}^{d_1}} \to \mathbb{N}) \to \mathbb{R}^{d'}$ in Ω. We define $\xi_{\Theta_{\bar{x}}[e_1|e_2]}(G,\mathbf{v})$ as

$$\Theta\Big(\{\!\{\xi_{e_1}(G,\mathbf{u}) \mid \xi_{e_2}(G,\mathbf{u})\neq\mathbf{0}, \mathbf{v} \text{ extends } \mathbf{u}\}\!\}\Big) \in \mathbb{R}^{d'},$$

where we use $\{\!\{\,\}\!\}$ to denote multisets.

Perhaps only the semantics of aggregation deserves a bit more explanation. We are given G and $\mathbf{v}$. First, we minimally extend $\mathbf{v}$ into a tuple $\mathbf{u}$ such that $\xi_{e_1}(G,\mathbf{u})$ and $\xi_{e_2}(G,\mathbf{u})$ are defined. Then, for all such extensions $\mathbf{u}$ for which $\xi_{e_2}(G,\mathbf{u})$ does not evaluate to the zero vector $\mathbf{0} \in \mathbb{R}^{d_2}$, we put the vector $\xi_{e_1}(G, \mathbf{u}) \in \mathbb{R}^{d_1}$ in a multiset and then apply Θ, to finally obtain a vector in $\mathbb{R}^{d'}$. Intuitively, we perform some conditional aggregation.

As shown in Geerts and Reutter (2022), the $\mathrm{GEL}(\Omega)$ language is sufficiently powerful to encode many of the embedding methods listed in Section 3. Furthermore, whereas in the graph learning community it sometimes regarded challenging to define invariant embedding methods, every embedding expressible in $\mathrm{GEL}(\Omega)$ is automatically invariant. Clearly, the language is closely related to aggregate query languages. We will next use proof techniques developed for analyzing aggregate query languages (Libkin, 2003, 2004; Hella et al., 2001) to assess the discriminative power of classes of embeddings, hereby reaching our goal.

6 Climax

We now reach our climax since we have a general recipe at hand for assessing the discriminative power of classes $\mathcal{L}$ of embedding methods:

1. Show that every p-vertex embedding in $\mathcal{L}$ can be expressed in some fragment $\mathcal{E}$ of $\mathrm{GEL}(\Omega)$ for some Ω.
2. Figure out the discriminative power of $\varrho_p(\mathcal{E})$, i.e., all pairs of inputs that cannot be discriminated by a p-vertex embedding ξ_e for expressions $e \in \mathcal{E}$.

Of particular interest are the following fragments of $\mathrm{GEL}(\Omega)$:

$\mathrm{GEL}_k(\Omega)$: The fragment of $\mathrm{GEL}(\Omega)$ in which only k variables $X = \{x_1, \ldots, x_k\}$ can be used; and

$\mathrm{GGEL}_2(\Omega)$: the guarded fragment of $\mathrm{GEL}_2(\Omega)$ in which aggregation is of a restricted form. Intuitively, in $\mathrm{GGEL}_2(\Omega)$ only aggregation over neighbors is allowed. This can be enforced by only allowing $\Theta_{x_2}[e_1|E(x_1, x_2)]$ in the $\mathrm{GEL}_2(\Omega)$ expressions. See Geerts and Reutter (2022) for details.

We have the following characterization of the discriminative power of the embedding language fragments $\mathrm{GGEL}_2(\Omega)$ and $\mathrm{GEL}_k(\Omega)$.

Theorem 2 ((Geerts and Reutter, 2022))**.** *For any Ω containing addition and multiplication, and summation as aggregation function, we have:*

- $\varrho_0(\mathrm{GEL}_2(\Omega)) = \varrho_0(1\text{-}\mathsf{WL})$;
- $\varrho_1(\mathrm{GGEL}_2(\Omega)) = \varrho_1(1\text{-}\mathsf{WL})$*; and finally*
- $\varrho_p(\mathrm{GEL}_{k+1}(\Omega)) = \varrho_p(k\text{-}\mathsf{WL})$.

This result can be strengthened by bringing in the number of iterations of k-WL and connect this number to the aggregation depth of GEL expressions (Geerts and Reutter, 2022).

So what do we learn from this? Well, take any embedding method listed in Section 3 or take your own newly designed embedding method. The theorem implies that it suffices to "implement" your method in $\mathrm{GEL}(\Omega)$ using the smallest possible number of variables, say k. Then, you automatically obtain a k-WL bound on the discriminative power. As illustrated in

Geerts and Reutter (2022), casting methods in $\mathrm{GEL}_k(\Omega)$ is often a very simple task. Using this approach, many existing results can be easily recovered and new results can be obtained (Geerts and Reutter, 2022).

In summary, the use of the embedding language provides a simple way of obtaining upper bounds on the discriminative power of embedding methods. Interestingly, it allows to reason over embedding methods using query language techniques. We believe that thinking of embedding methods as queries makes it is easier to understand what graph learning is all about.

7 Open Problems

Clearly, there are many interesting problems associated with the embedding language $\mathrm{GEL}(\Omega)$ itself. For example, the impact on the discriminative power of the *choice of aggregation functions* in Ω is rather unexplored. Exceptions are Brijder et al. (2019); Geerts (2021) in which aggregation is fixed to be summation and functions correspond to specific linear algebra operators. In the context of graph embeddings (Corso et al., 2020; Rosenbluth et al., 2023) investigate the impact of the choice of aggregation functions for message-passing neural networks. Having a more detailed picture of the discriminative power of $\mathrm{GEL}_k(\Omega)$ in terms of Ω would be of direct use to analyze embedding methods with restricted forms of aggregation.

Theorem 2 provides *upper bounds* on the discriminative power of embedding methods. To obtain the best possible bound, however, the minimal number of variables needs to be used. How to tell whether a given $\mathrm{GEL}(\Omega)$ expressions is equivalent to one using k variables? In Geerts and Reutter (2022) a rudimentary notion of treewidth was introduced to this aim, inspired by the work on FAQs (Khamis et al., 2016). That is, $\mathrm{GEL}(\Omega)$ expressions of treewidth $k + 1$ are shown to be equivalent to expressions in $\mathrm{GEL}_k(\Omega)$. A more principled and fine-grained analysis would be of interest, however. Also related, can *lower bounds* be established in some principled way, e.g., using a notion of reduction as is done in complexity theory?

More research is needed to understand how properties of $\mathrm{GEL}(\Omega)$ fragments affect generalization properties. For example, a detailed analysis of the VC dimension of $\mathrm{GEL}(\Omega)$ fragments could be carried out (Morris et al., 2023). Alternatively, one can try to derive Graph Neural Tangent Kernels of fragments of $\mathrm{GEL}(\Omega)$ using techniques from Yang and Littwin (2021).

Inspired by new embedding methods, Qian et al. (2022); Morris et al. (2020, 2022) identify hierarchies, different from the k-WL hierarchy. To what $\mathrm{GEL}(\Omega)$ fragments do these new hierarchies correspond? And are there other hierarchies of interest than those based on k-WL?

Beyond discriminative power and approximation properties, one can consider uniform expressiveness (Geerts et al., 2022). That is, which target functions can be uniformly expressed by fragments of $\mathrm{GEL}(\Omega)$, and this without making compactness assumptions. This is the notion of expressiveness that we normally consider in query language research. We refer to Barceló et al. (2020b,a) in which the logical expressiveness of MPNNs (part of $\mathrm{GEL}_2(\Omega)$) was studied. More recently, Grohe (2023) showed tight connections between GNNs' expressiveness and circuit complexity. Whether and how these results can be generalized to richer $\mathrm{GEL}(\Omega)$ fragments is wide open.

Not all methods from Section 3 have been analyzed yet and new methods are proposed in every machine learning conference. It may be that $\mathrm{GEL}(\Omega)$ needs to be revised or updated with new functionalities to stay relevant for new and upcoming methods.

There are many more problems one could consider. We hope that by viewing embedding methods as queries in $\mathrm{GEL}(\Omega)$ or some other formalism, an additional bridge has been made between graph learning and database research. We look forward to new insights in graph learning originating from database theory!

References

Anders Aamand, Justin Y Chen, Piotr Indyk, Shyam Narayanan, Ronitt Rubinfeld, Nicholas Schiefer, Sandeep Silwal, and Tal Wagner. 2022. Exponentially Improving the Complexity of Simulating the Weisfeiler-Lehman Test with Graph Neural Networks. In *Advances in Neural Information Processing Systems*.

Ralph Abboud, İsmail İlkan Ceylan, Martin Grohe, and Thomas Lukasiewicz. 2021. The Surprising Power of Graph Neural Networks with Random Node Initialization. In *Joint Conference on Artificial Intelligence*. 2112–2118.

Waïss Azizian and Marc Lelarge. 2021. Characterizing the Expressive Power of Invariant and Equivariant Graph Neural Networks. In *International Conference on Learning Representations*.

Muhammet Balcilar, Pierre Héroux, Benoit Gaüzère, Pascal Vasseur, Sébastien Adam, and Paul Honeine. 2021a. Breaking the Limits of Message Passing Graph Neural Networks. In *International Conference on Machine Learning*. 599–608.

Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. 2021b. Analyzing the Expressive Power of Graph Neural Networks in a Spectral Perspective. In *International Conference on Learning Representations*.

Pablo Barceló, Mikhail Galkin, Christopher Morris, and Miguel Romero Orth. 2022. Weisfeiler and Leman Go Relational. In *The First Learning on Graphs Conference*.

Pablo Barceló, Floris Geerts, Juan L. Reutter, and Maksimilian Ryschkov. 2021. Graph Neural Networks with Local Graph Parameters. In *Advances in Neural Information Processing Systems*. 25280–25293.

Pablo Barceló, Nelson Higuera, Jorge Pérez, and Bernardo Subercaseaux. 2022. On the Expressiveness of LARA: A Proposal for Unifying Linear and Relational Algebra. *Theoretical Computer Science* 935 (2022), 105–127.

Pablo Barceló, Egor V. Kostylev, Mikaël Monet, Jorge Pérez, Juan L. Reutter, and Juan Pablo Silva. 2020a. The Expressive Power of Graph Neural Networks as a Query Language. *SIGMOD Rec.* 49, 2 (2020), 6–17.

Pablo Barceló, Egor V. Kostylev, Mikaël Monet, Jorge Pérez, Juan L. Reutter, and Juan Pablo Silva. 2020b. The Logical Expressiveness of Graph Neural Networks. In *International Conference on Learning Representations*.

Dominique Beaini, Saro Passaro, Vincent Létourneau, William L. Hamilton, Gabriele Corso, and Pietro Lió. 2021. Directional Graph Networks. In *International Conference on Machine Learning*. 748–758.

Beatrice Bevilacqua, Fabrizio Frasca, Derek Lim, Balasubramaniam Srinivasan, Chen Cai, Gopinath Balamurugan, Michael M. Bronstein, and Haggai Maron. 2022. Equivariant Subgraph Aggregation Networks. In *International Conference on Learning Representations*.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yu Guang Wang, Pietro Liò, Guido Montúfar, and Michael M. Bronstein. 2021a. Weisfeiler and Lehman Go Cellular: CW Networks. In *Advances in Neural Information Processing Systems*. 2625–2640.

Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F. Montúfar, Pietro Lió, and Michael M. Bronstein. 2021b. Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks. In *International Conference on Machine Learning*. 1026–1037.

Karsten M. Borgwardt, M. Elisabetta Ghisu, Felipe Llinares-López, Leslie O'Bray, and Bastian Rieck. 2020. Graph Kernels: State-of-the-Art and Future Challenges. *Foundations and Trends in Machine Learning* 13, 5–6 (2020).

Karsten M. Borgwardt, Cheng Soon Ong, Stefan Schönauer, S. V. N. Vishwanathan, Alex J. Smola, and Hans-Peter Kriegel. 2005. Protein Function Prediction via Graph Kernels. *Bioinformatics* 21, Supplement 1 (2005), 47–56.

Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M. Bronstein. 2020. Improving graph neural network expressivity via subgraph isomorphism counting. (2020). arXiv:2006.09252

Robert Brijder, Floris Geerts, Jan Van den Bussche, and Timmy Weerwag. 2018. On the Expressive Power of Query Languages for Matrices. In *International Conference on Database Theory*. 10:1–10:17.

Robert Brijder, Floris Geerts, Jan Van den Bussche, and Timmy Weerwag. 2019. On the Expressive Power of Query Languages for Matrices. *ACM TODS* 44, 4 (2019), 15:1–15:31.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral Networks and Deep Locally Connected Networks on Graphs. In *International Conference on Learning Representation*.

Jin-yi Cai, Martin Fürer, and Neil Immerman. 1992. An Optimal Lower Bound on the Number of Variables for Graph Identifications. *Combinatorica* 12, 4 (1992), 389–410.

Zhengdao Chen, Soledad Villar, Lei Chen, and Joan Bruna. 2019. On the Equivalence Between Graph Isomorphism Testing and Function Approximation With GNNs. In *Advances in Neural Information Processing Systems*. 15868–15876.

Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. 2020. Principal Neighbourhood Aggregation for Graph Nets. In *Advances in Neural Information Processing Systems*.

Fabrizio Costa and Kurt De Grave. 2010. Fast Neighborhood Subgraph Pairwise Distance Kernel. In *International Conference on Machine Learning*. 255–262.

Leonardo Cotta, Christopher Morris, and Bruno Ribeiro. 2021. Reconstruction for Powerful Graph Representations. In *Advances in Neural Information Processing Systems*. 1713–1726.

George Cybenko. 1989. Approximation by Superpositions of a Sigmoidal Dunction. *Mathematics of Control, Signals, and Systems* 2, 4 (1989), 303–314.

George Dasoulas, Ludovic Dos Santos, Kevin Scaman, and Aladin Virmaux. 2020. Coloring Graph Neural Networks for Node Disambiguation. In *International Joint Conference on Artificial Intelligence*. 2126–2132.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems*. 3837–3845.

Simon S. Du, Kangcheng Hou, Ruslan Salakhutdinov, Barnabás Póczos, Ruosong Wang, and Keyulu Xu. 2019. Graph Neural Tangent Kernel: Fusing Graph Neural Networks with Graph Kernels. In *Advances in Neural Information Processing Systems*. 5724–5734.

David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. 2015. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Advances in Neural Information Processing Systems*. 2224–2232.

Pascal Mattia Esser, Leena C. Vankadara, and Debarghya Ghoshdastidar. 2021. Learning Theory Can (Sometimes) Explain Generalisation in Graph Neural Networks. In *Advances in Neural Information Processing Systems*. 27043–27056.

Jiarui Feng, Yixin Chen, Fuhai Li, Anindya Sarkar, and Muhan Zhang. 2022. How Powerful are K-hop Message Passing Graph Neural Networks. In *Advances in Neural Information Processing Systems*.

Fabrizio Frasca, Beatrice Bevilacqua, Michael M. Bronstein, and Haggai Maron. 2022. Understanding and Extending Subgraph GNNs by Rethinking Their Symmetries. (2022).

Holger Fröhlich, Jörg K. Wegner, Florian Sieker, and Andreas Zell. 2005. Optimal Assignment Kernels for Attributed Molecular Graphs. In *International Conference on Machine Learning*. 225–232.

Fernando Gama, Antonio G. Marques, Geert Leus, and Alejandro Ribeiro. 2019. Convolutional Neural Network Architectures for Signals Supported on Graphs. *IEEE Transactions on Signal Processing* 67, 4 (2019), 1034–1049.

Vikas K. Garg, Stefanie Jegelka, and Tommi S. Jaakkola. 2020. Generalization and Representational Limits of Graph Neural Networks. In *International Conference on Machine Learning*. 3419–3430.

Thomas Gärtner, Peter A. Flach, and Stefan Wrobel. 2003. On Graph Kernels: Hardness Results and Efficient Alternatives. In *Conference on Computational Learning Theory*. 129–143.

Floris Geerts. 2019. On the Expressive Power of Linear Algebra on Graphs. In *International Conference on Database Theory*. 7:1–7:19.

Floris Geerts. 2021. On the Expressive Power of Linear Algebra on Graphs. *Theory Comput. Syst.* 65, 1 (2021), 179–239.

Floris Geerts, Filip Mazowiecki, and Guillermo A. Pérez. 2021a. Let's Agree to Degree: Comparing Graph Convolutional Networks in the Message-Passing Framework. In *International Conference on Machine Learning*. 3640–3649.

Floris Geerts, Thomas Muñoz, Cristian Riveros, and Domagoj Vrgoč. 2021b. Expressive Power of Linear Algebra Query Languages. In *Symposium on Principles of Database Systems*.

Floris Geerts, Thomas Muñoz, Cristian Riveros, Jan Van den Bussche, and Domagoj Vrgoc. 2021c. Matrix Query Languages. *SIGMOD Rec.* 50, 3 (2021), 6–19.

Floris Geerts and Juan L. Reutter. 2022. Expressiveness and Approximation Properties of Graph Neural Networks. In *International Conference on Learning Representations*.

Floris Geerts, Jasper Steegmans, and Jan Van den Bussche. 2022. On the Expressive Power of Message-Passing Neural Networks as Global Feature Map Transformers. In *International Symposium on Foundations of Information and Knowledge Systems*. 20–34.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural Message Passing for Quantum Chemistry. In *International Conference on Machine Learning*. 1263–1272.

Martin Grohe. 2020. Word2vec, Node2vec, Graph2vec, X2vec: Towards a Theory of Vector Embeddings of Structured Data. In *Symposium on Principles of Database Systems*. 1–16.

Martin Grohe. 2021. The Logic of Graph Neural Networks. In *Symposium on Logic in Computer Science*. 1–17.

Martin Grohe. 2023. The Descriptive Complexity of Graph Neural Networks. (2023). arXiv:2303.04613

William L. Hamilton. 2020. *Graph Representation Learning*. Morgan & Claypool Publishers.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*. 1024–1034.

Lauri Hella, Leonid Libkin, Juha Nurmonen, and Limsoon Wong. 2001. Logics with Aggregate Operators. *J. ACM* 48, 4 (2001), 880–907.

Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten M. Borgwardt. 2022. Topological Graph Neural Networks. In *International Conference on Learning Representations*.

Kurt Hornik. 1991. Approximation Capabilities of Multilayer Feedforward Networks. *Neural Networks* 4, 2 (1991), 251–257.

Yinan Huang, Xingang Peng, Jianzhu Ma, and Muhan Zhang. 2023. Boosting the Cycle Counting Power of Graph Neural Networks with I^2-GNNs. (2023).

Dylan Hutchison, Bill Howe, and Dan Suciu. 2017. LaraDB: A Minimalist Kernel for Linear and Relational Algebra Computation. In *ACM SIGMOD Workshop on Algorithms and Systems for MapReduce and Beyond*. 2:1–2:10.

Stefanie Jegelka. 2022. Theory of Graph Neural Networks: Representation and Learning. (2022). arXiv:2204.07697

U Kang, Hanghang Tong, and Jimeng Sun. 2012. Fast Random Walk Graph Kernel. In *SIAM International Conference on Data Mining*. 828–838.

Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. 2003. Marginalized Kernels Between Labeled Graphs. In *International Conference on Machine Learning*. 321–328.

Nicolas Keriven and Gabriel Peyré. 2019. Universal Invariant and Equivariant Graph Neural Networks. In *Advances in Neural Information Processing Systems*. 7090–7099.

Mahmoud Abo Khamis, Ryan R. Curtin, Benjamin Moseley, Hung Q. Ngo, Xuanlong Nguyen, Dan Olteanu, and Maximilian Schleich. 2020. Functional Aggregate Queries with Additive Inequalities. *ACM TODS* 45, 4, Article 17 (2020).

Mahmoud Abo Khamis, Hung Q. Ngo, and Atri Rudra. 2016. FAQ: Questions Asked Frequently. In *Symposium on Principles of Database Systems*. 13–28.

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Risi Kondor and Horace Pan. 2016. The Multiscale Laplacian Graph Kernel. In *Advances in Neural Information Processing Systems*. 2982–2990.

Nils M. Kriege, Pierre-Louis Giscard, and Richard C. Wilson. 2016. On Valid Optimal Assignment Kernels and Applications to Graph Classification. In *Advances in Neural Information Processing Systems*. 1615–1623.

Nils M. Kriege, Fredrik D. Johansson, and Christopher Morris. 2020. A Survey on Graph Kernels. *Applied Network Science* 5, 1 (2020).

Nils M. Kriege and Petra Mutzel. 2012. Subgraph Matching Kernels for Attributed Graphs. In *International Conference on Machine Learning*.

Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. 1993. Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate any Function. *Neural Networks* 6, 6 (1993), 861–867.

Ron Levie, Federico Monti, Xavier Bresson, and Michael M. Bronstein. 2019. CayleyNets: Graph Convolutional Neural Networks With Complex Rational Spectral Filters. *IEEE Transactions on Signal Processing* 67, 1 (2019), 97–109.

Pan Li, Yanbang Wang, Hongwei Wang, and Jure Leskovec. 2020. Distance Encoding: Design Provably More Powerful Neural Networks for Graph Representation Learning. In *Advances in Neural Information Processing Systems*.

Renjie Liao, Raquel Urtasun, and Richard S. Zemel. 2021. A PAC-Bayesian Approach to Generalization Bounds for Graph Neural Networks. In *International Conference on Learning Representations*.

Leonid Libkin. 2003. Expressive Power of SQL. *Theoretical Computer Science* 296 (2003), 379–404.

Leonid Libkin. 2004. *Elements of Finite Model Theory.* Springer.

Yao Ma and Jiliang Tang. 2021. *Deep Learning on Graphs.* Cambridge University Press.

Takanori Maehara and Hoang NT. 2019. A Simple Proof of the Universality of Invariant/Equivariant Graph Neural Networks. (2019). arXiv:1910.03802

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. 2019a. Provably Powerful Graph Networks. In *Advances in Neural Information Processing Systems.* 2153–2164.

Haggai Maron, Ethan Fetaya, Nimrod Segol, and Yaron Lipman. 2019b. On the Universality of Invariant Networks. In *International Conference on Machine Learning.* 4363–4371.

Karolis Martinkus, Pál András Papp, Benedikt Schesch, and Roger Wattenhofer. 2022. Agent-based Graph Neural Networks. (2022). arXiv:2206.11010

Sohir Maskey, Ron Levie, Yunseok Lee, and Gitta Kutyniok. 2022. Generalization Analysis of Message Passing Neural Networks on Large Random Graphs. In *Advances in Neural Information Processing Systems.*

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. *Foundations of Machine Learning.* MIT Press.

Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. 2017. Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs. In *IEEE Conference on Computer Vision and Pattern Recognition.* 5425–5434.

Christopher Morris, Floris Geerts, Jan Tönshoff, and Martin Grohe. 2023. WL meet VC. (2023). arXiv:2301.11039

Christopher Morris, Kristian Kersting, and Petra Mutzel. 2017. Glocalized Weisfeiler-Lehman Kernels: Global-Local Feature Maps of Graphs. In *IEEE International Conference on Data Mining.* 327–336.

Christopher Morris, Yaron Lipman, Haggai Maron, Bastian Rieck, Nils M. Kriege, Martin Grohe, Matthias Fey, and Karsten M. Borgwardt. 2021. Weisfeiler and Leman go Machine Learning: The Story so far. (2021). arXiv:2112.09992

Christopher Morris, Gaurav Rattan, Sandra Kiefer, and Siamak Ravanbakhsh. 2022. SpeqNets: Sparsity-aware permutation-equivariant graph networks. In *International Conference on Machine Learning.* 16017–16042.

Christopher Morris, Gaurav Rattan, and Petra Mutzel. 2020. Weisfeiler and Leman Go Sparse: Towards Higher-Order Graph Embeddings. In *Advances in Neural Information Processing Systems.*

Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. 2019. Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks. In *AAAI Conference on Artificial Intelligence.* 4602–4609.

Ryan L. Murphy, Balasubramaniam Srinivasan, Vinayak A. Rao, and Bruno Ribeiro. 2019. Relational Pooling for Graph Representations. In *International Conference on Machine Learning*. 4663–4673.

Hoang Nguyen and Takanori Maehara. 2020. Graph Homomorphism Convolution. In *International Conference on Machine Learning*. 7306–7316.

Giannis Nikolentzos, Polykarpos Meladianos, and Michalis Vazirgiannis. 2017. Matching Node Embeddings for Graph Similarity. In *AAAI Conference on Artificial Intelligence*. 2429–2435.

Pál András Papp, Karolis Martinkus, Lukas Faber, and Roger Wattenhofer. 2021. DropGNN: Random Dropouts Increase the Expressiveness of Graph Neural Networks. In *Advances in Neural Information Processing Systems*.

Pál András Papp and Roger Wattenhofer. 2022. A Theoretical Comparison of Graph Neural Network Extensions. In *International Conference on Machine Learning*. 17323–17345.

João Bosco Prolla. 1994. On the Weierstrass-Stone Theorem. *Journal of Approximation Theory* 78, 3 (1994), 299–313.

Omri Puny, Derek Lim, Bobak Toussi Kiani, Haggai Maron, and Yaron Lipman. 2023. Equivariant Polynomials for Graph Neural Networks. (2023). arXiv:2302.11556

Chendi Qian, Gaurav Rattan, Floris Geerts, Christopher Morris, and Mathias Niepert. 2022. Ordered Subgraph Aggregation Networks. In *Advances in Neural Information Processing Systems*.

Eran Rosenbluth, Jan Tönshoff, and Martin Grohe. 2023. Some Might Say All You Need Is Sum. (2023). arXiv:2302.11603

Ryoma Sato, Makoto Yamada, and Hisashi Kashima. 2021. Random Features Strengthen Graph Neural Networks. In *SIAM International Conference on Data Mining*. 333–341.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks* 20, 1 (2009), 61–80.

Franco Scarselli, Ah Chung Tsoi, and Markus Hagenbuchner. 2018. The Vapnik-Chervonenkis Dimension of Graph and Recursive Neural Networks. *Neural Networks* (2018), 248–259.

Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

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

Nino Shervashidze, S. V. N. Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten M. Borgwardt. 2009. Efficient Graphlet Kernels for Large Graph Comparison. In *International Conference on Artificial Intelligence and Statistics*. 488–495.

Mahito Sugiyama and Karsten M. Borgwardt. 2015. Halting in Random Walk Kernels. In *Advances in Neural Information Processing Systems*. 1639–1647.

Rajat Talak, Siyi Hu, Lisa Peng, and Luca Carlone. 2021. Neural Trees for Learning on Graphs. In *Advances in Neural Information Processing Systems*. 26395–26408.

Erik H. Thiede, Wenda Zhou, and Risi Kondor. 2021. Autobahn: Automorphism-based Graph Neural Nets. In *Advances in Neural Information Processing Systems*. 29922–29934.

Vlad Timofte. 2005. Stone–Weierstrass Theorems Revisited. *Journal of Approximation Theory* 136, 1 (2005), 45–59.

Jan Tönshoff, Martin Ritzert, Hinrikus Wolf, and Martin Grohe. 2021. Graph Learning with 1D Convolutions on Random Walks. (2021). arXiv:2102.08786

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

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Saurabh Verma and Zhi-Li Zhang. 2019. Stability and Generalization of Graph Convolutional Neural Networks. In *International Conference on Knowledge Discovery & Data Mining*. 1539–1548.

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

Asiri Wijesinghe and Qing Wang. 2022. A New Perspective on "How Graph Neural Networks Go Beyond Weisfeiler-Lehman?". In *International Conference on Learning Representations*.

Lingfei Wu, Peng Cui, Jian Pei, and Liang Zhao (Eds.). 2022. *Graph Neural Networks: Foundations, Frontiers, and Applications*. Springer.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *International Conference on Learning Representations*.

Greg Yang and Etai Littwin. 2021. Tensor Programs IIb: Architectural Universality Of Neural Tangent Kernel Training Dynamics. In *International Conference on Machine Learning*. 11762–11772.

Jiaxuan You, Jonathan Gomes-Selman, Rex Ying, and Jure Leskovec. 2021. Identity-aware Graph Neural Networks. In *AAAI Conference on Artificial Intelligence*. 10737–10745.

Bohang Zhang, Guhao Feng, Yiheng Du, Di He, and Liwei Wang. 2023a. A Complete Expressiveness Hierarchy for Subgraph GNNs via Subgraph Weisfeiler-Lehman Tests. (2023). arXiv:2302.07090

Bohang Zhang, Shengjie Luo, Liwei Wang, and Di He. 2023b. Rethinking the Expressive Power of GNNs via Graph Biconnectivity. (2023). arXiv:2301.09505

Muhan Zhang and Pan Li. 2021. Nested Graph Neural Networks. In *Advances in Neural Information Processing Systems*. 15734–15747.

Lingxiao Zhao, Wei Jin, Leman Akoglu, and Neil Shah. 2022. From Stars to Subgraphs: Uplifting Any GNN with Local Structure Awareness. In *International Conference on Learning Representations*.