Expressive Power of Deep Homomorphism Networks over Relational Databases

Moritz Schönherr
Leipzig University & ScaDS.AI Dresden/Leipzig
schoenherr@informatik.uni-leipzig.de

Balder ten Cate
University of Amsterdam
b.d.tencate@uva.nl

Maurice Funk
Leipzig University & ScaDS.AI Dresden/Leipzig
maurice.funk@uni-leipzig.de

Benny Kimelfeld
Technion & RelationalAI
bennyk@cs.technion.ac.il

Carsten Lutz
Leipzig University & ScaDS.AI Dresden/Leipzig
clu@informatik.uni-leipzig.de

Arie Soeteman
University of Amsterdam
a.w.soeteman@uva.nl

Abstract

The expressive limitations of message-passing Graph Neural Networks (GNNs) have motivated a wide range of more powerful graph learning architectures. We advocate Deep Homomorphism Networks (DHNs) as a model particularly well-suited for learning over relational databases, due to their close connection to important fragments of SQL such as conjunctive queries. We study the precise expressive power of DHNs by relating them to various natural fragments and extensions of first-order logic (FO). For DHNs with max, sum, and mean aggregation, we establish connections to the unary negation fragment (UNFO) and to the extensions of UNFO with counting quantifiers and with ratio quantifiers. We further relate sum-aggregation DHNs to the unary quantifier alternation fragment of FO and to a fragment of FO extended with expressive counting. Through the classical correspondence between FO and SQL, these results also illuminate the relation between DHNs and SQL. They further enable us to study the decidability of fundamental static analysis problems for DHNs, the emptiness problem and the containment problem. We confirm through experiments that the established differences in expressive power are reflected in the performance on suitable prediction tasks.

1 Introduction

Message-passing Graph Neural Networks (GNNs) are a prominent class of graph learning models that have proven highly effective across a wide range of domains, including social network analysis [11], molecular property prediction [29], recommender systems [27], and data analytics for relational databases [8]. However, it is well known that the expressive power of the standard GNN model is bounded from above by the one-dimensional Weisfeiler–Leman test. This is a serious limitation: in essence, GNNs can only detect tree-shaped graph properties whereas even simple and common structures such as triangles, cliques, and fixed-length cycles lie beyond their expressive power. Consequently, a growing body of work has sought to design more expressive graph neural architectures, including higher-order GNNs [21], subgraph GNNs [23, 34, 35], and hierarchical ego GNNs [28]. These have exhibited improved performance on many tasks such as predicting constrained solubility of molecules in the ZINC benchmark, where substructure features such as triangles and rings are highly relevant [14].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

One important and practical way to increase the expressive power of GNNs, breaking free from their tree-shaped limitations, is to provide the ability to count homomorphisms from selected graph patterns. This is often done by feature augmentation: every vertex is adorned with homomorphism counts that capture local graph structure, such as the number of triangles it lies on [1, 3, 15, 22, 32]. However, homomorphism counts can also be integrated into the GNN architecture more deeply, replacing the standard edge-based message passing mechanism of GNNs by a more general form of message passing based on complex graph patterns. The latter, more powerful approach is taken by Deep Homomorphism Networks (DHNs), recently introduced in [20]. As shown in this article, DHNs offer strictly higher expressive power than standard GNNs even when the latter are augmented with homomorphism-count features. In addition, we argue that DHNs have another important advantage: they provide a natural connection to relational databases. Indeed, a vast amount of data is stored in relational systems and analyzing such data is an important application of GNNs, usually approached by first translating the database into a graph [6, 8, 30, 31]. For DHNs, such a translation is unnecessary as their definition extends gracefully from graphs to databases. Moreover, pattern homomorphisms are essentially an alternative presentation of conjunctive queries (CQs), arguably the most important query language for relational database systems. We recall that CQs correspond to the select-project-join fragment of SQL and that in practice a large fraction of SQL queries fall into this fragment [5]. Thus, DHNs provide a natural form of expressive power for database properties while also opening up the possibility to benefit from decades of work on the efficient evaluation of CQs.

Our contributions. The aim of this paper is to provide a careful analysis of the exact expressive power of DHNs used as vertex classifiers, with a focus on their relation to natural fragments and extensions of first-order logic (FO). Since, famously, a core of SQL corresponds to FO, this also helps clarify which SQL queries are expressible by DHNs and which are not. In addition, it enables us to obtain results about the static analysis of DHNs. We study the *emptiness problem*, which asks whether a given DHN defines the empty set of vertices, thereby indicating a potential flaw in the model, and the related *containment problem*, which asks whether the vertex classifier realized by one given DHN is subsumed by that realized by another. Our logical characterizations also allow us to prove that DHNs are strictly more expressive than GNNs augmented with initial homomorphism count features, as proposed by Barceló et al. [3].

We study DHNs with max, sum, and mean aggregation. For max-DHNs, we obtain a complete and uniform logical characterization: the database properties expressible by max-DHNs are exactly those definable in the unary negation fragment of FO (UNFO) [26]. By known results, this implies that the emptiness and containment problems for max-DHNs are decidable. Since directed graphs are a special case of databases, these results also apply to the former, and the same is true for all results discussed below.

For sum-DHNs, our results offer a broad yet incomplete picture. We prove that, for connected patterns, the expressive power on databases of bounded degree is exactly that of UNFOC, the extension of UNFO with counting quantifiers. This also yields a characterization in the non-uniform setting often studied in the GNN literature, where a different GNN may be used for each graph size [9]. We then prove that the emptiness and containment problems for sum-DHNs are undecidable, both on unrestricted-degree graphs and on bounded-degree graphs. On bounded-degree graphs, these problems become decidable when all patterns are connected. It is interesting to contrast this with the fact that, under mild assumptions the same problems are decidable for sum-GNNs [4].

There is, however, much more to say about sum-DHNs. While we do not obtain a precise logical characterization of their expressive power on databases of unrestricted degree, we are able to sandwich them between two logical formalisms that provide meaningful lower and upper bounds. For the lower bound, we define the new fragment UQAFO of FO that is based on unary quantifier alternation and admits inequalities, and hence counting. We show that UQAFO is strictly more expressive than UNFOC and, at the same time, no more expressive than sum-DHNs. This perspective also motivates a natural variation of DHNs, Deep Embedding Networks (DENs), in which homomorphisms are replaced by embeddings. Using Lovász' homomorphism count theorem, we prove that sum-DENs have the same expressive power as sum-DHNs, and we also show that they are strictly more expressive than UQAFO. Making a short digression to max aggregation, we further observe that max-DENs are strictly more expressive than max-DHNs, and in fact their expressive power is exactly that of UQAFO. To establish an upper bound on the expressive power of sum-DHNs, we propose a homomorphism-based logic inspired by first-order logic with counting, as studied in descriptive complexity [13].

Building on recent developments [10], we show that every sum-DHN can be translated into an equivalent formula in this logic.

We only briefly address the case of mean aggregation. Our main result is that, on bounded-degree databases and for connected patterns, mean-DHNs have the same expressive power as a variant of UNFOC in which counting quantifiers are replaced by ratio quantifiers, in the spirit of a recent study on the expressiveness of GNNs with mean aggregation [25].

We repeat that all results stated above hold both on relational databases and on graphs, subject to mild additional assumptions for the undecidability results, namely the presence of vertex colors.

We provide experiments that corroborate our theoretical findings on expressive power in practice. In particular, they confirm our two main separation results: sum-DHNs have higher expressive power than max-DHNs and than GNNs with initial homomorphism counts. Fully detailed proofs can be found in [24].

2 Preliminaries

Databases, Homomorphisms, Embeddings. For a vector $\bar{x} \in \mathbb{R}^k$ we use $x_1, \ldots, x_k$ to refer to its components. Fix a countably infinite set $\mathbf{V}$ of *values*. A *schema* $\mathbf{S}$ is a finite set of relation symbols R with associated arity $\mathsf{ar}(R) \geq 1$. An $\mathbf{S}$-*fact* is an expression of the form $R(\bar{v})$, where $R \in \mathbf{S}$ and $\bar{v}$ is an $\mathsf{ar}(R)$-tuple of values from $\mathbf{V}$. An $\mathbf{S}$-*database* is a finite set of $\mathbf{S}$-facts. We write $\mathsf{adom}(D)$ to denote the *active domain* of database D, that is, the set of values used in D. The *degree* of D is the degree of its Gaifman graph (defined in the appendix). That is, D is of degree n if any value $v \in \mathsf{adom}(D)$ co-occurs with at most n values in some fact.

Our DHNs run on databases, whereas in the original paper [20], DHNs are applied to undirected graphs. By choosing the *graph schema* $\mathbf{S} = \{E\}$ with $\mathsf{ar}(E) = 2$, we can capture the setting of directed graphs. Undirected graphs require the additional constraint that all inputs are irreflexive and symmetric. For example, the directed triangle corresponds to the database $D_3 = \{E(a,b), E(b,c), E(c,a)\}$, that is, $\mathsf{adom}(D_3)$ is the set of vertices of the graph and every edge of the graph becomes a database fact in D_3. One caveat is that databases cannot represent graphs that contain isolated vertices, that is, vertices that do not participate in any edge. All results in this article, however, also hold for graphs with isolated vertices.[1]

A *pointed* $\mathbf{S}$-*database* is a pair (D, v), where D is an $\mathbf{S}$-database and $v \in \mathsf{adom}(D)$. We typically write (D, v) as D^v or as $D^\bullet$ if the exact identity of v is not important. For $d \geq 0$, an $\mathbb{R}^d$-*embedded* $\mathbf{S}$-*database* is a pair (D, λ) where D is an $\mathbf{S}$-database and $\lambda \colon \mathsf{adom}(D) \to \mathbb{R}^d$ is an *embedding function*. Embedded databases also exist in a pointed version, defined as expected.

A homomorphism from a pointed database F^v to a pointed database D^u is a function $h \colon \mathsf{adom}(F) \to \mathsf{adom}(D)$ such that $h(v) = u$ and $R(\bar{v}) \in F$ implies $R(h(\bar{v})) \in D$, where $h(\bar{v})$ denotes the component-wise application of h. We use $\mathsf{Hom}(F^\bullet, D^\bullet)$ to denote the set of all homomorphisms from $F^\bullet$ to $D^\bullet$. A homomorphism from $F^\bullet$ to $D^\bullet$ is an *isomorphic embedding* if it is injective and, additionally, $R(h(\bar{v})) \in D$ implies $R(\bar{v}) \in F$. Isomorphic embeddings are also known as partial isomorphisms or induced substructure isomorphisms. The following theorem is widely known, the graph version can be found, e.g., in [17, Chapter 5.2.3]. A proof for databases is in the appendix.

Theorem 1. *[Lovász' homomorphism count theorem] Let* $\mathbf{S}$ *be a schema. For pointed* $\mathbf{S}$*-databases* $F^\bullet, D^\bullet$*, the number* k *of isomorphic embeddings from* $F^\bullet$ *to* $D^\bullet$ *is uniquely determined by the number of homomorphisms to* $D^\bullet$*, from all* $\mathbf{S}$*-databases with at most* $|\mathsf{adom}(F)|$ *values. Moreover,* k *can be written as a linear combination of the homomorphism counts. The same is true when the roles of homomorphisms and isomorphic embeddings are swapped.*

Deep Homomorphism Networks. A finite multiset over a set X is a function $M \colon X \to \mathbb{N}$ such that $M(x) > 0$ for only finitely many $x \in X$. An *aggregation function* is a function from finite multisets over $\mathbb{R}$ to $\mathbb{R}$, such as sum, arithmetic mean, and max. As a convention, we set $\mathsf{sum}(\emptyset) = \mathsf{mean}(\emptyset) = \mathsf{max}(\emptyset) = 0$. We shall often apply aggregation functions to multisets of vectors from $\mathbb{R}^d$. Such applications are always component-wise. For what follows, we reserve a

[1] To represent isolated vertices, one could for instance introduce a unary relation symbol to express "this is a vertex" and treat it in a special way in homomorphisms; cf. the adom relation symbol in Definition 2.

distinguished unary relation symbol adom that is treated in a special way by homomorphisms in that $\mathsf{adom}(v) \in F$ need not imply $\mathsf{adom}(h(v)) \in D$ in a homomorphism h from F to D.

Definition 2 (Homomorphism Query)**.** *Let* $\mathbf{S}$ *be a schema. A* homomorphism query *over* $\mathbf{S}$ *with input dimension* $d \geq 0$ *and output dimension* $d' \geq 0$ *is a triple* $(F^\bullet, \mu, \mathsf{agg})$ *where* $F^\bullet$ *is a pointed* $\mathbf{S} \uplus \{\mathsf{adom}\}$*-database,* μ *labels each value* v *in* F *with a function* μ_v *from* $\mathbb{R}^d$ *to* $\mathbb{R}^{d'}$ *referred to as a* transformation function*, and* agg *is an aggregation function. The* result *of evaluating* $(F^\bullet, \mu, \mathsf{agg})$ *on an* $\mathbb{R}^d$*-embedded pointed* $\mathbf{S}$*-database* $(D^\bullet, \lambda)$ *is the* $\mathbb{R}^{d'}$*-vector*

$$\mathsf{eval}((F^\bullet, \mu, \mathsf{agg}), (D^\bullet, \lambda)) := \mathsf{agg}\Big\{\!\!\Big\{ \prod_{v \in \mathsf{adom}(F)} \mu_v(\lambda(h(v))) \mid h \in \mathsf{Hom}(F^\bullet, D^\bullet) \Big\}\!\!\Big\}$$

where the product is applied component-wise. We say that $(F^\bullet, \mu, \mathsf{agg})$ *is* connected *if the Gaifman graph of* F *is connected.*

As an example, consider the graph schema $\mathbf{S} = \{E\}$ and the homomorphism query (F^a, μ, sum) of output dimension 1 where $F = \{E(a,b), E(b,c), E(c,a)\}$ and μ is constant 1. Then $\mathsf{eval}((F^a, \mu, \mathsf{sum}), (D^u, \lambda))$ counts the homomorphic images of directed triangles in D that pass through u. Note that over this schema, the pattern F of a homomorphism query can be viewed as a directed graph, possibly with isolated vertices. Moreover, the connectedness condition in Definition 2 then coincides with the usual notion of connectedness for directed graphs, edge directions ignored.

The database F in the definition of homomorphism queries can naturally be viewed as a conjunctive query. In [20], the vector $\mathsf{eval}((F^\bullet, \mu, \mathsf{agg}), (D^\bullet, \lambda))$ is called a *generalized homomorphism number*. This is because, for $\mathsf{agg} = \mathsf{sum}$, it is identical to weighted homomorphism numbers [17].

Definition 3 (DHN Layer)**.** *Let* $\mathbf{S}$ *be a schema. An* $\mathbf{S}$-DHN layer with input dimension $d \geq 0$ and output dimension $d' \geq 0$ *is a pair* $\mathcal{L} = (\mathcal{F}, \mathsf{com})$ *such that* $\mathcal{F}$ *is a sequence of homomorphism queries* $(F_1^\bullet, \mu_1, \mathsf{agg}_1), \ldots, (F_m^\bullet, \mu_m, \mathsf{agg}_m)$ *over* $\mathbf{S}$ *of input dimension* d *and* $\mathsf{com}\colon \mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_m} \to \mathbb{R}^{d'}$ *is the* combination function *with* d_i *the output dimension of* $(F_i^\bullet, \mu_i, \mathsf{agg}_i)$*. Given an* $\mathbb{R}^d$*-embedded* $\mathbf{S}$*-database* (D, λ)*,* $\mathcal{L}$ *produces an* $\mathbb{R}^{d'}$*-embedded* $\mathbf{S}$*-database* $(D, \lambda_\mathcal{L})$ *defined by setting, for each value* $u \in \mathsf{adom}(D)$*:*

$$\lambda_\mathcal{L}(u) = \mathsf{com}(\mathsf{eval}((F_1^\bullet, \mu_1, \mathsf{agg}_1), (D^u, \lambda)), \ldots, \mathsf{eval}((F_m^\bullet, \mu_m, \mathsf{agg}_m), (D^u, \lambda))).$$

As an example, we construct a DHN layer that determines whether at least half of the vertices in a graph have a reflexive loop. Let $q_1 = (F_1^a, \mu_1, \mathsf{sum})$ and $q_2 = (F_2^a, \mu_2, \mathsf{sum})$ be homomorphism queries of output dimension 1, where $F_1 = \{\mathsf{adom}(a), E(b,b)\}$, $F_2 = \{\mathsf{adom}(a), \mathsf{adom}(b)\}$, and all transformation functions are constant 1. Further let $\mathcal{L} = ((q_1, q_2), \mathsf{com})$ be the DHN layer with $\mathsf{com}(\ell, n) = 2\ell - n$. Then $\lambda_\mathcal{L}(u) \geq 0$ if at least half of the vertices in D have a reflexive loop.

Definition 4 (Deep Homomorphism Network)**.** *Let* $\mathbf{S}$ *be a schema. A* deep homomorphism network (DHN) over $\mathbf{S}$ *is a tuple* $\mathcal{N} = (\mathcal{L}_1, \ldots, \mathcal{L}_k, \mathsf{cls})$ *where each* $\mathcal{L}_i$ *is an* $\mathbf{S}$*-DHN layer and* $\mathsf{cls}\colon \mathbb{R}^{d'} \to \{0,1\}$ *is a* classification function*, with* $d' \geq 0$*. The input dimension of* $\mathcal{L}_1$ *must be* 0*, the output dimension of* $\mathcal{L}_i$ *must be identical to the input dimension of* $\mathcal{L}_{i+1}$ *for* $1 \leq i < k$*, and the output dimension of* $\mathcal{L}_k$ *must be* d'*. We call* $\mathcal{N}$ *an* agg-DHN*, with* $\mathsf{agg} \in \{\mathsf{sum}, \mathsf{mean}, \mathsf{max}\}$*, if* agg *is the only aggregation function used in* $\mathcal{N}$*, and* $\mathcal{N}$ *is* connected *if every homomorphism query in it is.*

For an $\mathbf{S}$*-database* D*, the DHN* $\mathcal{N}$ *produces a sequence* $(D, \lambda^0_{\mathcal{N},D}), \ldots, (D, \lambda^k_{\mathcal{N},D})$ *of embedded* $\mathbf{S}$*-databases, where* $\lambda^0_{\mathcal{N},D}$ *assigns to every value the empty vector* $()$ *and* $\lambda^i_{\mathcal{N},D}$ *is the result of applying layer* $\mathcal{L}_i$ *to* $(D, \lambda^{i-1}_{\mathcal{N},D})$*, for* $1 \leq i \leq k$*. The DHN* $\mathcal{N}$ accepts *the pointed* $\mathbf{S}$*-database* D^v *if* $\mathsf{cls}(\lambda^k_{\mathcal{N},D}(v)) = 1$*. It* rejects *this database otherwise.*

The combination and classification functions, as well as the transformation functions in homomorphism queries may be represented as feed-forward neural networks (FNNs), defined in the appendix. The parameters of these networks are then learned during the training phase of a DHN. We put special emphasis on FNNs that use the *truncated ReLU* $\mathrm{ReLU}^*(x) = \min(\mathsf{max}(0, x), 1)$ as the activation function. However, all our results also apply to the case of non-truncated $\mathrm{ReLU}(x) = \mathsf{max}(0, x)$, since $\mathrm{ReLU}^*(x) = \mathrm{ReLU}(x) - \mathrm{ReLU}(x-1)$ for all $x \in \mathbb{R}$. In a *simple* DHN, all combination and transformation functions must be represented by FNNs with ReLU^* activation, without hidden layers and with rational weights, and the classification function must be a threshold function; that is, for some $c \in \mathbb{Q}$, $i \in \{1, \ldots, d'\}$, and $\sim \in \{>, \geq\}$, we must have $\mathsf{cls}(x_1, \ldots, x_{d'}) = 1$ if $x_i \sim c$ and

$\mathsf{cls}(x_1, \ldots, x_{d'}) = 0$ otherwise. More details are in the appendix. It is easy to see and was already observed in [20] that DHNs over the graph schema $\mathbf{S} = \{E\}$ generalize GNNs. This even applies to GNNs with global readout when homomorphism queries may be disconnected, but not otherwise.

Remark 5. *In the literature, the most common approach for applying a graph neural network (GNN) to a database D is to first translate D into a graph whose vertices are the facts of D and whose edges are derived from the foreign keys in D [6, 8, 30, 31]. Since a GNN associates embeddings vectors with vertices of the input graph, this corresponds to associating embedding vectors with facts of the input database D. In contrast, DHNs as considered in this paper associate embedding vectors with values from $\mathsf{adom}(D)$. This discrepancy, however, can easily be resolved: increase the arity of every database relation by one and place a unique value in the fresh component of every fact, across all relations. The embedding vectors assigned to these unique values then correspond to the embedding vectors assigned to facts. We show in the appendix that, for databases with such* unique fact identifiers*, every GNN running on a graph translation of a database D can be converted into an equivalent DHN running on D itself. Note that unique vertex classifiers also allow a DHN to classify facts rather than only values, which seems natural in many database applications.*

Decision Problems. We consider two static analysis problems for DHNs. In the *emptiness problem*, the question is whether a given DHN accepts no pointed databases whatsoever. This corresponds to the classifier being unsatisfiable. In the *containment problem*, the input consists of two DHNs $\mathcal{N}_1, \mathcal{N}_2$ and the question is whether for every pointed database $G^\bullet$, acceptance by $\mathcal{N}_1$ implies acceptance by $\mathcal{N}_2$. In other words, the containment problem asks whether the vertex classifier represented by $\mathcal{N}_2$ is at least as general as the vertex classifier represented by $\mathcal{N}_1$. Note that there is no obvious way to finitely represent an arbitrary DHN in which no restrictions are placed on the form of the combination, transformation, and classification functions. In static analysis, we thus assume that all of these functions are given by FNNs in which all coefficients are rational numbers, and that the classification function is a threshold function as in simple DHNs.

UNFO and (G)HML. The formulas over a schema $\mathbf{S}$ of the *unary negation fragment of first-order logic with counting (UNFOC)* are defined according to the following syntax rule:

$$\varphi(\bar{x}) ::= R(\bar{x}) \mid \varphi(\bar{x}) \wedge \varphi(\bar{x}) \mid \varphi(\bar{x}) \vee \varphi(\bar{x}) \mid \exists^{\geq n} y\, \varphi(\bar{x}, y) \mid \neg\varphi(x_i)$$

where R ranges over the relation symbols in $\mathbf{S}$, $n \in \mathbb{N}$, and, in the last clause, φ has no free variables besides (possibly) x_i. The fragment of UNFOC in which n is restricted to 1 is the *unary negation fragment of first-order logic (UNFO)* [26]. Note that in UNFOC it is in general not possible to express $\exists^{\leq n} x\, \varphi$ as $\neg\exists^{\geq n+1} x\, \varphi$ because $\exists^{\geq n+1} x\, \varphi$ may have more than one free variable. This is analogous to the asymmetry between existential and universal quantification in UNFO. We use UNFO(C) to classify values and therefore focus, without further notice, on formulas in one free variable. For example, over the graph schema $\mathbf{S} = \{E\}$, the UNFOC formula

$$\varphi(x) = \exists^{\geq 2} y \left(E(x, y) \wedge \neg\exists^{\geq 3} w\, E(y, w) \wedge \exists z \left(E(x, z) \wedge E(y, z)\right)\right) \qquad (*)$$

defines the vertices that have at least two successors of outdegree at most two with which it shares a common successor.

We next introduce *homomorphism modal logic (HML)*, which may be viewed as a normal form for UNFO formulas. It was introduced already in [26], though not under this name. While we do not use modal syntax, HML can naturally be viewed as a polyadic modal logic in which the modal diamond operator is replaced by a generalized operator based on conjunctive queries (CQs), see [26]. HML formulas are defined according to the rule

$$\varphi(x) ::= \neg\varphi(x) \mid \varphi(x) \vee \varphi(x) \mid \exists \bar{y}\, \psi(x, \bar{y})$$

where $\psi(x, \bar{y})$ is a conjunction of relational atoms $R(\bar{z})$ and unary formulas $\varphi(z)$. We emphasize that, up to the latter, formulas $\exists \bar{y}\, \psi(x, \bar{y})$ are nothing but conjunctive queries and thus the expressive power of HML is closely linked to database query languages. The name HML mentions homomorphisms because the semantics of CQs is defined naturally in terms of homomorphisms. Note that unary atoms $P(x)$ can be expressed by formulas of the form $\exists \bar{y}\, \psi(x, \bar{y})$ with $\bar{y}$ empty. We say that an HML formula is *connected* if for every subformula $\exists \bar{y}\, \psi(x, \bar{y})$, (i) some relational atom in the conjunction ψ uses x and (ii) the following undirected graph is connected: the vertices are x and the variables in $\bar{y}$ and two variables participate in an edge if they co-occur in a relational atom in the conjunction ψ. It was proved in [26] that UNFO and HML have the same expressive power.

We next define an extension of HML that has the same expressive power as UNFOC. *Graded homomorphism modal logic (GHML)* is defined in the same way as HML, except that formulas of the form $\exists \bar{y}\, \psi(x, \bar{y})$ are replaced by formulas of the form $\exists^{\geq k_1} y_1 \cdots \exists^{\geq k_m} y_m \bigvee_{1 \leq i \leq \ell} \psi_i(x, y_1, \ldots, y_m)$ with ψ_i of the same shape as the formulas ψ in HML. A GHML formula is *connected* if in every existentially quantified subformula, all formulas ψ_i satisfy Conditions (i) and (ii). As an example, consider the UNFOC formula $(*)$. It does not fall within GHML because the subformula $\exists z\,(E(x, z) \wedge E(y, z))$ contains two free variables. However, it is equivalent to the GHML formula

$$\varphi'(x) = \exists^{\geq 2} y\, \exists^{\geq 1} z\, \big(E(x, y) \wedge E(x, z) \wedge E(y, z) \wedge \neg \exists^{\geq 3} w\, E(y, w)\big).$$

We show that GHML may be viewed as a normal form for UNFOC.

Lemma 6. *UNFOC and GHML have the same expressive power.*

3 Logical Characterizations of DHNs

We relate DHNs with different aggregation functions to variants of UNFO and HML. The following theorem summarizes our results on the expressive power of max-DHNs and sum-DHNs.

Theorem 7.

1. max*-DHNs, simple* max*-DHNs, UNFO, and HML all have the same expressive power.*
2. *Simple* sum*-DHNs are at least as expressive as GHML and UNFOC.*
3. *On databases of bounded degree, the following have the same expressive power: connected* sum*-DHNs, simple connected* sum*-DHNs, and connected GHML.*

UNFOC is not explicitly mentioned in Point 3 because it is awkward to define a connectedness condition for it. However, the resulting logic would have exactly the same expressive power as connected GHML. The connectedness assumption in Point 3 of the theorem cannot be dropped. For instance, consider the vertex property in directed graphs that includes all vertices of a graph if at least half of the vertices in G have a reflexive loop, and no vertices of G otherwise. This property is definable by a disconnected sum-DHN, but it is not expressible in FO, even on graphs of bounded degree, and thus also not in GHML. If the constant degree bound in Point 3 is strengthened to a constant bound on the size of the database, which amounts to considering non-uniform expressive power, then the statement remains true even without the assumption of connectedness, as our proofs still apply. Also the bounded degree assumption cannot be dropped from Point 3. In a sense, this is well-known: even GNNs with sum aggregation can express vertex properties that do not fall within FO such as vertices that have more red than blue successors. What is more surprising is that even when restricted to properties expressible in FO, the expressive power of connected sum-DHNs exceeds that of UNFOC.

Proposition 8. *There is a connected* sum*-DHN that expresses an FO property in the signature* $\mathbf{S} = \{E\}$ *of directed graphs and is not equivalent to a UNFOC formula.*

A concrete such property is *local transitivity* of a vertex v in a graph G, meaning that for all $u_1, u_2 \in V(G)$, $(v, u_1), (u_1, u_2) \in E(G)$ implies $(v, u_2) \in E(G)$.

By virtue of Theorem 7, we may use logics in place of DHNs to analyze static analysis problems for DHNs. Especially in the case of undecidability results and lower complexity bounds, this is much more convenient than working directly with DHNs. The following theorem summarizes our results on the static analysis of max-DHNs and sum-DHNs.

Theorem 9.

1. *The emptiness problem and the containment problem for* max*-DHNs are decidable.*
2. *The emptiness problem and the containment problem for connected* sum*-DHNs on databases of bounded degree are decidable and* coNEXPTIME*-complete.*
3. *The emptiness problem and the containment problem for* sum*-DHNs are undecidable. This already holds (a) for connected simple* sum*-DHNs on graphs and (b) for (not necessarily connected) simple* sum*-DHNs on graphs of bounded degree.*

The precise computational complexity of the problems in Point 1 of Theorem 9 remains open. We note, however, that the translation from HML to max-DHNs is polynomial (while the converse translation is not), and thus the 2EXPTIME-hardness of (finite) satisfiability and (finite) implication in HML, which as a rather direct consequence of Theorem 2 in [19], transfers to the emptiness and containment problems of max-DHNs. The reader might want to compare Point 3(a) of Theorem 9 to the result from [4] that, in an analogous setting, emptiness and containment of GNNs is decidable. What we mean here are sum-GNNs with truncated ReLU activation and with aggregation over both successors and predecessors (separately), but without global readout.

Turning towards mean aggregation, we define an extension RHML of HML with ratio quantifiers, in analogy with the extension of modal logic by such quantifiers considered in [25]. Unlike the logics considered up to this point, RHML is *not* a fragment of first-order logic, and neither of monadic second-order logic. We refrain from considering an analogous extension of UNFO because it is not clear how it can be defined. An RHML formula is defined according to the rule

$$\varphi(x) ::= \neg\varphi(x) \mid \varphi(x) \vee \varphi(x) \mid \exists_{\geq t}\, \bar{y}\, (\mu(x,\bar{y}), \nu(x,\bar{y})) \mid \exists_{>t}\, \bar{y}\, (\mu(x,\bar{y}), \nu(x,\bar{y})),$$

where $t \in \mathbb{Q}$, $\mu(x,\bar{y})$ is a conjunction of unary formulas of the form $\varphi(z)$, and $\nu(x,\bar{y})$ is a conjunction of relational atoms $R(\bar{z})$. In particular, unary atoms of the form $P(z)$ may occur in both μ and ν. The semantics of the RHML formula $\xi = \exists_{\geq t}\, \bar{y}\, (\mu(x,\bar{y}), \nu(x,\bar{y}))$, with $|\bar{y}| = n$, is as follows:

$$D \models \xi(v) \text{ if and only if } \frac{|\{\overline{w} \in \mathsf{adom}(D)^n \mid D \models \mu(v,\overline{w}) \wedge \nu(v,\overline{w})\}|}{|\{\overline{w} \in \mathsf{adom}(D)^n \mid D \models \nu(v,\overline{w})\}|} \geq t.$$

As a special case, we put $D \models \xi(v)$ if the denominator of the above fraction is 0. The semantics of the $\exists_{>t}$ quantifier is defined in the same way, except that such formulas are not satisfied when the denominator is 0. As an example, we can define in RHML over the graph schema the set of all vertices v such that the number of triangles rooted at v whose vertices all have a reflexive loop is at least 50% of the total number of triangles rooted at v:

$$\exists_{\geq 0.5}\, yz \big((E(x,x) \wedge E(y,y) \wedge E(z,z)), (E(x,y) \wedge E(y,z) \wedge E(z,x))\big).$$

Theorem 10.

1. mean-*DHNs are at least as expressive as RHML.*
2. *On databases of bounded degree, the following have the same expressive power: connected* mean-*DHNs, connected simple* mean-*DHNs, and connected RHML.*

Point 1 does not mention *simple* mean-DHNs because we use a discontinuous activation function (Heaviside), cf. [25].

4 Deep Embedding Networks and Unary Quantifier Alternation

Section 3 leaves open the question of a logic that characterizes the expressive power of sum-DHNs on databases of unrestricted degree. In this section, we establish an improved lower bound, that is, we identify a novel fragment of first-order logic that is strictly more expressive than UNFOC and can be expressed by sum-DHNs. This fragment is the *unary quantifier alternation fragment (UQAFO)*, replacing the unary negation of UNFO with unary quantifier alternation. While UQAFO does not include counting quantifiers, it admits inequalities and can express such quantifiers, and thus generalizes UNFOC. In the presentation as a modal logic, UQAFO corresponds to *embedding modal logic (EML)* which is defined like HML, but with isomorphic embeddings in place of homomorphisms. This leads us to also introduce *deep embedding networks (DENs)*, defined exactly like DHNs, but with homomorphisms replaced by isomorphic embeddings; see [18] for a related proposal. From now on, we assume that it is understood what we mean by an *(isomorphic) embedding query*, a *DEN layer*, and a *DEN*. It is an easy consequence of Theorem 1 that sum-DHNs and sum-DENs have the same expressive power. Interestingly, although motivated by sum aggregation, UQAFO turns out to have exactly the same expressive power as max-DENs.

The formulas over schema $\mathbf{S}$ of the *unary quantifier alternation fragment of first-order logic (UQAFO)* are the formulas $\varphi(\bar{x})$ formed according to the following syntax rule:

$$\begin{aligned}
\varphi(\bar{x}) &::= \varphi_\exists(\bar{x}) \mid \varphi_\forall(\bar{x}) \mid \varphi(\bar{x}) \circ \varphi(\bar{x}) \\
\varphi_\exists(\bar{x}) &::= R(\bar{x}) \mid \neg R(\bar{x}) \mid \varphi_\exists(\bar{x}) \circ \varphi_\exists(\bar{x}) \mid x_i = x_j \mid x_i \neq x_j \mid \exists \bar{y}\, \varphi_\exists(\bar{x},\bar{y}) \mid \varphi_\forall(x_i) \\
\varphi_\forall(\bar{x}) &::= R(\bar{x}) \mid \neg R(\bar{x}) \mid \varphi_\forall(\bar{x}) \circ \varphi_\forall(\bar{x}) \mid x_i = x_j \mid x_i \neq x_j \mid \forall \bar{y}\, \varphi_\forall(\bar{x},\bar{y}) \mid \varphi_\exists(x_i)
\end{aligned}$$

where "$\circ$" ranges over "$\vee$" and "$\wedge$". It is easy to see that quantifier alternation is only possible for unary formulas: if $\exists \bar{x}\, \psi$ is a UQAFO formula and $\forall \bar{y}\, \vartheta$ is a subformula of ψ that is not nested within another quantifier in ψ, then $\forall \bar{y}\, \vartheta$ has at most one free variable, and likewise when $\exists$ and $\forall$ are swapped. For example, the formula $\exists x \exists y\, (\forall z\, R(x,z) \wedge \forall z\, S(y,z))$ is a UQAFO formula while the formula $\exists x \exists y \forall z\, (E(x,z) \wedge E(z,y))$ is not. Like for UNFO, we concentrate on UQAFO formulas in one free variable.

We next observe that UQAFO is strictly more expressive than UNFOC. The separating property is local transitivity.

Lemma 11. *UQAFO is strictly more expressive than UNFOC.*

We next introduce a presentation of UQAFO in the same modal style as HML and GHML. *Embedding modal logic (EML)* is defined exactly like HML except that in formulas of the form $\exists \bar{y}\, \psi(x, \bar{y})$, the subformula $\psi(x, \bar{y})$ may now also contain negated relational atoms $\neg R(\bar{z})$ and inequalities $z_1 \neq z_2$. It is shown in the appendix that, regarding the expressive power, this is equivalent to replacing in HML homomorphisms with isomorphic embeddings, which also explains the name EML. Just like UQAFO, EML does not become more expressive when quantifiers $\exists \bar{y}\, \psi(x, \bar{y})$ are replaced with a counting version. We remark that EML is closely related to the modal logic GML($\mathcal{T}$) recently introduced in [18].

Theorem 12.

1. *UQAFO, EML,* max*-DENs and simple* max*-DENs have the same expressive power.*
2. max*-DENs are strictly more expressive than* max*-DHNs, and the same is true for their simple versions.*
3. *simple* sum*-DENs are strictly more expressive than UQAFO. In particular, there exists a first-order property that is not expressible in UQAFO, but by a simple* sum*-DEN can express.*
4. sum*-DENs and* sum*-DHNs have the same expressive power. Also, simple* sum*-DHNs are strictly more expressive than UQAFO.*

The separating property for Point 3 concerns directed graphs with three unary relation symbols, P, Q, and R. We are interested in all vertices v in a graph G such that v and all vertices reachable from it form a linear order $v = v_0, \ldots, v_n$, with $n \bmod 3 = 2$, in which each vertex satisfies exactly one of P, Q, and R: v_i satisfies P if $i \bmod 3 = 0$, Q if $i \bmod 3 = 1$, and R if $i \bmod 3 = 2$. We call this property *PQR-order*. A separating property for Point 2 is local transitivity. Regarding Point 4, it remains open whether the equivalence between sum-DENs and sum-DHNs continues to hold when both models are restricted to their simple version.

Barceló et al. [3] studied GNNs in which the initial vertex features include homomorphism counts for a finite set of pointed pattern graphs. While it is obvious and was already observed in [20] that DHNs subsume such GNNs, a strict separation had been left open. Making use of Theorem 12 we show that, already on undirected graphs, sum-DHNs can express properties that are not expressible by a sum-GNN with initial homomorphism counts. In fact, this is even true for max-DENs. We call the separating property the *sun property*: vertices that lie on a 6-cycle on which every vertex has a degree 1 neighbor. The sun property can be expressed as an UQAFO formula:

$$\varphi_{\text{☼}}(x_1) = \exists x_2 \cdots x_6 \Big(\bigwedge_{1 \leq i \leq 6} (E(x_i, x_{(i \bmod 6)+1}) \wedge \exists y\, (E(x_i, y) \wedge \exists^{=1} z\, E(y,z)) \wedge \bigwedge_{i<j\leq 6} x_i \neq x_j) \Big)$$

where $\exists^{=1} z\, E(y,z)$ is shorthand for $\exists z\, E(y,z) \wedge \forall z, z'\, (\neg E(y,z) \vee \neg E(y,z') \vee z = z')$. By Theorem 12, the sun property can thus also be expressed by a sum-DHN and by a max-DEN. We prove that $\varphi_{\text{☼}}$ cannot be expressed by GNNs with initial homomorphism counts, even when they are equipped with global readout, by reusing the result by Chen et al. that homomorphism counts cannot detect isolated vertices [7].

Theorem 13. *On both undirected and directed graphs,* sum*-DHNs are strictly more expressive than* sum*-GNNs with global readout and homomorphism count features.*

5 An Upper Bound via More Expressive Counting

To identify an upper bound on the expressive power of sum-DHNs, we propose a logic HML+C that is in the spirit of first-order logic with counting as studied in descriptive complexity [13]. Our development closely follows that of Grohe [10] who introduces a certain version of first-order logic with counting called FO+C and then adapts his development to the guarded fragment GFO+C. HML+C is an adaptation in the same style, based on HML.

HML+C is a two-sorted logic, with an individual sort that ranges over the active domain of the database and a number sort, ranging over $\mathbb{N}$. We denote number variables with n, m and individual variables with x, y. The *(numerical) terms* and *(individual) formulas* of HML+C are defined by the following mutually recursive syntax rules:

$$\begin{array}{rcl} t(x,\bar{n}) & ::= & c \mid n \mid t + t \mid \#(\bar{y}, m_1 < t_1, \ldots, m_k < t_k).\psi(x,\bar{y},\bar{n},m_1,\ldots,m_k) \\ \varphi(x,\bar{n}) & ::= & \neg\varphi(x,\bar{n}) \mid \varphi(x,\bar{n}) \vee \varphi(x,\bar{n}) \mid t(x,\bar{n}) \leq t'(x,\bar{n}) \end{array}$$

where c ranges over $\mathbb{N}$, each *bounding term* t_i may use x, all variables from $\bar{y}$, as well as the variables $m_1, \ldots, m_{i-1}$ and all variables in $\bar{n}$, and ψ is a conjunction of relational atoms $R(\bar{z})$ where $\bar{z}$ contains only x and variables from $\bar{y}$, as well as formulas with one free individual variable $\chi(z,\bar{n},m_1,\ldots,m_k)$ where z is x or a variable from $\bar{y}$. We call such a formula ψ a *pattern formula*. Note that HML+C may be viewed as a modal logic in the sense that all formulas and terms have at most one free individual variable. On the other hand, the numerical part of HML+C is exactly that of FO+C as defined in [10]: every FO+C formula that does not use individual variables is also an HML+C formula. It is easy to see that the multiplication of counting terms $t \cdot t'$ can be defined in HML+C, see the appendix.

The semantics of HML+C is just the standard semantics of multi-sorted FO, with the exception of the counting terms. Let $t(x,\bar{n})$ be an HML+C term, D a database, $u \in \mathsf{adom}(D)$, and $\bar{c} \in \mathbb{N}^{|\bar{n}|}$. We write $[\![t]\!]^D(u,\bar{c})$ to denote the value of the term t in D under the assignment that sends x to u and $\bar{n}$ to $\bar{c}$. The semantics of counting terms is then defined by setting $[\![\#(\bar{y}, m_1 < t_1, \ldots, m_k < t_k).\psi]\!]^{(D,\alpha)}$ to be the number of tuples $\bar{a}\bar{b}$ with $\bar{a} \in \mathsf{adom}(D)^{|\bar{y}|}$ and $\bar{b} = (b_1, \ldots, b_k) \in \mathbb{N}^k$ such that

$$b_i < [\![t_i]\!]^{(D,\alpha[\bar{y}/\bar{a}][m_1/b_1]\cdots[m_{i-1}/b_{i-1}])} \text{ for } 1 \leq i \leq k \quad \text{and} \quad [\![\psi]\!]^{(D,\alpha[\bar{y}/\bar{a}][\bar{m}/\bar{b}])} = 1$$

where $\bar{m} = (m_1, \ldots, m_k)$. The following example HML+C formula expresses local transitivity:

$$\#(y,z).(E(x,y) \wedge E(y,z)) \;\leq\; \#(y,z).(E(x,y) \wedge E(y,z) \wedge E(x,z)).$$

A dyadic rational is a rational number of the form $\frac{z}{2^s}$ with $z \in \mathbb{Z}$ and $s \in \mathbb{N}$. Equivalently, a rational is dyadic if its binary expansion is finite. We say that an DHN *uses continuous dyadic rational piecewise-linear activation* if the activation function in all FNNs in it is piecewise-linear with finitely many pieces and without discontinuities at the thresholds, and all slopes, intercepts, and thresholds are dyadic rationals. For instance, ReLU and truncated ReLU satisfy these properties. A DHN *uses dyadic rational coefficients* if every FNN in it does. With a *simple classification function*, we mean a classification function as required for simple DHNs.

Our main result in this section is the following.

Theorem 14. *Every* sum*-DHN that uses continuous dyadic rational piecewise-linear activation, dyadic rational coefficients, and a simple classification function is equivalent to an HML+C formula.*

By Theorem 12, the same is also true for sum-DENs and max-DENs. Also recall that, by Theorem 7, max-DHNs can even be translated into the fragment HML of HML+C. We conjecture that the expressive power of HML+C is strictly greater than that of sum-DHNs. It was observed in [10] that the logic GFO+C studied there can define the vertex property of having even degree, and that GNNs cannot do that. It is easy to see that also HML+C can express this property while there is no obvious way to express it using a DHN in which all functions are represented as FNNs, cf. Remark 7.7 in [10].

6 Experiments

To experimentally evaluate our theoretical findings, we concentrate on the fact that sum-DHNs have higher expressive power than max-DHNs (Theorems 7 and 12 and Lemma 11) and than GNNs with

Table 1: Local Transitivity Experiments

Architecture	F1	AUROC
sum-DHN	1.00	1.00
max-DHN	0.84	0.93
GCN	0.76	0.86
GraphSAGE	0.78	0.89
GIN	0.93	0.98

Table 2: Sun Property Experiments

Architecture	F1	AUROC
sum-DHN	1.00	1.00
GIN with hom. counts	0.85	0.93
GCN	0.76	0.84
GraphSAGE	0.69	0.50
GIN	0.89	0.87

initial homomorphism counts (Theorem 13). The relevant properties are local transitivity and the sun property. Our experiments are based on synthetic datasets. For both properties, we construct datasets that contain a similar number of positive and negative examples. For local transitivity, we start with the disjoint union of many linear-order graphs, and then randomly delete edges. This results in 1921 out of 4000 vertices being locally transitive. For the sun property, we start with a disjoint union of 6-cycles and then randomly attach fresh vertices and other 6-cycles in a way that results in 600 out of 1200 vertices satisfying the property.

For local transitivity, we equip DHNs with all 13 connected homomorphism queries (up to injective homomorphic equivalence) that use at most three variables, no reflexive loops, and no back-and-forth edges between two element, as such loops and edges do not occur in the dataset. For the sun property, we use a 6-cycle query and a single-edge query, as there are no other obvious candidate patterns for our dataset. The symmetric nature of the 6-cycle results in an excessive number of homomorphisms, leading to performance issues. We handle these by restricting to injective homomorphisms in the sun property case. We represent DHN transformation and combination functions by FNNs with 1 and 3 hidden layers respectively, and with leaky ReLU activation, the leakiness serving to alleviate vanishing gradients.

We compare the performance of DHNs to the standard GCN [16], GraphSAGE [12], and GIN [33] implementations provided by PyTorch Geometric. For all models, we use an embedding dimension of 32, three message passing layers, the Adam optimizer, and single batch training. DHNs are trained with a learning rate of 0.0003 for 1000 epochs, and the other models with a learning rate of 0.001 for 10000 epochs. We split the datasets using 6:2:2 ratios in training, validation and test set, and report results on the test set. All experiments were performed on the CPU of a MacBook Pro with M5 chip. More details on the experimental setup can be found in Appendix E of the extended version. Code and datasets are available under `https://github.com/relational-dhn/DHN-experiments`.

Results are reported in Tables 1 and 2. Both experiments confirm the theoretical findings for which they were designed. Moreover, DHNs perform consistently better than all other models. Only the performance of GINs on the local transitivity dataset is remarkable. We believe that it is due to the relative simplicity of the data, on which comparing the number of successors with the number of descendants at distance two should enable a good approximation of local transitivity.

7 Conclusion and Limitations

DHNs provide a principled neural architecture for relational data, as homomorphism-based message passing connects directly to conjunctive queries. We have developed a logical account of the expressive power of DHNs over relational databases. Our results fully characterize max aggregation, while exact characterizations for sum aggregation and mean aggregation over databases of unrestricted degree remain open. Our results offer a stepping stone toward linking DHNs to SQL-style query languages, cf. [2]. Further directions include tighter integration of DHNs with database management systems, both for optimized training and inference and for end-to-end training of neuro-relational pipelines combining DHNs with relational query processing. Finally, making DHNs usable out of the box will require principled pattern-selection methods. For this, results by Barceló et al. [3] and by Jin et al. [15] can provide a starting point.

Limitations. As discussed above, several theoretical questions remain open, as does a more systematic account of the relationship between DHNs and relational databases. Our implementation should be viewed as a proof of concept rather than a highly optimized system. Although some static analysis problems for DHNs turn out to be decidable, their worst-case complexity is high, and developing practically efficient algorithms for them is a non-trivial challenge.

Acknowledgments and Disclosure of Funding

Carsten Lutz was supported by DFG project LU 1417/4-1. This work is partly supported by BMFTR (Federal Ministry of Research, Technology and Space) in DAAD project 57616814 (SECAI, School of Embedded Composite AI) as part of the program Konrad Zuse Schools of Excellence in Artificial Intelligence. The work of Benny Kimelfeld was funded by the ISF grant 863/25.

References

[1] Linus Bao, Emily Jin, Michael M. Bronstein, İsmail İlkan Ceylan, and Matthias Lanzinger. Homomorphism counts as structural encodings for graph learning. In *Proc. of ICLR*. OpenReview.net, 2025.

[2] Vince Bárány, Balder ten Cate, and Martin Otto. Queries with guarded negation. *Proc. VLDB Endow.*, 5(11):1328–1339, 2012.

[3] Pablo Barceló, Floris Geerts, Juan Reutter, and Maksimilian Ryschkov. Graph neural networks with local graph parameters. In *Proc. of NeurIPS*, pages 25280–25293, 2021.

[4] Michael Benedikt, Chia-Hsuan Lu, Boris Motik, and Tony Tan. Decidability of graph neural networks via logical characterizations. In *Proc. of ICALP*, pages 127:1–127:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[5] Surajit Chaudhuri. An overview of query optimization in relational systems. In *Proc. of PODS*, pages 34–43. ACM Press, 1998.

[6] Tianlang Chen, Charilaos I. Kanatsoulis, and Jure Leskovec. RelGNN: Composite message passing for relational deep learning. In *Proc. of ICML*. PMLR / OpenReview.net, 2025.

[7] Yijia Chen, Jörg Flum, Mingjun Liu, and Zhiyang Xun. On algorithms based on finitely many homomorphism counts. *Information and Computation*, 306:105326, 2025.

[8] Matthias Fey, Weihua Hu, Kexin Huang, Jan Eric Lenssen, Rishabh Ranjan, Joshua Robinson, Rex Ying, Jiaxuan You, and Jure Leskovec. Position: Relational deep learning - graph representation learning on relational databases. In *Proc. of ICML*, pages 13592–13607. PMLR / OpenReview.net, 2024.

[9] Martin Grohe. The logic of graph neural networks. In *Proc. of LICS*, pages 1–17. IEEE, 2021.

[10] Martin Grohe. The descriptive complexity of graph neural networks. *TheoretiCS*, 3, 2024.

[11] Zhiwei Guo and Heng Wang. A deep graph neural network-based mechanism for social recommendations. *IEEE Trans. Ind. Informatics*, 17(4):2776–2783, 2021.

[12] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proc. of NeurIPS*, pages 1024–1034, 2017.

[13] Neil Immerman. *Descriptive complexity*. Springer Science & Business Media, 2012.

[14] John J. Irwin, Teague Sterling, Michael M. Mysinger, Erin S. Bolstad, and Ryan G. Coleman. ZINC: a free tool to discover chemistry for biology. *Journal of chemical information and modeling*, 52(7):1757–1768, 2012.

[15] Emily Jin, Michael M. Bronstein, İsmail İlkan Ceylan, and Matthias Lanzinger. Homomorphism counts for graph neural networks: All about that basis. In *Proc. of ICML*, pages 22075–22098. PMLR / OpenReview.net, 2024.

[16] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. of ICLR*. OpenReview.net, 2017.

[17] László Lovász. *Large networks and graph limits*. American Mathematical Soc., 2012.

[18] Huan Luo and Jonni Virtema. Unifying approach to uniform expressivity of graph neural networks. *arXiv preprint arXiv:2602.18409*, 2026.

[19] Carsten Lutz. The complexity of conjunctive query answering in expressive description logics. In *Proc. of IJCAR*, number 5195 in LNAI, pages 179–193. Springer, 2008.

[20] Takanori Maehara and Hoang NT. Deep homomorphism networks. In *Proc. of NeurIPS*, pages 56076–56107. Curran Associates, Inc., 2024.

[21] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and Leman go neural: Higher-order graph neural networks. In *Proc. of AAAI*, pages 4602–4609, 2019.

[22] Hoang Nguyen and Takanori Maehara. Graph homomorphism convolution. In *Proc. of ICML*, pages 7306–7316. PMLR, 2020.

[23] Chendi Qian, Gaurav Rattan, Floris Geerts, Mathias Niepert, and Christopher Morris. Ordered subgraph aggregation networks. In *Proc. of NeurIPS*, pages 21030–21045, 2022.

[24] Moritz Schönherr, Balder ten Cate, Maurice Funk, Benny Kimelfeld, Carsten Lutz, and Arie Soeteman. Expressive power of deep homomorphism networks over relational databases. *arXiv preprint arXiv:2605.22852*, 2026.

[25] Moritz Schönherr and Carsten Lutz. Logical characterizations of GNNs with mean aggregation. In *Proc. of AAAI*. AAAI Press, 2026.

[26] Luc Segoufin and Balder ten Cate. Unary negation. *Log. Methods Comput. Sci.*, 9(3), 2013.

[27] Kartik Sharma, Yeon-Chang Lee, Sivagami Nambi, Aditya Salian, Shlok Shah, Sang-Wook Kim, and Srijan Kumar. A survey of graph neural networks for social recommender systems. *ACM Comput. Surv.*, 56(10):265, 2024.

[28] Arie Soeteman and Balder ten Cate. Logical expressiveness of graph neural networks with hierarchical node individualization. In *Proc. of NeurIPS*, 2025.

[29] Ruoxi Sun, Hanjun Dai, and Adams Wei Yu. Does GNN pretraining help molecular representation? In *Proc. of NeurIPS*, 2022.

[30] Jan Tönshoff, Neta Friedman, Martin Grohe, and Benny Kimelfeld. Stable tuple embeddings for dynamic databases. In *Proc. of ICDE*, pages 1286–1299. IEEE, 2023.

[31] Minjie Wang, Quan Gan, David Wipf, Zheng Zhang, Christos Faloutsos, Weinan Zhang, Muhan Zhang, Zhenkun Cai, Jiahang Li, Zunyao Mao, Yakun Song, Jianheng Tang, Yanlin Zhang, Guang Yang, Chuan Lei, Xiao Qin, Ning Li, Han Zhang, Yanbo Wang, and Zizhao Zhang. 4DBInfer: A 4D benchmarking toolbox for graph-centric predictive modeling on RDBs. In *Proc. of NeurIPS*, 2024.

[32] Hinrikus Wolf, Luca Oeljeklaus, Pascal Kühner, and Martin Grohe. Structural node embeddings with homomorphism counts. *CoRR*, abs/2308.15283, 2023.

[33] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *Proc. of ICLR*. OpenReview.net, 2019.

[34] Jiaxuan You, Jonathan M Gomes-Selman, Rex Ying, and Jure Leskovec. Identity-aware graph neural networks. In *Proc. of AAAI*, pages 10737–10745, 2021.

[35] DingYi Zeng, Wanlong Liu, Wenyu Chen, Li Zhou, Malu Zhang, and Hong Qu. Substructure aware graph neural networks. In *Proc. of AAAI*, pages 11129–11137. AAAI, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All claims made in the abstract and introduction are backed up by formal theorems and/or experimental result presented in the paper. Conversely, all results obtained in the paper are summarized in the intro.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: There is a "Limitations" subsection in the conclusions

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Every result is formally stated in the paper, including all necessary assumptions, and comes with a rigorous proof in the full version of this paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: We provide a link to the code and data and all our experiments are fully reproducible.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code and data is publicly available. It contains instructions how to reproduce the results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes].

Justification: All training and test details are provided. The most important ones are in the main body of the paper and the rest is in the full version.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: While we give only the results of one training run in the main part, we report the mean and standard deviation across multiple training runs in the full version, which support that the results are significant. For the limited experiments and small sample we believe this to be appropriate.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer:[Yes]

   Justification: We provide full information regarding computing resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: We have reviewed the NeurIPS Code of Ethics and fully comply to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A].

    Justification: The paper presents foundation research about the expressive power of GNNs and we are not aware of any direct paths to negative applications or societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not provide models or data of this kind.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: We use standard tools such as PyTorch and give proper credit in the paper. All data used is self-generated.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: Instructions on how to use our datasets and code are included in the linked Git repository.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: LLMs have been used to improve some formulations in the paper text, but in no other way.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.