Logical Characterizations of GNNs with Mean Aggregation

Moritz Schönherr, Carsten Lutz
Department of Computer Science, Leipzig University, Germany
Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), Dresden/Leipzig, Germany
{schoenherr, clu}@informatik.uni-leipzig.de

Abstract

We study the expressive power of graph neural networks (GNNs) with mean as the aggregation function, with the following results. In the non-uniform setting, such GNNs have exactly the same expressive power as ratio modal logic, which has modal operators expressing that at least a certain ratio of the successors of a vertex satisfies a specified property. In the uniform setting, the expressive power relative to MSO is exactly that of modal logic, and thus identical to the (absolute) expressive power of GNNs with max aggregation. The proof, however, depends on constructions that are not satisfactory from a practical perspective. This leads us to making the natural assumptions that combination functions are continuous and classification functions are thresholds. The resulting class of GNNs with mean aggregation turns out to be much less expressive: relative to MSO and in the uniform setting, it has the same expressive power as alternation-free modal logic. This is in contrast to the expressive power of GNNs with max and sum aggregation, which is not affected by these assumptions.

1 Introduction

Graph neural networks (GNNs) are a family of deep-learning architectures that act directly on graphs, removing the need for prior serialization or encoding, in this way ensuring isomorphism invariance (Scarselli et al. 2009; Wu et al. 2021; Zhou et al. 2020). GNNs have been applied successfully in domains ranging from molecular property prediction and drug discovery (Bongini, Bianchini, and Scarselli 2021) to fraud detection (Deng and Hooi 2021) and e-commerce recommendation (Wu et al. 2022), traffic forecasting (Jiang and Luo 2022), and physics simulation (Shlomi, Battaglia, and Vlimant 2020). Numerous GNN variants exist, all sharing central features such as message passing and the iterative update of vertex embeddings, and an important foundational question is which properties a given GNN architecture can represent. Beginning with Barceló et al. (2020) and Grohe (2021), an expanding body of work has addressed this question by associating the expressive power of GNNs with that of various logical formalisms. Such logical characterizations may be used to guide the choice of a GNN model for a specific task at hand and to reveal potential gaps between a model's representational capacity and the task's requirements.

	Non-Uniform	Uniform wrt. MSO	Uniform absolute
$\text{Mean}^{c,t}$	RML Th. 2, 3, 5	AFML Th. 13, 14, 15	$>$ AFML Th. 16
Mean	RML Th. 2, 3, 5	ML Cor. 3, Th. 10, 11	$>$ ML Th. 16
Sum	GML Th. 3, 5	$\text{GML}^{\dagger}$	$> \text{GML}^{\ddagger}$
Max	ML Th. 4, Th. 6	ML Th. 4, Th. 6	ML Th. 4, Th. 6

Table 1: Overview of results, with $\cdot^{\dagger}$ from (Barceló et al. 2020) and $\cdot^{\ddagger}$ from (Benedikt et al. 2024).

Barceló et al. (2020) focus on GNNs with constant iteration depth, sum as the aggregation function, and truncated ReLU as the activation function. One main result is that, relative to first-order logic (FO), the expressive power of GNNs as a vertex classifier is exactly that of graded modal logic (GML). In other articles, such as (Tena Cucala et al. 2023; Tena Cucala and Cuenca Grau 2024), sum aggregation is replaced with max aggregation, and GML is replaced with certain restrictions of non-recursive datalog. The main purpose of the current paper is to characterize the expressive power of GNNs as a vertex classifier when the aggregation function is arithmetic mean. This is in fact a very natural and important case. Influential graph learning systems such as GraphSAGE (Hamilton, Ying, and Leskovec 2017) and Pinterest's web-scale recommender PinSAGE (Ying et al. 2018) use mean aggregation, and the popular Graph Convolutional Networks (GCNs) use a weighted version of mean (Kipf and Welling 2017). Moreover, mean aggregation is amenable to random sampling and thus a good choice for graphs in which vertices may have very large degree. Our characterizations are in terms of suitable versions of modal logic. We also provide several new observations regarding GNNs with max and sum aggregation.

We consider the expressive power of GNNs in two different settings that have both received attention in the literature. In the uniform setting, a GNN model $\mathcal{M}$ has the same expressive power as a logic $\mathcal{L}$ if $\mathcal{M}$ and $\mathcal{L}$ can express exactly the same vertex classifiers, across all graphs. In the non-uniform setting, we only require that for every $n \geq 1$, $\mathcal{M}$ and $\mathcal{L}$ can express exactly the same vertex classifiers across all graphs with n vertices. In the uniform case, we follow

Barceló et al. (2020) in studying the expressive power relative to FO, that is, we restrict our attention to GNNs that express a vertex property definable by an FO formula. In fact we use monadic second-order logic (MSO) in place of FO, based on the observation from (Ahvonen et al. 2024) that every GNN that expresses an MSO-definable property actually expresses an FO-definable property. In the non-uniform case, we seek absolute characterizations that are independent of any background logic.

In the non-uniform case, we prove that GNNs with mean aggregation have the same expressive power as ratio modal logic (RML) which provides modal operators $\Diamond^{\geq r}\varphi$ and $\Diamond^{>r}\varphi$ expressing that the fraction of successors that satisfy φ is at least r (resp. exceeds r). This should be contrasted with GNNs based on sum aggregation and max aggregation, which have the same expressive power as GML and modal logic (ML), respectively. While these latter results are not surprising given existing work, we are not aware that they have been proved anywhere in this form, and we provide proof details here. All results are summarized in Table 1. The expressive equivalence between Max-GNNs and ML even holds in the uniform case. In the non-uniform setting, GML is strictly more expressive than RML, which in turn is strictly more expressive than ML. Our results thus reflect the known fact that Sum-GNNs are (non-uniformly) strictly more expressive than Mean-GNNs, which are in turn strictly more expressive than Max-GNNs (Xu et al. 2019).

The uniform setting turns out to be significantly more subtle and interesting. The fragment of RML that is expressible in MSO is exactly ML. Given our results from the non-uniform case, one may thus expect that in the uniform case and relative to MSO, Mean-GNNs have the same expressive power as ML. We prove that this is indeed the case. It follows that (uniformly and relative to MSO) Mean-GNNs have the same expressive power as Max-GNNs, but are strictly less expressive than Sum-GNNs. However, the translation of ML formulas into Mean-GNNs is unsatisfactory from a practical standpoint: one may either use a combination function that is not differentiable, and in fact not even continuous, or a rather unnatural classification function (in our translation, the rational numbers are classified as 0 and the irrational numbers as 1). This leads us to study Mean-GNNs under the natural assumptions that (i) the combination functions are continuous and (ii) the classification function is a threshold function. This case is denoted $\text{Mean}^{c,t}$ in Table 1.

We prove that assumptions (i) and (ii) result in a significant drop of expressive power: relative to MSO, GNNs with mean aggregation now have the same expressive power as *alternation-free* modal logic (AFML) in which modal diamonds and boxes cannot be mixed. We believe that from a practical perspective, AFML provides a more realistic characterization of the expressive power of Mean-GNNs than ML (relative to MSO). It is also interesting to note that assumptions (i) and (ii) have no impact on expressive power in the cases of sum aggregation and max aggregation.

Throughout the paper, we also consider a 'simple' version of GNNs in which the combination function is a feedforward neural network without hidden layers. We then pay special attention to the activation function. Notably, we prove that the non-uniform results for mean and sum aggregation stated above hold for all continuous non-polynomial activation functions. We also prove that our result relating Mean-GNNs and AFML in the uniform case also holds for ReLU, truncated ReLU, and sigmoid activation. This is in contrast to Sum-GNNs where transitioning from (truncated) ReLU to sigmoid reduces the expressive power in the uniform case (Khalife and Tonelli-Cueto 2025).

Detailed proofs are available in (Schönherr and Lutz 2025).

Related Work. We start with the uniform setting. The link between GNNs and modal logic was established in (Barceló et al. 2020), relative to FO. The expressive power of Sum-GNNs beyond FO was studied in (Benedikt et al. 2024), and linked to modal logic with Presburger quantifiers. In (Tena Cucala and Cuenca Grau 2024), Max-GNNs are translated into non-recursive datalog programs with negation-as-failure that adhere to a certain tree-shape. This formalism is closely related to modal logic. Recently, GNNs with transformer layers have been characterized in terms of modal logic (Ahvonen et al. 2026). Such layers are closely related to mean aggregation. Sound logical explanations for a certain kind of monotone Mean-GNN have been studied in (Morris and Horrocks 2025). In the non-uniform setting, Grohe (2024) shows that GNNs have the same expressive power as an extension of GFO+C, the guarded fragment of FO with counting capabilities, with built-in relations. Some related results are in (Grohe and Rosenbluth 2024). Without reference to logic, the expressive power of different aggregation functions is studied in (Xu et al. 2019; Rosenbluth, Toenshoff, and Grohe 2023).

2 Preliminaries

We use $\mathbb{N}$ and $\mathbb{N}^+$ to denote the set of all non-negative and positive integers, respectively. For $n \geq 1$, we write $[n]$ for the set $\{1, \ldots, n\}$. With $\mathcal{M}(X)$ we mean the set of all finite multisets over the set X, that is, the set of functions $X \to \mathbb{N}$ where all but finitely many elements of X are mapped to 0. For a vector $\bar{x} \in \mathbb{R}^\delta$ we use $x_1, \ldots, x_\delta$ or, when more readable, $(\bar{x})_1, \ldots, (\bar{x})_\delta$ to refer to its components.

Graph Neural Networks. Let $\Pi = \{P_1, \ldots, P_n\}$ be a finite set of *vertex labels*. A *(Π-labeled directed) graph* is a tuple $G = (V, E, \pi)$ that consists of a set of vertices V, a set of edges $E \subseteq V \times V$, and a vertex labeling function $\pi : V \to 2^\Pi$. Unless noted otherwise, all graphs in this article are finite. For easier reference, we may write V^G for V, E^G for E, and π^G for π. The *neighborhood* of a vertex v in a graph G is the set of its successors, formally $\mathcal{N}(G, v) = \{u \mid (v, u) \in E\}$. A *pointed graph* is a pair (G, v) with G a graph and $v \in V^G$ a distinguished vertex. A *(vertex) property* is a class of pointed graphs, over a common set of labels Π, that is closed under isomorphism.

A *graph neural network (GNN)* on Π-labeled graphs is a tuple

$$\mathcal{G} = (L, \{\mathsf{AGG}^\ell\}_{\ell \in [L]}, \{\mathsf{COM}^\ell\}_{\ell \in [L]}, \mathsf{CLS})$$

where $L \geq 1$ is the number of layers and for each $\ell \in [L]$, $\mathsf{AGG}^\ell : \mathcal{M}(\mathbb{R}^{\delta^{\ell-1}}) \to \mathbb{R}^{\delta^{\ell-1}}$ is an *aggregation function*, $\mathsf{COM}^\ell : \mathbb{R}^{\delta^{\ell-1}} \times \mathbb{R}^{\delta^{\ell-1}} \to \mathbb{R}^{\delta^\ell}$ is a *combination function*, and $\mathsf{CLS} : \mathbb{R}^{\delta^L} \to \{0,1\}$ is a *classification function*. We call $\delta^{\ell-1}$ the *input dimension* of layer ℓ and δ^ℓ the *output dimension*, with the input dimension δ^0 of Layer 1 being at least $|\Pi|$. Typical aggregation functions are sum, max, and mean, applied component-wise. We only consider GNNs in which every layer uses the same aggregation function. If we want to highlight the aggregation function, we may speak of a Sum-GNN, a Max-GNN, or a Mean-GNN.

We now make precise the semantics of GNNs. For $1 \leq \ell \leq L$, the ℓ-th layer assigns to each Π-labeled graph G and vertex $v \in V^G$ a feature vector $\bar{x}^\ell_{G,v} \in \mathbb{R}^{\delta^\ell}$. The *initial feature vector* $\bar{x}^0_{G,v} \in \mathbb{R}^{\delta^0}$ of vertex v in graph G is defined as follows: for all $i \in [|\Pi|]$, the i-th value of $\bar{x}^0_{G,v}$ is 1 if $P_i \in \pi^G(v)$, and all other values are 0. The feature vector $\bar{x}^\ell_{G,v}$ assigned by the ℓ-th layer is

$$\bar{x}^\ell_{G,v} = \mathsf{COM}^\ell(\bar{x}^{\ell-1}_{G,v}, \mathsf{AGG}^\ell(\{\{\bar{x}^{\ell-1}_{G,u} \mid u \in \mathcal{N}(v)\}\})) \quad (*)$$

where $\{\{\cdots\}\}$ denotes multiset; we define the mean of the empty multiset to be 0. We write $\bar{x}^\ell_v$ instead of $\bar{x}^\ell_{G,v}$ if the graph G is clear from the context. The output of the last layer is then passed to the classification function. The property *defined* by a GNN $\mathcal{G}$ is the set of pointed Π-labeled graphs

$$\{(G,v) \mid \mathsf{CLS}(\bar{x}^L_{G,v}) = 1\}.$$

We also say that $\mathcal{G}$ *accepts* the graphs in this set.

In a *simple* GNN, as considered for example in (Barceló et al. 2020; Ahvonen et al. 2024; Tena Cucala et al. 2023), the combination function is restricted to the form

$$\mathsf{COM}(\bar{x}_v, \bar{x}_a) = f(\bar{x}_v \cdot C + \bar{x}_a \cdot A + \bar{b}),$$

where $f : \mathbb{R} \to \mathbb{R}$ is a (typically non-linear) *activation function* applied component-wise, $A, C \in \mathbb{R}^{\delta^{\ell-1} \times \delta^\ell}$ are matrices, $\bar{b} \in \mathbb{R}^{\delta^\ell}$ is a bias vector, and $\bar{x}_a$ is the result of the aggregation function, applied as in $(*)$. Relevant choices for the activation function f include the *truncated ReLU* $\mathsf{ReLU}^*(x) = \min(\max(0,x),1)$, the *ReLU* $\mathsf{ReLU}(x) = \max(0,x)$ and the *sigmoid function* $\sigma(x) = \frac{1}{1+e^{-x}}$. The following is an easy consequence of the observation that $\mathsf{ReLU}^*(x) = \mathsf{ReLU}(x) - \mathsf{ReLU}(x-1)$ for all $x \in \mathbb{R}$.

Lemma 1. *Let* $\mathsf{AGG} \in \{\mathsf{Max}, \mathsf{Sum}, \mathsf{Mean}\}$. *If a property is definable by a* simple AGG-*GNN with* ReLU^* *activation, then it is definable by a* simple AGG-*GNN with* ReLU *activation.*

Modal Logic Formulas of *modal logic (ML)* over a set of vertex labels Π are defined by the grammar rule

$$\varphi ::= P \mid \neg\varphi \mid \varphi \vee \varphi \mid \Diamond\varphi,$$

where P ranges over Π (Blackburn, de Rijke, and Venema 2001). As usual, we use $\varphi \wedge \psi$ as abbreviation for $\neg(\neg\varphi \vee \neg\psi)$, $\Box\varphi$ as abbreviation for $\neg\Diamond\neg\varphi$, $\top$ as abbreviation for $P \vee \neg P$ with $P \in \Pi$ chosen arbitrarily, and $\bot$ as abbreviation for $\neg\top$. Satisfaction of a formula φ by a vertex v in a Π-labeled graph G is defined inductively as follows:

$$\begin{array}{lcll} G,v & \models & P & \text{if } P \in \pi(v) \\ G,v & \models & \neg\varphi & \text{if not } G,v \models \varphi \\ G,v & \models & \varphi \vee \psi & \text{if } G,v \models \varphi \text{ or } G,v \models \psi \\ G,v & \models & \Diamond\varphi & \text{if } G,u \models \varphi \text{ for some } u \in \mathcal{N}(G,v). \end{array}$$

Graded modal logic (GML) is a well-known extension of ML in which the diamond is replaced with a counting version $\Diamond^{\geq n}$, $n \in \mathbb{N}$, where

$$G,v \models \Diamond^{\geq n}\varphi \text{ if } |\{u \in \mathcal{N}(G,v) \mid G,u \models \varphi\}| \geq n.$$

As a shortcut, we may use $\Diamond^{=k}\psi$ to mean $\Diamond^{\geq k}\psi \wedge \neg\Diamond^{\geq k+1}\psi$. For more details on GML, see for instance (Goble 1970; De Rijke 2000).

We next introduce *ratio modal logic (RML)* in which the standard ML diamond is also replaced with a counting version, but here the counting is relative rather than absolute. Diamonds take the form $\Diamond^{\geq r}$ and $\Diamond^{>r}$ with $r \in [0,1]$. We have $G,v \models \Diamond^{\geq r}\varphi$ (resp. $G,v \models \Diamond^{>r}$) if the fraction of successors of v that satisfy φ is at least r (resp. exceeds r). If a vertex v has no successors, then we define $G,v \models \Diamond^{\geq t}\varphi$ and $G,v \not\models \Diamond^{>t}\varphi$ for all φ. As a useful shortcut, we may write $\Diamond^{=f_i}$ to mean $\Diamond^{\geq f_i}\varphi \wedge \neg \Diamond^{>f_i} \varphi$. Modal operators of this kind have occasionally been considered in the literature, see for instance (Pacuit and Salame 2004). RML is a fragment of modal logic with Presburger constraints (Demri and Lugiez 2010). We remark that while ML and GML are fragments of first-order logic (FO), the diamond operators of RML cannot even be expressed in MSO.

The *modal depth* of a modal formula φ, no matter whether GML, RML, or ML, is the nesting depth of diamonds in φ.

Notions of Expressive Power. Both GNNs and modal logic formulas may be viewed as vertex classifiers. We say that a property P is *(uniformly) expressible* by a class of vertex classifiers $\mathcal{C}$, such as $\mathcal{C} =$ GML, if there is a $C \in \mathcal{C}$ such that the pointed graphs accepted by C are exactly those in P. For classes of classifiers $\mathcal{C}_1, \mathcal{C}_2$, we write $\mathcal{C}_1 \subseteq \mathcal{C}_2$ to mean that every property expressible by a classifier from $\mathcal{C}_1$ is also expressible by a classifier from $\mathcal{C}_2$. We then also say that $\mathcal{C}_2$ is *at least as expressive* as $\mathcal{C}_1$. This notion of expressive power is commonly referred to as the *uniform setting*.

This is in contrast to the *non-uniform setting* where a property P is *(non-uniformly) expressible* by a class of classifiers $\mathcal{C}$ if for every graph size $n \geq 0$, there is a $C \in \mathcal{C}$ such that the pointed graphs of size n accepted by C are exactly those in P. Also in this setting, we may write $\mathcal{C}_1 \subseteq \mathcal{C}_2$ to mean that every property (non-uniformly) expressible by a classifier from $\mathcal{C}_1$ is also (non-uniformly) expressible by a classifier from $\mathcal{C}_2$. We shall always make clear whether we refer to the uniform or non-uniform setting.

In both the uniform and the non-uniform setting, we may write $\mathcal{C}_1 = \mathcal{C}_2$ as an abbreviation for $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \mathcal{C}_1$. The following is clear from the definitions.

Lemma 2. *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two classes of classifiers. If $\mathcal{C}_1 \subseteq \mathcal{C}_2$ in the uniform setting, then $\mathcal{C}_1 \subseteq \mathcal{C}_2$ in the non-uniform setting.*

We now clarify the relative expressive power of the modal logics introduced above, both in the uniform and in the non-uniform setting.

Lemma 3. *In the uniform setting,*

1. ML $\subsetneq$ RML *and* ML $\subsetneq$ GML*;*
2. RML $\not\subseteq$ GML *and* GML $\not\subseteq$ RML.

In the non-uniform setting, ML $\subsetneq$ RML $\subsetneq$ GML.

3 Non-Uniform Setting

We give logical characterizations of GNNs in the non-uniform setting. The characterizations are absolute, that is, they are not relative to FO, MSO, or any other background logic. We start with a summary of all results in this section.

Theorem 1. *In the non-uniform setting,*

1. Mean-GNN $\subseteq$ RML $\subseteq$ simple Mean-GNN
2. Sum-GNN $\subseteq$ GML $\subseteq$ simple Sum-GNN
3. Max-GNN $\subseteq$ ML $\subseteq$ simple Max-GNN.

This holds for truncated ReLU and ReLU, and for Points 1 and 2 for every continuous non-polynomial activation function.

Point 3 of Theorem 1 even holds in the uniform setting, also there without relativization to a background logic. The same is true for the second inclusion in Point 2 as per (Barceló et al. 2020), but as we shall see, not for any of the other inclusions. Together with Lemma 3, Theorem 1 also reproves the following result from (Xu et al. 2019).

Corollary 1. *In the non-uniform setting,*

$$\text{Max-GNN} \subsetneq \text{Mean-GNN} \subsetneq \text{Sum-GNN}.$$

To prove Theorem 1, we first give the translations from logic to GNNs, starting with Point 1. The general strategy is as in (Barceló et al. 2020), that is, we translate a ratio modal logic (RML) formula φ with L subformulas into a simple GNN with L layers, each of output dimension L. We first observe that we may assume w.l.o.g. that φ contains no RML diamonds of the form $\Diamond^{\geq t}$. This is because on graphs with at most n vertices such a diamond can be replaced by $\Diamond^{>t'}$ where t' is the largest rational number that is smaller than t and can occur as a fraction in a graph where every vertex has at most n successors.

For the GNN translation, we choose a suitable enumeration $\varphi_1, \dots, \varphi_L$ of the subformulas of φ and design the GNN to compute the truth value (that is, 0 or 1) of each subformula φ_i at every vertex v and store it in the i-th component of the feature vector for v. The only interesting case are subformulas of the form $\Diamond^{>t}\varphi_i$. If such a diamond is satisfied at a node v in a graph G with at most n vertices, then at least a fraction of

$$t' = \min\left\{\frac{\ell}{m} \mid 0 \leq \ell \leq m \leq n, \frac{\ell}{m} > t\right\}$$

successors of v must satisfy φ_i. Note that t' is the smallest rational number that is larger than t and can occur as a fraction in a graph where every vertex has at most n successors. If $\Diamond^{>t}\varphi_i$ is violated, then at most a fraction of t'' successors of v can satisfy φ_i where t'' is defined like t' except that $\min$ is now $\max$ and '$>$' is '$\leq$'. The gap between these fractions is at least $\frac{1}{n^2}$ and can be amplified using the matrix A from simple combination functions, which through the bias vector allows us to compute the desired truth value.

Theorem 2. RML $\subseteq$ simple Mean-GNN *in the non-uniform setting. This holds for truncated ReLU and ReLU activation.*

We next strengthen Theorem 2 to all continuous non-polynomial activation functions. This is based on universal approximation theorems from machine learning (Pinkus 1999). Intuitively, approximation suffices because the constant bound on the size of graphs imposed in the non-uniform setting ensures that a GNN can generate only a constant number of different feature vectors, across all input graphs, and we are good as long as we can distinguish these. A variation of the proof also works for sum aggregation, delivering the second inclusion in Point 2 of Theorem 1.

Theorem 3. *In the non-uniform setting and for all continuous non-polynomial activation functions:*

1. RML $\subseteq$ simple Mean-GNN*;*
2. GML $\subseteq$ simple Sum-GNN.

Point 2 was proved in (Barceló et al. 2020) in the uniform setting, but only for the special case of truncated ReLU activation.

We next treat the second inclusion in Point 3 of Theorem 1, using a minor variation of the proof in (Barceló et al. 2020). Like that proof, our proof even works in the uniform setting.

Theorem 4. ML $\subseteq$ simple Max-GNN *in the uniform setting. This holds for truncated ReLU and ReLU activation.*

We do not know whether this can be strengthened to all continuous non-polynomial activation functions.

We now turn to the translations from GNNs to logic. These also rely on the fact that in the non-uniform setting, a GNN can generate only a constant number of different feature vectors, across all input graphs. For every feature vector $\bar{x}$ and every layer ℓ of the GNN, we can construct a modal logic formula $\varphi^\ell_{\bar{x}}$ such that the GNN assigns $\bar{x}$ to a vertex v in layer ℓ if and only if $G, v \models \varphi^\ell_{\bar{x}}$. Depending on the aggregation function of the GNN, the formula $\varphi^\ell_{\bar{x}}$ needs to describe the distribution of feature vectors at the successors of v computed by level $\ell - 1$ in varying degrees of detail. For mean, we only need to know the fraction of successors at which each feature vector was computed, and thus the formula can be formulated in RML. For sum, we need exact multiplicities and thus require a GML formula.

Theorem 5. *In the non-uniform setting,*

1. Mean-GNN $\subseteq$ RML*;*
2. Sum-GNN $\subseteq$ GML.

As observed already in (Tena Cucala and Cuenca Grau 2024), with max aggregation the set of feature vectors ever generated by a GNN, across all input graphs, is finite even without bounding the graph size. Using similar arguments as for Theorem 5, we can thus show the following.

Theorem 6. *In the uniform setting,* Max-GNN $\subseteq$ ML.

We remark that all our results obtained in the non-uniform setting also hold in the uniform setting when a constant bound is imposed on the outdegree of vertices.

4 Uniform Setting

We again start with a summary of our results. Note that these are now relative to MSO, except for the case of Max-GNNs.

Theorem 7. *In the uniform setting,*

1. Mean-GNN $\cap$ MSO $\subseteq$ ML $\subseteq$ simple Mean-GNN
2. Sum-GNN $\cap$ MSO $\subseteq$ GML $\subseteq$ simple Sum-GNN

3. Max-GNN $\subseteq$ ML $\subseteq$ simple Max-GNN.

This holds for truncated ReLU and ReLU activation.

Also note that, compared to Theorem 1, the logic associated with Mean-GNNs has changed from RML to ML, because RML diamonds are not expressible in MSO. With Lemma 3, we obtain the following corollary.

Corollary 2. *In the uniform setting,*

$$\text{Max-GNN} = \text{Mean-GNN} \cap \text{MSO} \subsetneq \text{Sum-GNN} \cap \text{MSO}.$$

We now prove Theorem 7. We have already shown Point 3 as Theorems 4 and 6. The result in Point 2 is from (Barceló et al. 2020), stated there for truncated ReLU and for FO in place of MSO. We may invoke Lemma 1 for ReLU. Regarding the replacement of FO with MSO, it was observed in (Ahvonen et al. 2024) that the following is a consequence of results by (Elberfeld, Grohe, and Tantau 2016).

Lemma 4. *Any property expressible in MSO and by a GNN is also FO-expressible. This only depends on invariance under unraveling and thus holds for all choices of aggregation, activation, and classification function.*

In the remainder of this section, we prove Point 1 of Theorem 7. For the first inclusion, we need suitable versions of Ehrenfeucht-Fraïssé (EF) games. Such games are played by two players, *Spoiler (S)* and *Duplicator (D)*, who play on two potentially infinite pointed graphs $(G_1, v_1), (G_2, v_2)$. Spoiler's aim is to show that the graphs are dissimilar while D wishes to show that they are similar. The game is played in rounds. In the *GML game*, which in addition is parameterized by a *number of rounds* $\ell \in \mathbb{N}$ and a *counting bound* $c \in \mathbb{N}^+$, each round consists of the following steps (Otto 2019):

1. S chooses $i \in \{1, 2\}$ and a set $U_i \subseteq \mathcal{N}(G_i, v_i)$ with $0 < |U_i| \leq c$;
2. D selects a set $U_{3-i} \subseteq \mathcal{N}(G_{3-i}, v_{3-i})$ with $|U_1| = |U_2|$;
3. S selects a vertex $u_{3-i} \in U_{3-i}$;
4. D selects a vertex $u_i \in U_i$.

The game proceeds on the graphs (G_1, u_1) and (G_2, u_2). Spoiler wins as soon as one of the following conditions hold, possibly at the very beginning of the game:

- $\pi^{G_1}(v_1) \neq \pi^{G_2}(v_2)$;
- D fails in Step 2 because $|U_i| > |\mathcal{N}(G_{3-i}, v_{3-i})|$.

Duplicator wins if one of the following conditions hold:

- S cannot choose a non-empty set U_i in Step 1,
- after ℓ rounds, S has not won.

We write $\mathcal{E}_\ell^{\text{GML}[c]}(G_1, v_1, G_2, v_2)$ to denote the ℓ-round GML game with counting bound c on pointed graphs (G_1, v_1) and (G_2, v_2). We may vary GML games to obtain games for ML. An *ML game* is a GML game with counting bound 1. Thus Spoiler selects a singleton set U_i in Step 1 and consequently Steps 3 and 4 are trivialized. In other words, a round consists of first S choosing $i \in \{1, 2\}$ and a vertex $u_i \in \mathcal{N}(G_i, v_i)$, and D replying with a vertex $u_{3-i} \in \mathcal{N}(G_{3-i}, v_{3-i})$. We denote these games with $\mathcal{E}_\ell^{\text{ML}}(G_1, v_1, G_2, v_2)$.

We use GML$[c]$ to denote the fragment of GML in which in all diamonds $\Diamond^{\geq n}$ we have $n \leq c$.

Theorem 8. *Let $\mathcal{L} \in \{\text{ML}\} \cup \{\text{GML}[c] \mid c \geq 0\}$, and let P be a vertex property. The following are equivalent for all $\ell \geq 0$:*

1. *there exists an $\mathcal{L}$ formula φ of modal depth at most ℓ such that for all pointed graphs (G, v): $G, v \models \varphi$ if and only if $(G, v) \in P$.*
2. *Spoiler has a winning strategy in $\mathcal{E}_\ell^{\mathcal{L}}(G_1, v_1, G_2, v_2)$ for all pointed graphs $(G_1, v_1), (G_2, v_2)$ with $(G_1, v_1) \in P$ and $(G_2, v_2) \notin P$.*

Proofs can be found in the literature, see for instance (Otto 2019) and Chapter 3.2 in (Goranko and Otto 2007). We next recall that the following was proved in (Barceló et al. 2020), not specifically for Mean-GNNs, but in fact independently of the aggregation function used.

Theorem 9. Mean-GNN $\cap$ MSO $\subseteq$ GML *in the uniform setting.*

We improve this from GML to ML, exploiting the fact that properties definable by Mean-GNNs are invariant under scaling the graph, that is, choosing a $c \geq 1$ and multiplying each vertex in the graph exactly c times. To make this formal, let $G = (V, E, \pi)$ be a Π-labeled graph and $c \geq 1$. The *c-scaling of G* is the Π-labeled graph $c \cdot G := (V', E', \pi')$ where $V' = \{(v, i) \mid v \in V,\ 1 \leq i \leq c\}$, $E' = \{((v, i), (u, j)) \mid (v, u) \in E\}$, and $\pi'((v, i)) = \pi(v)$ for all $v \in V$ and $i \in [c]$. The following is immediate from the fact that the mean of a multiset is invariant under multiplying all multiplicities by a constant $c \geq 1$.

Lemma 5. *Let $\mathcal{G}$ be a* Mean-GNN *on Π-labeled graphs, (G, v) a Π-labeled pointed graph, and $c \geq 1$. Then for all $i \in [c]$: $\bar{x}_{G,v}^L = \bar{x}_{c \cdot G, (v,i)}^L$.*

The following relates EF-games for ML to EF-games for GML$[c]$ on the corresponding c-scaled graphs.

Lemma 6. *Let (G_1, v_1), (G_2, v_2) be pointed graphs and $\ell \geq 0$. If D has a winning strategy in $\mathcal{E}_\ell^{\text{ML}}(G_1, v_1, G_2, v_2)$, then D also has a winning strategy in*

$$\mathcal{E}_\ell^{\text{GML}[c]}(c \cdot G_1, (v_1, k_1), c \cdot G_2, (v_2, k_2)),$$

for all $c \geq 1$ and $k_1, k_2 \in [c]$.

With Lemma 6 as the main ingredient, we can now show the following.

Corollary 3. Mean-GNN $\cap$ MSO $\subseteq$ ML *in the uniform setting.*

Proof. Let P be a vertex property that is expressible by a Mean-GNN and by an MSO formula. By Theorem 9, P is definable by a GML formula φ. Let c be maximal such that φ contains a diamond $\Diamond^{\geq c}$.

Assume to the contrary of what we have to show that P cannot be expressed in modal logic (ML). Then by Theorem 8 for each $\ell \geq 0$ there exist pointed graphs $(G_1, v_1) \in P$ and $(G_2, v_2) \notin P$ such that Duplicator wins $\mathcal{E}_\ell^{\text{ML}}(G_1, v_1, G_2, v_2)$. By Lemma 6, Duplicator also wins $\mathcal{E}_\ell^{\text{GML}[c]}(c \cdot G_1, (v_1, 1), c \cdot G_2, (v_2, 1))$. Since P is definable by a Mean-GNN, Lemma 5 yields $(c \cdot G_1, (v_1, 1)) \in P$ and $(c \cdot G_2, (v_2, 1)) \notin P$. Therefore, again by Theorem 8, P cannot be defined by a GML$[c]$ formula; a contradiction. ❑

Regarding the second inclusion of Point 1 of Theorem 7, we actually prove something stronger.

Theorem 10. RML $\subseteq$ simple Mean-GNN *in the uniform setting.*

The proof is similar to that of Theorem 2. In particular, truth and falsity of formulas is represented by the values 1 and 0 in feature vectors. Importantly, we use a step function as the activation function. The reason why we cannot use a continuous activation function such as ReLU in our translation is the modal diamond $\Diamond\varphi$: if a vertex has a successor that satisfies φ, then the mean over all successors may still be arbitrarily close to 0, inducing a discontinuity.

5 Uniform Setting, Reloaded

From a practical perspective, the use of a non-continuous activation function (resulting in a non-continuous combination function) in the proof of Theorem 10 is unsatisfactory. The combination function of a GNN is often represented as a feed-forward neural network (FNN) with a continuous activation function such as truncated ReLU, ReLU, or sigmoid, and is then guaranteed to be continuous. Importantly, the use of a non-continuous and thus non-differentiable combination function precludes a direct use of backpropagation, gradient descent, and related methods. It is thus natural to ask whether the translation of ML to Mean-GNNs can also be realized using a continuous combination function. The answer turns out to be positive. We use Mean^c-GNN to denote the class of GNNs in which the combination functions COM^ℓ, viewed as functions $\mathsf{COM}^\ell : \mathbb{R}^{2\delta^{\ell-1}} \to \mathbb{R}^{\delta^\ell}$, are continuous.

Theorem 11. ML $\subseteq$ Mean^c-GNN *in the uniform setting.*

From a practical perspective, however, the construction in the proof of Theorem 11 is even more unsatisfactory. It uses a combination function that is continuous, but very artificial: the function is inspired by and derived from the proof of Cantor's isomorphism theorem. Moreover, there is a price to pay in terms of a rather unnatural classification function. We represent truth and falsity of logical formulas as irrational and rational numbers, and consequently the classification function has to return 1 for all irrational numbers and 0 for all rational ones. We conjecture that our proof can be improved to yield simple Mean^c-GNNs, at the expense of making it more technical.

In practice, classification functions are often threshold functions.[1] It is therefore relevant to ask about the expressive power of Mean-GNNs that use a continuous combination function and a threshold classification function. We denote this class with Mean^{c,t}-GNN. To be more precise, the classification function is required to be of the form

$$\mathsf{CLS}(\bar{x}) = \begin{cases} 1 & \text{if } x_i \sim c, \\ 0 & \text{otherwise} \end{cases}$$

[1]Or they are represented by a feed-forward neural network (FNN) to which a threshold is applied. Our model captures this case because we can include the FNN in the COM function of the final GNN layer.

where $i \in [\delta^L]$, $\sim \in \{\geq, >\}$, and $c \in \mathbb{R}$. Note that adding the options $\sim \in \{\leq, <\}$ is syntactic sugar because we can replace the last combination function COM^L with $(-1)\cdot\mathsf{COM}^L$ and compare x_i against $-c$ in the classification function. We remark that all translations from logic to GNN given in this paper use threshold classification, except Theorem 11.

We characterize the expressive power of Mean^{c,t}-GNNs, relative to MSO, in terms of an alternation-free fragment of ML. Formally, *alternation-free modal logic (AFML)* is defined by the grammar rule

$$\begin{aligned} \varphi &::= \psi \mid \vartheta \\ \psi &::= P \mid \neg P \mid \Box\bot \mid \psi \wedge \psi \mid \psi \vee \psi \mid \Diamond\psi \\ \vartheta &::= P \mid \neg P \mid \Diamond\top \mid \vartheta \wedge \vartheta \mid \vartheta \vee \vartheta \mid \Box\vartheta. \end{aligned}$$

We use AFML[1] to denote all AFML formulas formed according to the grammar rule for ψ in the definition of AFML, and likewise for AFML[2] and the grammar rule for ϑ. Our main result is as follows.

Theorem 12. *In the uniform setting,*

$$\text{Mean}^{c,t}\text{-GNN} \cap \text{MSO} \subseteq \text{AFML} \subseteq \text{simple Mean}^{c,t}\text{-GNN}.$$

This holds for truncated ReLU, ReLU, and sigmoid activation.

The proof of the first inclusion in Theorem 12 relies on EF games for AFML. For $k \in \{1, 2\}$, an AFML$[k]$ *game* is an ML game subject to the modification that Spoiler chooses the same value $i = k$ in the first step of each round, except that S may choose $i = 3 - k$ in case that v_k has no successors. Note that in the latter case, Spoiler immediately wins because Duplicator cannot respond with a successor of v_k (we in fact have $G_k, v_k \models \Box\bot$ and $G_{3-k}, v_{3-k} \not\models \Box\bot$). We denote these games with $\mathcal{E}_\ell^{\text{AFML}[k]}(G_1, v_1, G_2, v_2)$. A version of Theorem 8 for AFML[1] and AFML[2] is proved in the appendix.

Using AFML games, it is easy to prove that basic ML properties such as $\varphi = \Diamond P \wedge \Box Q$ are not expressible in AFML, that is, AFML $\subsetneq$ ML (both in the uniform and non-uniform setting). Details are in the appendix.

To prove Theorem 12, we start from Corollary 3. We need some preliminaries. Let $G = (V, E, \pi)$ be a graph and $v \in V$. A *path* in G is a sequence $p = v_0, \ldots, v_n$ of vertices from V such that $(v_i, v_{i+1}) \in E$ for all $i < n$. The path *starts at* v_0 and is of *length* n, and we use $\mathsf{tail}(p)$ to denote v_n. The *unraveling of* G *at* v is the potentially infinite tree-shaped graph $\mathsf{Unr}(G, v) = (V', E', \pi')$ defined as follows:

- V' is the set of all paths in G that start at v;
- E' contains an edge (p, pu) if $(\mathsf{tail}(p), u) \in E$;
- $\pi'(p) = \pi(\mathsf{tail}(p))$.

For $L \geq 0$, the *unraveling of* G *at* v *up to depth* L, denoted $\mathsf{Unr}^L(G, v)$, is the (finite) subgraph of $\mathsf{Unr}(G, v)$ induced by all paths of length at most L.

It is well-known that modal formulas are invariant under unraveling up to their modal depth. A similar statement holds for GNNs.

Lemma 7 (Barceló et al. (2020))**.** *Let G be a graph, $v \in V^G$, $\mathcal{G} = (L, \{\mathsf{AGG}^\ell\}_{\ell\in[L]}, \{\mathsf{COM}^\ell\}_{\ell\in[L]}, \mathsf{CLS})$, and $1 \leq \ell \leq L$. Then $\bar{x}^\ell_{G,v} = \bar{x}^\ell_{\mathsf{Unr}^L(G,v),v}$.*

We next show that slightly changing a highly scaled input graph to a Mean^{c,t}-GNN does not change the computed value in an unbounded way. We first formalize what we mean by 'slight change'.

Definition 1. *Let $G = (V, E, \pi)$ be a graph and $n \geq 0$. A graph $G' = (V', E', \pi')$ is an n-*extension *of G if it satisfies the following conditions for all $v \in V$:*

1. *$V \subseteq V'$, $E \subseteq E'$,*
2. *for all $v \in V$: $\pi(v) = \pi'(v)$*
3. *if $\mathcal{N}(G, v) \neq \emptyset$, then $|\mathcal{N}(G', v) \setminus \mathcal{N}(G, v)| \leq n$, and*
4. *if $\mathcal{N}(G, v) = \emptyset$, then $\mathcal{N}(G', v) = \emptyset$.*

Note that n-extensions can add up to n fresh successors to any vertex that already has at least one successor. Intuitively, the value of n will be small compared to the scaling of the graph G of which the n-extension is taken.

In what follows, we use the maximum metric and define the distance of two vectors $\bar{x}$ and $\bar{y}$ of dimension δ to be $||\bar{x} - \bar{y}||_\infty = \max_{1 \leq i \leq \delta} |x_i - y_i|$. The following makes precise why we are interested in n-extensions.

Lemma 8. *Let $\mathcal{G}$ be a* Mean^c-GNN *with L layers. Then for all $\varepsilon > 0$, $n \geq 1$, and $\ell \in [L]$, there exists a constant c such that for all $c' \geq c$, graphs $H = c' \cdot G$, vertices (v, i) in H, and n-extensions H' of H: $||\bar{x}^\ell_{H,(v,i)} - \bar{x}^\ell_{H',(v,i)}||_\infty < \varepsilon$.*

One ingredient to the proof of Lemma 8 is the observation that for every Mean^c-GNN, there are constant upper and lower bounds on the values that may occur in feature vectors, across all input graphs. The reader might want to compare this with Max-GNNs where even the number of such values is bounded by a constant, see the discussion before Theorem 6.

Theorem 13. Mean^{c,t}-GNN $\cap$ MSO $\subseteq$ AFML *in the uniform setting.*

Proof. (sketch) Assume to the contrary that there exists a Mean^{c,t}-GNN $\mathcal{G}$ with L layers that is equivalent to an MSO formula, but not to an AFML formula. By Corollary 3, $\mathcal{G}$ is equivalent to an ML formula φ. Assume first that the classification function of $\mathcal{G}$ uses '$>$' rather than '$\geq$'.

Because φ is not expressible in AFML, for each $\ell \in \mathbb{N}$ there exist pointed graphs (G, v) and (G', v') with $G, v \models \varphi$ and $G', v' \not\models \varphi$ such that D has a winning strategy in $\mathcal{E}^{\text{AFML}[1]}_\ell(G, v, G', v')$. Our aim is to transform (G, v) into a pointed graph (H, u) such that

(i) $H, u \models \varphi$ and

(ii) D has a winning strategy in $\mathcal{E}^{\text{ML}}_\ell(H, u, G', v')$.

Then, by Theorem 8, φ is not expressible in ML. A contradiction.

It can be shown that since D has a winning strategy in $\mathcal{E}^{\text{AFML}[1]}_\ell(G, v, G', v')$, they also have one in $\mathcal{E}^{\text{AFML}[1]}_\ell(\mathsf{Unr}^K(G, v), v, G', v')$ for all $K \geq \ell$. Moreover, in AFML[1]-games in which the first graph is tree-shaped, the existence of a winning strategy for D implies the existence of a memoryless winning strategy. We may view such a strategy as a function $\mathsf{ws} : V^{\mathsf{Unr}^K(G,v)} \to V^{G'}$ such that if S plays vertex u in $\mathsf{Unr}^K(G, v)$, then D always answers with $\mathsf{ws}(u)$. We also set $\mathsf{ws}(v) = v'$.

Let $m = |V^{G'}|$ and $K = \max(\ell, L)$. Since classification is based on '$>$', we find an $\varepsilon > 0$ such that all pointed graphs (H, u) with $||\bar{x}^L_{G,v} - \bar{x}^L_{H,u}||_\infty < \varepsilon$ are accepted by $\mathcal{G}$. By Lemma 8, there exists a c such that $||\bar{x}^L_{G'',v} - \bar{x}^L_{X,v}||_\infty < \varepsilon$ in each m-extension X of $G'' = c \cdot \mathsf{Unr}^K(G, v)$. Now, (H, u) is defined as follows:

1. start with $G'' = c \cdot \mathsf{Unr}^K(G, v)$;
2. take the disjoint union with all $\mathsf{Unr}^K(G', v')$, $v' \in V^{G'}$;
3. for each vertex $(u, i) \in V^{G''}$ that has at least one successor, let $\mathcal{N}(G', \mathsf{ws}(u)) = \{u'_1, \ldots, u'_m\}$. Add to (u, i) the fresh successors $u'_1, \ldots, u'_m$;
4. $u = (v, 1)$.

We show in the appendix that Conditions (i) and (ii) are satisfied. We also deal there with the case where the classification function is based on '$\geq$', using AFML[2]. ❑

Next, we show that each formula in AFML can be realized by a simple Mean^{c,t}-GNN. As in the proof of Theorem 10, we face the challenge that the mean over all successors may diminish values. Here, we address this by representing the truth of subformulas by values from the range $(0, 1]$ and falsity by value 0. With this encoding, we can realize the modal diamonds of AFML[1] using truncated ReLU activation, but we cannot realize modal boxes. Still, it is easy to also treat AFML[2] using duality arguments.

Theorem 14. AFML $\subseteq$ simple Mean^{c,t}-GNN *in the uniform setting. This holds for truncated ReLU and ReLU as activation functions.*

The proof of Theorem 14 can be adapted to sigmoid activation. This once more requires non-trivial modifications. Truth is now encoded by values from the range $(\frac{1}{2}, 1)$ and falsity by value $\frac{1}{2}$.

Theorem 15. AFML $\subseteq$ simple Mean^{c,t}-GNN *in the uniform setting, with sigmoid as the activation function.*

6 Conclusion

We have identified logical characterizations of graph neural networks with mean aggregation, in several different settings. Some interesting questions remain open. For instance, we would like to know whether Theorem 4 can be strengthened to all continuous non-polynomial activation functions, in the spirit of Theorem 3. It would also be interesting to consider broader classes of activation functions in the uniform setting. Finally, it would be interesting to find absolute logical characterization of Mean-GNNs and Mean^{c,t}-GNNs, that is, characterizations that are not relative to any background logic such as FO or MSO. In fact, we can already provide two first observations on this subject.

Theorem 16.

1. *The property 'there exist more successors that satisfy P_1 than successors that satisfy P_2' is not expressible in RML, but by a simple* Mean^{c,t}-GNN.
2. *The RML formula $\Diamond^{>\frac{1}{2}} \Diamond^{>\frac{1}{2}} P$ is not expressible by a* Mean^{c,t}-GNN.

Acknowledgements

Carsten Lutz was supported by DFG project LU 1417/4-1.

References

Ahvonen, V.; Funk, M.; Heimann, D.; Kuusisto, A.; and Lutz, C. 2026. Expressive Power of Graph Transformers via Logic. In *Proc. of AAAI*. AAAI Press.

Ahvonen, V.; Heiman, D.; Kuusisto, A.; and Lutz, C. 2024. Logical characterizations of recurrent graph neural networks with reals and floats. In *Proc. of NeurIPS*, volume 37, 104205–104249. Curran Associates, Inc.

Barceló, P.; Kostylev, E. V.; Monet, M.; Pérez, J.; Reutter, J.; and Silva, J.-P. 2020. The Logical Expressiveness of Graph Neural Networks. In *Proc. of ICLR*.

Barceló, P.; Kostylev, E. V.; Monet, M.; Pérez, J.; Reutter, J. L.; and Silva, J. P. 2020. The Expressive Power of Graph Neural Networks as a Query Language. *SIGMOD Rec.*, 49(2): 6–17.

Benedikt, M.; Lu, C.; Motik, B.; and Tan, T. 2024. Decidability of Graph Neural Networks via Logical Characterizations. In *Proc. of ICALP*, volume 297 of *LIPIcs*, 127:1–127:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Blackburn, P.; de Rijke, M.; and Venema, Y. 2001. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press. ISBN 978-1-10705088-4.

Bongini, P.; Bianchini, M.; and Scarselli, F. 2021. Molecular generative Graph Neural Networks for Drug Discovery. *Neurocomputing*, 450: 242–252.

De Rijke, M. 2000. A note on graded modal logic. *Studia Logica*, 64(2): 271–283.

Demri, S.; and Lugiez, D. 2010. Complexity of modal logics with Presburger constraints. *J. of Applied Logic*, 8(3): 233–252.

Deng, A.; and Hooi, B. 2021. Graph Neural Network-Based Anomaly Detection in Multivariate Time Series. *Proc. of the AAAI*, 35(5): 4027–4035.

Elberfeld, M.; Grohe, M.; and Tantau, T. 2016. Where First-Order and Monadic Second-Order Logic Coincide. *ACM Transactions on Computational Logic*, 17(4).

Goble, L. F. 1970. Grades of modality. *Logique et Analyse*, 13(51): 323–334.

Goranko, V.; and Otto, M. 2007. 5 Model theory of modal logic. In Blackburn, P.; Van Benthem, J.; and Wolter, F., eds., *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, 249–329. Elsevier.

Grohe, M. 2021. The logic of graph neural networks. In *Proc. of LICS*, 5:1–5:17. ACM.

Grohe, M. 2024. The Descriptive Complexity of Graph Neural Networks. *TheoretiCS*, 3.

Grohe, M.; and Rosenbluth, E. 2024. Are Targeted Messages More Effective? In *Proc. of LICS*, 40:1–40:14. ACM.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jiang, W.; and Luo, J. 2022. Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications*, 207: 117921.

Khalife, S.; and Tonelli-Cueto, J. 2025. Is uniform expressivity too restrictive? Towards efficient expressivity of GNNs. In *Proc. of ICLR*. OpenReview.net.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proc. of ICLR*.

Morris, M.; and Horrocks, I. 2025. Sound Logical Explanations for Mean Aggregation Graph Neural Networks. In *Proc. of NeurIPS*.

Otto, M. 2019. Graded modal logic and counting bisimulation. *arXiv preprint arXiv:1910.00039*.

Pacuit, E.; and Salame, S. 2004. Majority Logic. In *Proc. of KR*, 598–605. AAAI Press.

Pinkus, A. 1999. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8: 143–195.

Rosenbluth, E.; Toenshoff, J.; and Grohe, M. 2023. Some might say all you need is sum. In *Proc. of IJCAI*.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1): 61–80.

Schönherr, M.; and Lutz, C. 2025. Logical Characterizations of GNNs with Mean Aggregation. arXiv:2507.18145.

Shlomi, J.; Battaglia, P.; and Vlimant, J.-R. 2020. Graph neural networks in particle physics. *Machine Learning: Science and Technology*, 2(2): 021001.

Tena Cucala, D.; Cuenca Grau, B.; Motik, B.; and Kostylev, E. V. 2023. On the Correspondence Between Monotonic Max-Sum GNNs and Datalog. In *Proc. of KR*, 658–667.

Tena Cucala, D. J.; and Cuenca Grau, B. 2024. Bridging Max Graph Neural Networks and Datalog with Negation. In *Proc. of KR*, 950–961.

Wu, S.; Sun, F.; Zhang, W.; Xie, X.; and Cui, B. 2022. Graph Neural Networks in Recommender Systems: A Survey. *ACM Computing Surveys*, 55(5).

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1): 4–24.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *Proc. of ICLR*. OpenReview.net.

Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W. L.; and Leskovec, J. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proc. of SIGKDD*, KDD '18, 974–983. ACM.

Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2020. Graph neural networks: A review of methods and applications. *AI Open*, 1: 57–81.