Lost in Aggregation

On a Fundamental Expressivity Limit of Message-Passing Graph Neural Networks

Anonymous authors
Paper under double-blind review

Abstract

We define a generic property for aggregation functions, capturing a vast range of practical aggregations, and prove that any Message-Passing Graph Neural Network (MP-GNN) model with such aggregations induces only a polynomial number of equivalence classes on all graphs - while the number of non-isomorphic graphs is super-exponential (in number of vertices). Adding a familiar perspective, we observe that merely 2-iterations of Color Refinement (CR) induce at least an exponential number of equivalence classes, making the aforementioned MP-GNNs relatively infinitely weaker.

Previous studies state that sum-aggregation MP-GNNs match full CR however they consider a weak, 'non-uniform', notion of distinguishing-power where each graph size may require a different MP-GNN to distinguish graphs up to that size.

Our results concern both distinguishing between non-equivariant vertices and distinguishing between non-isomorphic graphs.

1 Introduction

Message-Passing Graph Neural Networks (MP-GNNs) (Kipf & Welling, 2017; Gilmer et al., 2017) are a class of parameterized algorithms for graphs, often used as architectures in graph learning tasks. Such tasks may be learning on graphs that represent molecules and biological structures (Gilmer et al., 2017; Gaudelet et al., 2021), graphs that represent social networks and knowledge bases (Yasunaga et al., 2021), and graphs that represent combinatorial-optimization problems (Tönshoff & Grohe, 2025; Tönshoff et al., 2023). Hence, characterizing the expressivity of MP-GNNs is of great importance.

An MP-GNN is defined by a sequence of layers $L_1, \ldots, L_m$ for some $m \in \mathbb{N}$, each layer $L_t = (\mathsf{mlp}_t, \mathrm{agg}_t, \mathrm{msg}_t)$ comprising a message; aggregation; and combination functions. The combination is implemented always by a Multilayer Perceptron (MLP), and in this paper all MLPs are ReLU-activated and rationally-weighted. Denote by $N_G(v)$ and $v^{(0)}$ the neighborhood and initial feature of a vertex v in a graph G, respectively, then v's value after applying layer $t+1$ is

$$v^{(t+1)} := \mathsf{mlp}_{t+1}(v^{(t)}, \mathrm{agg}_{t+1}\{\!\{\, \mathrm{msg}_{t+1}(v^{(t)}, w^{(t)}) | w \in N_G(v) \}\!\})$$

That is, the layers are applied sequentially, each layer applied in parallel to all vertices: Computing a message for each neighbor; aggregating the messages; and combining the aggregation value with the subject-vertex value. Note that the aggregation can be any function on multisets, with a fixed output-dimension, and in this paper a computable one. For graph-level tasks, an MP-GNN model

has a final *readout* step $R = (\mathsf{mlp}_R, \mathrm{agg}_R)$ comprising an aggregation of the final vertices' values followed by the operation of a final MLP. Denote the readout value for a graph G by $G^{(R)}$, then

$$G^{(R)} := \mathsf{mlp}_R(\mathrm{agg}_R\{\{v^{(m)} : v \in V(G)\}\})$$

The MLP part of the layers gives MP-GNNs their learnability qualities. The node-level definition of the algorithm, together with the fixed-dimension output aggregation, mean every GNN model can technically be applied to graphs of all sizes and degrees. Finally, no order or unique-ids of the nodes are considered, only the nodes' features and graph structure, hence GNNs are invariant to isomorphism.

A necessary condition for an MP-GNN model to *express* a function, i.e. approximate it by some ε, is to have the adequate distinguishing-power i.e. to output different values for every two inputs on which the function differs (by 2ε). Thus, we are interested in the distinguishing-power of MP-GNN architectures. A well-studied algorithm for *distinguishing* vertices and graphs is the Color Refinement (CR) algorithm (a.k.a. Weisfeiler-leman algorithm (Morgan, 1965; Weisfeiler & Leman, 1968), see also (Cardon & Crochemore, 1982; Paige & Tarjan, 1987; Berkholz et al., 2017; Grohe, 2021)): An iterative local algorithm which assigns a *color* to each node. In each iteration, the color of each node is updated by adding to it the multiset of its neighbors' current colors. Given a graph G, CR runs for $|G|$ iterations by which point maximum granularity of color-classes is reached. The color of a graph after each iteration is the multiset of current colors of its vertices. For $t \in \mathbb{N}$ we denote the algorithm that runs the first t iterations of CR by $\mathrm{CR}^{(t)}$.

It is known that the distinguishing-power of MP-GNNs is upper-bounded by that of CR (Xu et al., 2019; Morris et al., 2019; Aamand et al., 2022). It has also been shown there that the CR bound is tight i.e. there exists an MP-GNN model that distinguishes graphs and vertices if they are distinguishable by CR, however the proof is in a *non-uniform* notion: It proves existence of a distinguishing model **per graph size**. That setting has limited relevance to practice as it implies that a learned model can be correct only on graphs of sizes up to the maximum training-graph size. Such model will be incorrect in many practical scenarios: When there are not enough resources to train on large graphs or when the graphs grow over time.

The notion by which it is required to have (at least) one model that is correct on graphs of all sizes is called *uniform*, and this is the notion of distinguishing-power and expressivity that we consider in this paper. There, the following are straightforward:

1. MP-GNNs do not subsume the distinguishing-power of CR, if only because the value assigned to a vertex by an MP-GNN with m layers is not affected by nodes of distance $> m$.

2. With an auxiliary-dimension initialized to '1', a trivial sum-aggregation MP-GNN subsumes $\mathrm{CR}^{(1)}$, as the sum of that dimension amounts to the number of neighbors.

3. With no restriction on the aggregation function other than being computable and having a fixed output dimension, MP-GNNs with m layers subsume the distinguishing-power of $\mathrm{CR}^{(m)}$ by having an aggregation that simply implements CR and encodes the state in one rational number. However, the use of such information for an MLP, in expressing a target function, is limited i.e. such aggregation is less relevant to practice.

The above calls for a general characterization of practical aggregations, and for bounding[1] their distinguishing-power by a range narrower than [$\mathrm{CR}^{(1)}$, CR].

Upper bounds that relate directly to function approximation are proved in several works: In terms of logic (Barceló et al., 2020); circuit complexity (Grohe, 2023); or comparative between different MP-GNNs sub-classes (Rosenbluth et al., 2023; Grohe & Rosenbluth, 2024). In all these however, excluding to some extent (Rosenbluth et al., 2023, Section 6), only specific aggregations are considered.

In (Corso et al., 2020) an inexpressivity result for a general class of aggregations is given, however it is proved only for one message-pass iteration; it assumes that the feature-domain is the real numbers - not only finite precision; and it assumes that the aggregation function is continuous. The domain assumption is unnecessarily permissive - with respect to practice - as operations on infinite-precision real numbers are incomputable, and the assumption on the aggregation functions is unnecessarily restrictive as computable functions can be non-continuous.

In (Khalife & Basu, 2023) it is essentially shown that with exponential activation functions, such as *sigmoid,tanh*, the distinguishing-power of MP-GNNs subsumes $\mathrm{CR}^{(2)}$. However, these functions cannot be precisely computed, hence the result does not apply to computable MP-GNNs. See Section 4 (future research (3)) for further discussion.

Recently, a tight bound has been shown (Rosenbluth & Grohe, 2025) both for the distinguishing-power and the expressivity of *recurrent* MP-GNNs (going back to (Scarselli et al., 2008; Gallicchio & Micheli, 2010)), highlighting the missing knowledge about (non-rec.) MP-GNNs even further.

New Results

We consider the domain of graphs with vertices featured over the boolean domain. We describe a general class of aggregation functions (Definition 3.1), which captures most of the reasonable aggregations that do not involve exponentiation or division by a graph-size-dependent value, and we analyze their effect on the distinguishing-power of MP-GNNs.

Denote by $\mathcal{N}$ the class of MP-GNNs comprising (only) such aggregations, denote the number of equivalence classes that an MP-GNN N induces on vertices in graphs of size n, and on whole graphs of size n, by $N_{\mathrm{dp}}(n)$ and $N_{\mathrm{gdp}}(n)$ respectively, and similarly for $\mathrm{CR}^{(2)}$ by $\mathrm{CR}^{(2)}_{\mathrm{dp}}(n)$ and $\mathrm{CR}^{(2)}_{\mathrm{gdp}}(n)$, then we prove the following.

1. The uniform distinguishing-power of each MP-GNN in $\mathcal{N}$ is at most polynomial in the graph size (Theorem 3.8). Formally,

$$\forall N \in \mathcal{N}\ N_{\mathrm{dp}}(n) = \mathrm{poly}(n),\ \ N_{\mathrm{gdp}}(n) = \mathrm{poly}(n)$$

   As the number of non-isomorphic graphs is super-exponential, that upper bound is highly significant.

2. Observing a lower-bound for $\mathrm{CR}^{(2)}_{\mathrm{dp}}$, we add that not only the distinguishing-power of $\mathcal{N}$ is weaker, i.e. there are vertices distinguishable $\mathrm{CR}^{(2)}$ and not by $\mathcal{N}$, but it gets infinitely

[1] To be precise, by referring to $\mathrm{CR}^{(t)}$ as a strict bound we do not imply inclusion but rather that it is not subsumed by MP-GNNs. Obviously, when considering graphs of diameters large than t, there are nodes distinguishable by a trivial MP-GNN with $t+1$ layers and not by $\mathrm{CR}^{(t)}$.

weaker as the graph size grows (Corollary 3.10). Formally,

$$\forall N \in \mathcal{N} \ \lim_{n\to\infty} \frac{N_{\mathrm{dp}}(n)}{\mathrm{CR}^{(2)}_{\mathrm{dp}}(n)} = 0, \ \frac{N_{\mathrm{gdp}}(n)}{\mathrm{CR}^{(2)}_{\mathrm{gdp}}(n)} = 0$$

While we focus on MP-GNNs that consist of ReLU-activated MLPs for their message and combination functions, our results may apply also to other MP-GNNs architectures (Remark 3.7).

2 Preliminaries

By $\mathbb{N}; \mathbb{Z}; \mathbb{Q}$ we denote the natural, integer, and rational numbers respectively. For $m \in \mathbb{N}$ we define $[m] \coloneqq \{i : i \in \mathbb{N}, 1 \leq i \leq m\}$. For a set S and size $m \in \mathbb{N}$ we denote the set of all multisets of size m with elements from S by $\left(\!\!\binom{S}{m}\!\!\right)$, and of any finite size by $\left(\!\!\binom{S}{*}\!\!\right)$. Let X be a multiset of rationals or a multiset of rational vectors, we define $\mathrm{cd}(X)$ to be the *least common denominator* of the elements in X (or elements of its vectors). For a vector $v \in \mathbb{Q}^d$ we define $\dim(v) \coloneqq d$, and for a matrix $W \in \mathbb{Q}^{d_1 \times d_2}$ we define $\dim(W) \coloneqq (d_1, d_2)$.

Encoding and Bit-Length

Let $x \in \mathbb{Q}$ and let $\frac{p}{q} = x, \ p; q \in \mathbb{Z}$ be its *reduced form*, a *fractional representation* of x is a bit-representation that encodes $p; q$ in separate - using any $O(\log n)$ integer encoding. In this paper we assume that all rational values are represented in that way. For $q \in \mathbb{Q}$ we denote its reduced-form fractional-representation bit-length by $\lambda_{\mathfrak{f}}(q)$. For a vector $v \in \mathbb{Q}^d$ we define $\lambda_{\mathfrak{f}}(v) \coloneqq \Sigma_{i\in[d]}\lambda_{\mathfrak{f}}(v(i))$ the bit-length of v. For a sequence or multiset of vectors $S = (v_i)_{i\in[n]}, M = \{\{w_i\}\}_{i\in[m]}$ we define their bit-length $\lambda_{\mathfrak{f}}(S) \coloneqq \Sigma_{i\in[n]}\lambda_{\mathfrak{f}}(v_i), \ \lambda_{\mathfrak{f}}(M) \coloneqq \Sigma_{i\in[m]}\lambda_{\mathfrak{f}}(w_i)$. For $d, k \in \mathbb{N}$ we define $\mathbb{Q}^d_k \coloneqq \{q : q \in \mathbb{Q}^d, \lambda_{\mathfrak{f}}(q) \leq k\}$ the dimension-d rational vectors of bit-length no greater than k.

Featured Graph

A (vertex) *featured graph* $G = \langle V(G), E(G), S, Z(G)\rangle$ is a 4-tuple being the usual undirected graph definition, with the addition of a *feature map* $Z(G) : V(G) \to S$ which maps each vertex to a value in some set S. For $v \in V(G)$ we define $N_G(v) \coloneqq \{w \in V(G) : vw \in E(G)\}$ the neighborhood of v. For $v \in V(G)$ we denote $Z(G)(v)$ also by $Z(G, v)$. We define the *order*, or *size*, of a graph G to be the number of its vertices i.e. $|G| \coloneqq |V(G)|$. We denote the domain of graphs featured over a set S by $\mathcal{G}_S$ and the set of all featured graphs by $\mathcal{G}_*$. In this paper we consider graphs with boolean input-features, and denote this domain by $\mathcal{G}_{\mathcal{B}}$, that is, $\mathcal{G}_{\mathcal{B}} \coloneqq \{G \mid \forall v \in V(G) \ Z(G)(v) \in \{0, 1\}\}$. For a graph domain $\mathcal{G} \subseteq \mathcal{G}_*$, and $n \in \mathbb{N}$, we define $\mathcal{G}(n) \coloneqq \{G \in \mathcal{G} : |G| = n\}$ the graphs in $\mathcal{G}$ of size n. We denote the set of all feature maps that map to some set T by $\mathcal{Z}_T$, and we denote the set of all feature maps by $\mathcal{Z}_*$. Let $\mathcal{G} \subseteq \mathcal{G}_*$ be a domain, a mapping $f : \mathcal{G} \to \mathcal{Z}_*$ to new feature maps is called a *feature transformation*, and for $d \in \mathbb{N}$ a mapping $f : \mathcal{G} \to \mathbb{Q}^d$ is called a *graph embedding*.

Multilayer Perceptron

A ReLU-activated Multilayer Perceptron (MLP) $F = (l_1, \ldots, l_m), \ l_i = (w_i, b_i)$, of I/O dimensions $d_{in}; d_{out}$, and depth m, is a sequence of rational matrices w_i and bias vectors b_i such that

$$\dim(w_1)(2) = d_{in}, \dim(w_m)(1) = d_{out},$$

$$\forall i > 1 \ \dim(w_i)(2) = \dim(w_{i-1})(1), \ \forall i \in [m] \dim(b_i) = \dim(w_i)(1)$$

It defines a function $f_F(x)$, which we denote also by $F(x)$, such that

$$f_F(x) \coloneqq w_m(...\mathrm{ReLU}(w_2(\mathrm{ReLU}(w_1(x) + b_1)) + b_2)...) + b_m$$

Message-Passing Graph Neural Network

A *Message Passing Graph Neural Network* (MP-GNN) of depth m and dimensions $r_0, \{p_i, q_i, r_i\}_{i\in[m]}$

$$N = (\mathsf{mlp}_1, \mathrm{agg}_1, \mathrm{msg}_1), \ldots, (\mathsf{mlp}_m, \mathrm{agg}_m, \mathrm{msg}_m)$$

is a sequence of m triplets, referred to as *layers*, such that for $i \in [m]$ layer i comprises a message and aggregation functions and an MLP,

$$\mathrm{msg}_i : \mathbb{Q}^{r_{i-1}} \times \mathbb{Q}^{r_{i-1}} \to \mathbb{Q}^{p_i},\ \mathrm{agg}_i : \left(\!\!\left(\begin{matrix}\mathbb{Q}^{p_i}\\ *\end{matrix}\right)\!\!\right) \to \mathbb{Q}^{q_i},\ \mathsf{mlp}_i : \mathbb{Q}^{r_{i-1}} \times \mathbb{Q}^{q_i} \to \mathbb{Q}^{r_i}$$

The message function is usually either $(x, y) \mapsto y$ or an MLP, but not necessarily. In this paper we will assume it is an MLP i.e. the more expressive among the two. The aggregation function is typically per-dimension sum; mean; or max, but can also be other functions that operate on a multiset and have a fixed output-dimension. The sequence of layers defines a feature transformation $f_N : \mathcal{G}_{\mathbb{Q}^{r_0}} \to \mathcal{Z}_{\mathbb{Q}^{r_m}}$ as follows: Let $G \in \mathcal{G}_{\mathbb{Q}^{r_0}}$ and $v \in V(G)$, then we define:

1. $v_N^{(0)} \coloneqq N^{(0)}(G, v) \coloneqq Z(G)(v)$ the initial value of v.
2. $\forall t \in [m] \quad v_N^{(t)} \coloneqq N^{(t)}(G, v) \coloneqq \mathsf{mlp}_t\Big(v_N^{(t-1)}, \mathrm{agg}_t(\{\!\{\, \mathrm{msg}_t(v_N^{(t-1)}, w_N^{(t-1)}) \mid w \in N_G(v)\}\!\})\Big)$ the value of v after applying the first t layers of N.
3. $N(G, v) \coloneqq N^{(m)}(G, v)$ the final value of v.

If in addition to its layers N includes a readout step $R = (\mathsf{mlp}_R, \mathrm{agg}_R)$, then it defines a graph embedding:

$$N(G) \coloneqq \mathsf{mlp}_R(\mathrm{agg}_R\{\!\{v_N^{(m)} \mid v \in V(G)\}\!\})$$

Color Refinement

Let $G \in \mathcal{G}_*$. For $t \geq 0$ and $v \in V(G)$ we define the *color of v after t iterations*, notated $\mathsf{cr}_G^{(t)}(v)$, inductively: The initial value of v is its initial feature, that is, $\mathsf{cr}_G^{(0)}(v) \coloneqq Z(G)(v)$, and for all $t > 0$ we define

$$\mathsf{cr}_G^{(t)}(v) \coloneqq (\mathsf{cr}_G^{(t-1)}(v), \{\!\{\mathsf{cr}_G^{(t-1)}(w) \mid w \in N_G(v)\}\!\})$$

Maximum color-classes granularity is reached after at most $|G|$ iterations, hence we define the *color of v* to be $\mathsf{cr}_G(v) \coloneqq \mathsf{cr}_G^{|G|}(v)$. We define the color of G at iteration t, and overall, to be

$$\mathsf{cr}^{(t)}(G) \coloneqq \{\!\{\mathsf{cr}^{(t)}(v) \mid v \in V(G)\}\!\}, \quad \mathsf{cr}(G) \coloneqq \mathsf{cr}^{|G|}(G)$$

3 Limited by Aggregation

We start with defining the aggregation class that is our main focus. Our characterization relates to the information complexity of an aggregation's output. We would like our definition to be general and capture a wide range of practical aggregations, and at the same time imply a significant upper bound on distinguishing-power. As we consider rational numbers, our characterization must account also for the common denominator of the values to-be-aggregated.

Definition 3.1 (Logarithmic Aggregation). *Let $d, d' \in \mathbb{N}$, $agg : \left(\!\!\left(\begin{smallmatrix}\mathbb{Q}^d \\ *\end{smallmatrix}\right)\!\!\right) \to \mathbb{Q}^{d'}$ be an algorithm from a multiset of rational vectors to a single rational vector. We denote by $S_{agg} : \mathbb{N}^3 \to \mathbb{N}$ the output complexity of agg, depending on the number of vectors n, maximum bit-length k of any vector, and the bit-length of the common denominator of all values.*

$$S_{agg}(n, k, \ell) \coloneqq \max\left(\lambda_{\mathfrak{f}}(agg(M)) : M \in \left(\!\!\left(\begin{smallmatrix}\mathbb{Q}_k^d \\ n\end{smallmatrix}\right)\!\!\right), \lambda_{\mathfrak{f}}(\mathrm{cd}(M)) \leq \ell\right)$$

In addition, we denote by $S^{\mathrm{cd}}_{agg} : \mathbb{N}^3 \to \mathbb{N}$ the complexity of the common denominator of n aggregations on subsets of a multiset of n vectors, bit-length k per vector, and multiset-common-denominator of bit-length ℓ, that is,

$$S^{\mathrm{cd}}_{agg}(n, k, \ell) \coloneqq \max\left(\lambda_{\mathfrak{f}}(\mathrm{cd}(M)) : M = \{\!\{agg(M_1), \ldots, agg(M_n)\}\!\},\right.$$
$$\left. M_i \subseteq M', M' \in \left(\!\!\left(\begin{smallmatrix}\mathbb{Q}_k^d \\ n\end{smallmatrix}\right)\!\!\right), \lambda_{\mathfrak{f}}(\mathrm{cd}(M')) \leq \ell\right)$$

We say that agg is logarithmic*, notated $\gamma(agg)$, if and only if for every $f, g : \mathbb{N} \to \mathbb{N}$ such that $f(n) = O(\log n), g(n) = O(\log n)$ it holds that:*

1. *$S_{agg}(n, f(n), g(n)) = O(\log n)$.*
2. *$S^{\mathrm{cd}}_{agg}(n, f(n), g(n)) = O(\log n)$.*

An example where S_{agg} is potentially non-logarithmic is the aggregation in Graph Attention Networks (Veličković et al., 2017) and Graph Transformers (Dwivedi & Bresson, 2020), which uses the *softmax* function - involving exponentiation by the input as well as unbounded division. For the arithmetic mean, the condition on $S^{\mathrm{cd}}_{\mathrm{mean}}$ does not hold: Let $n \in \mathbb{N}$ and define $M_n = \{1, 0, \ldots, 0\}, |m| = n$, and subsets $M'_{n,i} = \{1, 0, \ldots, 0\}, |M'_{n,i}| = i + 1, i \in [n-1]$, then we have $\mathrm{mean}(M'_{n,i}) = \frac{1}{i}$, hence $\lim_{n\to\infty} \ln(\mathrm{cd}(\{\!\{M'_{n,i}\}\!\}_{i\in[n-1]})) = n$.

However, a vast range of aggregations is logarithmic. The following theorem proves that all aggregations defined by either of two general formula are logarithmic, and the subsequent example puts it to use in showing several commonly-used aggregations to be logarithmic. (See appendix for proofs details).

Theorem 3.2. *The following per-dimension aggregations $agg : \left(\!\!\left(\begin{smallmatrix}\mathbb{Q}^d \\ *\end{smallmatrix}\right)\!\!\right) \to \mathbb{Q}^{d'}$ are logarithmic:*

1. *$agg(M) \coloneqq p_2(\Sigma_{x\in T} p_1(x)), T \subseteq M$, for rational polynomials $p_1, p_2 : \mathbb{Q} \to \mathbb{Q}$.*
2. *$agg(M) \coloneqq (agg_1, \ldots, agg_a), \gamma(agg_i), a \in \mathbb{N}$. That is, the concatenation of a fixed number of logarithmic aggregations.*

Example 3.3. *The following common aggregations are logarithmic:*

1. *sum.*
2. *Selection of k elements, for a fixed $k \in \mathbb{N}$, by any criteria e.g. highest; lowest; quintile.*
3. *k-bins agg bin-aggregation, for a fixed $k \in \mathbb{N}$ and logarithmic aggregation agg.*

We proceed to quantify the distinguishing-power of MP-GNNs and $\mathrm{CR}^{(2)}$.

Distinguishing-Power.

Let $\mathcal{G}$ be a graph domain, and N be an MP-GNN, we define the *distinguishing-power* of N on $\mathcal{G}$, $N_{\text{dp}}(\mathcal{G})$, to be the number of vertices equivalence-classes that N induces on $\mathcal{G}$. That is,

$$N_{\text{dp}}(\mathcal{G}) \coloneqq |\{N(G, v) : G \in \mathcal{G}, v \in V(G)\}|$$

Similarly, for $\text{CR}^{(2)}$ we define

$$\text{CR}^{(2)}_{\text{dp}}(\mathcal{G}) \coloneqq |\{\mathsf{cr}^{(2)}_G(v) : G \in \mathcal{G}, v \in V(G)\}|$$

For distinguishing between graphs, we define

$$\text{N}_{\text{gdp}}(\mathcal{G}) \coloneqq |\{N(G) : G \in \mathcal{G}\}|,\ \text{CR}^{(2)}_{\text{gdp}}(\mathcal{G}) \coloneqq |\{\mathsf{cr}^{(2)}(G) : G \in \mathcal{G}\}|$$

When the domain is defined with a size parameter, i.e. $\mathcal{G} = \mathcal{G}'(n)$ for some domain $\mathcal{G}'$, we may refer to the distinguishing-power as a function $f : \mathbb{N} \to \mathbb{N}$. For example, for N_{dp}, $f(n) \coloneqq N_{\text{dp}}(\mathcal{G}'(n))$.

Main Result

Our fundamental result (Theorem 3.8) is that for any MP-GNN N comprising (only) logarithmic aggregations, the distinguishing-power of N is polynomial i.e.

$$N_{\text{dp}}(\mathcal{G}_{\mathcal{B}}(n)) = \text{poly}(n),\ N_{\text{gdp}}(\mathcal{G}_{\mathcal{B}}(n)) = \text{poly}(n)$$

To prove our main result, we take the following steps:

1. For any MP-GNN N, we define an information-complexity measure as a function of the graph size, $L_N : \mathbb{N} \to \mathbb{N}$, and observe that $N_{\text{dp}}(\mathcal{G}_{\mathcal{B}}(n)), N_{\text{gdp}}(\mathcal{G}_{\mathcal{B}}(n)) \leq 2^{L_N(n)}$. (Lemma 3.4)
2. We prove that for logarithmic-aggregations it holds that $L_N(n) = O(\log n)$. (Lemma 3.6)
3. We conclude the result from combining (1) and (2).

Information Complexity.

We measure the information conveyed by the aggregations in an MP-GNN's computation, as follows. Let $N = (\mathsf{mlp}_1, \text{agg}_1, \text{msg}_1), \ldots, (\mathsf{mlp}_m, \text{agg}_m, \text{msg}_m), (\mathsf{mlp}_R, \text{agg}_R)$ be an MP-GNN, possibly with a final-readout layer. For a graph $G \in \mathcal{G}_{\mathcal{B}}$ and a vertex $v \in V(G)$ we define $I_N(G, v)$ to be the sequence of values produced by the aggregations operating on (G, v), that is,

$$I_N(G, v) \coloneqq \left(\text{agg}_i\{\!\{\, \text{msg}_i(v^{(i-1)}, w^{(i-1)}) : w \in N_G(v)\}\!\}\right)_{i \in [m]}$$

, and we define

$$I_N(G) \coloneqq (\text{agg}_R\{\!\{N(G, v) : v \in V(G)\}\!\})$$

to be the information produced by the readout aggregation - if such exists. We define the complexity of I_N, $L_N : \mathbb{N} \to \mathbb{N}$, to be the maximum bit-length of I_N, that is, for a feature transformation

$$L_N(n) \coloneqq \max\left(\lambda_{\mathfrak{f}}(I_N(G, v)) : G \in \mathcal{G}_{\mathcal{B}},\ |G| = n, v \in V(G)\right)$$

, and for a graph embedding

$$L^{(E)}_N(n) \coloneqq \max\left(\lambda_{\mathfrak{f}}(I_N(G)) : G \in \mathcal{G}_{\mathcal{B}},\ |G| = n\right)$$

The reason for the specific definition of L_N is the following key observation.

Lemma 3.4. *Let* $N = (\mathsf{mlp}_1, agg_1, \mathrm{msg}_1), \ldots, (\mathsf{mlp}_m, agg_m, \mathrm{msg}_m), (\mathsf{mlp}_R, agg_R)$ *be an MP-GNN, then*

$$N_{\mathrm{dp}}(\mathcal{G}_{\mathcal{B}}(n)) \leq 2^{L_N(n)+1},\ N_{\mathrm{gdp}}(\mathcal{G}_{\mathcal{B}}(n)) \leq 2^{L_N^{(E)}(n)}$$

Proof. For $I_N(G, v)$, we show by induction on m that

$$\forall G, G' \in \mathcal{G}_{\mathcal{B}}\ \forall v \in V(G)\ \forall v' \in V(G')\ Z(G)(v) = Z(G)(v') \wedge I_N(G, v) = I_N(G', v') \Rightarrow$$

$$N(G, v) = N(G', v')$$

For $m = 1$, $I_N(G, v) = \mathrm{agg}_1\{\!\{\mathrm{msg}_1(v^{(0)}, w^{(0)}) : w \in N_G(v)\}\!\}$, hence $Z(G)(v) = Z(G')(v') \wedge I_N(G, v) = I_N(G', v') \Rightarrow \mathsf{mlp}_1(v^{(0)}, \mathrm{agg}_1(\{\!\{\mathrm{msg}_1(v^{(0)}, w^{(0)}) : w \in N_G(v)\}\!\})) = \mathsf{mlp}_1(v'^{(0)}, \mathrm{agg}_1(\{\!\{\mathrm{msg}_1(v'^{(0)}, w^{(0)}) : w \in N_{G'}(v')\}\!\})) \Rightarrow N(G, v) = N(G', v')$. Assuming correctness for $m = k$, we prove for $m = k + 1$. By the induction assumption, $Z(G)(v) = Z(G')(v') \wedge I_N(G, v) = I_N(G', v') \Rightarrow v^{(k)} = v'^{(k)}$. Also, $I_N(G, v) = I_N(G', v') \Rightarrow \mathrm{agg}_{k+1}\{\!\{\mathrm{msg}_{k+1}(v^{(k)}, w^{(k)}) : w \in N_G(v)\}\!\} = \mathrm{agg}_{k+1}\{\!\{\mathrm{msg}_{k+1}(v'^{(k)}, w^{(k)}) : w \in N_{G'}(v')\}\!\}$. Hence, $\mathsf{mlp}_{k+1}(v^{(k)}, \mathrm{agg}_{k+1}(\{\!\{\mathrm{msg}_{k+1}(v^{(k)}, w^{(k)}) : w \in N_G(v)\}\!\})) = \mathsf{mlp}_{k+1}(v'^{(k)}, \mathrm{agg}_{k+1}(\{\!\{\mathrm{msg}_{k+1}(v'^{(k)}, w^{(k)}) : w \in N_{G'}(v)\}\!\})) \Rightarrow N(G, v) = N(G', v')$.

For $I_N(G)$, $I_N(G) = I_N(G') \Rightarrow \mathrm{agg}_R\{\!\{N(G, v) : v \in V(G)\}\!\} = \mathrm{agg}_R\{\!\{N(G', v') : v' \in V(G')\}\!\} \Rightarrow \mathsf{mlp}_R(\mathrm{agg}_R\{\!\{N(G, v) : v \in V(G)\}\!\}) = \mathsf{mlp}_R(\mathrm{agg}_R\{\!\{N(G', v) : v \in V(G')\}\!\}) \Rightarrow N(G) = N(G')$.

Hence, the distinguishing-power of N is upper-bounded by the number of possible values of I_N (and initial feature, in case of a feature transformation), which in turn is upper-bounded exponentially by the maximum bit-length of I_N (plus the 1 bit of initial feature, in case of a feature transformation). □

The output of the aggregation of each layer in an MP-GNN depends on the output of the computation steps preceding it, hence, in order to calculate the total aggregations' output-complexity we need to calculate the intermediate-value complexity through the MP-GNN's computation steps. Before considering the complete MP-GNN's computation, we first observe the output-complexity of a single MLP. One may see why the following is true, as an MLP's effect on the magnitude of the input is limited - it is *Lipschitz continuous*, and also its effect (using ReLU activation) on the denominator of input values is bounded. Nevertheles, proof details can be found in the appendix.

Lemma 3.5. *Let* F *be an MLP of input dimension* d*, we define* $S_F : \mathbb{N} \to \mathbb{N}$ *be the output-size complexity of* F*, that is,*

$$S_F(k) \coloneqq \max(\lambda_{\mathfrak{f}}(F(x)) : x \in \mathbb{Q}^d, \lambda_{\mathfrak{f}}(x) = k)$$

In addition, we denote by $S^{\mathrm{cd}}_{agg} : \mathbb{N}^3 \to \mathbb{N}$ *the complexity of the common denominator of* n *applications of* F *on elements of a multiset of* n *vectors, bit-length* k *per vector, and multiset-common-denominator of bit-length* ℓ*, that is,*

$$S^{\mathrm{cd}}_F(n, k, \ell) \coloneqq \max\left(\lambda_{\mathfrak{f}}(\mathrm{cd}(M)) : M = \{\!\{F(x_i)\}\!\}_{i \in [m]}, x_i \in M', M' \in \left(\!\!\binom{\mathbb{Q}^d_k}{n}\!\!\right), \lambda_{\mathfrak{f}}(\mathrm{cd}(M')) \leq \ell\right)$$

Then,

1. $S_F(k) = O(k)$.
2. *For every* $f : \mathbb{N} \to \mathbb{N}$ *such that* $f(n) = O(\log n)$ *it holds that* $S^{\mathrm{cd}}_F(n, k, f(n)) = O(\log n)$.

We will now state and prove our main lemma. Note that it is stated in the context of our graph domain $\mathcal{G}_{\mathcal{B}}$ where the features are boolean thus their bit-length trivially does not exceed $O(\log n)$.

Lemma 3.6. *Let $N = (\mathsf{mlp}_1, agg_1, \mathrm{msg}_1), \ldots, (\mathsf{mlp}_m, agg_m, \mathrm{msg}_m), (\mathsf{mlp}_R, agg_R)$ be an MP-GNN, possibly with a final-readout layer, then: If all the aggregations are logarithmic then the total-information complexity of N is logarithmic, formally*

$$\forall i\ \gamma(agg_i) \Rightarrow L_N(n) = O(\log n), \ \ \forall i\ \gamma(agg_i) \wedge \gamma(agg_R) \Rightarrow L_N^{(E)}(n) = O(\log n)$$

Proof. For $l \in [m]$ we define $S^{\mathrm{msg}}_{N^{(l)}}, S^{\mathrm{agg}}_{N^{(l)}}, S_{N^{(l)}} : \mathbb{N} \to \mathbb{N}$ the complexities of intermediate outputs throughout the operation of N: The output of msg_l, the output of agg_l, and the output of layer l i.e. the output of mlp_l. Note that these are not the complexities of the standalone functions - which we have defined and discussed earlier. Formally,

$$S^{\mathrm{msg}}_{N^{(l)}}(n) \coloneqq \max\Big(\lambda_{\mathfrak{f}}(\,\mathrm{msg}_l(N^{(l-1)}(G,v), N^{(l-1)}(G,w))) : G \in \mathcal{G}_{\mathcal{B}},\ |G| = n,\ v \in V(G),\ w \in N_G(v)\Big)$$

$$S^{\mathrm{agg}}_{N^{(l)}}(n) \coloneqq \max\Big(\lambda_{\mathfrak{f}}(\mathrm{agg}_l\{\!\{\,\mathrm{msg}_l(N^{(l-1)}(G,v), N^{(l-1)}(G,w)) \mid w \in N_G(v)\}\!\}) :$$
$$G \in \mathcal{G}_{\mathcal{B}},\ |G| = n,\ v \in V(G)\Big)$$

$$S_{N^{(l)}}(n) \coloneqq \max\Big(\lambda_{\mathfrak{f}}(N^{(l)}(G,v)) : G \in \mathcal{G}_{\mathcal{B}},\ |G| = n,\ v \in V(G)\Big)$$

In addition, we define $S^{\mathrm{cd,msg}}_{N^{(l)}}, S^{\mathrm{cd,agg}}_{N^{(l)}}, S^{\mathrm{cd}}_{N^{(l)}} : \mathbb{N} \to \mathbb{N}$ the complexities of the common denominator of intermediate values across the vertices, throughout the operation of N. Formally,

$$S^{\mathrm{cd,msg}}_{N^{(l)}}(n) \coloneqq \max\Big(\lambda_{\mathfrak{f}}\Big(\,\mathrm{cd}\,(\{\!\{\,\mathrm{msg}_l(N^{(l-1)}(G,v), N^{(l-1)}(G,w)) \mid v \in V(G),\ w \in N_G(v)\}\!\})\Big) :$$
$$G \in \mathcal{G}_{\mathcal{B}},\ |G| = n\Big)$$

$$S^{\mathrm{cd,agg}}_{N^{(l)}}(n) \coloneqq \max\Big(\lambda_{\mathfrak{f}}\Big(\,\mathrm{cd}\,(\{\!\{\mathrm{agg}_l\{\!\{\,\mathrm{msg}_l(N^{(l-1)}(G,v), N^{(l-1)}(G,w)) \mid w \in N_G(v)\}\!\} : v \in V(G)\}\!\})\Big) :$$
$$G \in \mathcal{G}_{\mathcal{B}},\ |G| = n\Big)$$

$$S^{\mathrm{cd}}_{N^{(l)}}(n) \coloneqq \max\Big(\lambda_{\mathfrak{f}}(\,\mathrm{cd}(\{\!\{N^{(l)}(G,v) : v \in V(G)\}\!\})) : G \in \mathcal{G}_{\mathcal{B}},\ |G| = n\Big)$$

We prove by induction on l that $\forall\ l \in [m]\ \ S^{\mathrm{msg}}_{N^{(l)}}, S^{\mathrm{agg}}_{N^{(l)}}, S_{N^{(l)}} = O(\log n)$. As the complexity of a sum of a fixed number of $O(\log n)$-complexity functions is $O(\log n)$, and by definition

$$L_N(n) = \max\Big(\Sigma_{l\in[m]}\lambda_{\mathfrak{f}}(\mathrm{agg}_l\{\!\{\,\mathrm{msg}_l\,(N^{(l-1)}(G,v), N^{(l-1)}(G,w))\}\!\}_{w\in N_G(v)}) :$$
$$G \in \mathcal{G}_{\mathcal{B}},\ |G| = n,\ v \in V(G)\Big)$$

, by proving the induction we would have proven that $L_N(n) = \log n$. As part of the induction proof we also proof by induction on l that $\forall\ l \in [m]\ \ S^{\mathrm{cd,msg}}_{N^{(l)}}, S^{\mathrm{cd,agg}}_{N^{(l)}}, S^{\mathrm{cd}}_{N^{(l)}} = O(\log n)$. For $l = 1$, by the initial features all being in $\{0,1\}$, clearly $\exists c \in \mathbb{N} : S^{\mathrm{msg}}_{N^{(1)}}(n), S^{\mathrm{cd,msg}}_{N^{(1)}}(n) \leq c$, hence trivially

$S^{\text{msg}}_{N^{(1)}}(n), S^{\text{cd,msg}}_{N^{(1)}}(n) = O(\log n)$. Then, by assumption on agg_1 we have $S^{\text{agg}}_{N^{(1)}}(n), S^{\text{cd,agg}}_{N^{(1)}}(n) = O(\log n)$. By Lemma 3.5 and since the complexity of a composition of a function of complexity $O(n)$ over a function of complexity $O(\log n)$ is $O(\log n)$, we have that $S_{N^{(1)}}(n), S^{\text{cd}}_{N^{(1)}}(n) = O(\log n)$. Assuming correctness for $l = k < m$ we prove for $l = k + 1$. By by the induction assumption on $S_{N^{(k)}}$ and by Lemma 3.5, $S^{\text{msg}}_{N^{(k+1)}}(n)$ is the complexity of a composition of an $O(n)$-complexity function over the concatenation of two $O(\log n)$-complexity functions, which is $O(\log n)$. Also, by the induction assumption on $S^{\text{cd}}_{N^{(k)}}$ and by Lemma 3.5 $S^{\text{cd,msg}}_{N^{(k+1)}}(n) = O(\log n)$. By assumption on agg_{k+1}, and by $S^{\text{msg}}_{N^{(k+1)}}(n), S^{\text{cd,msg}}_{N^{(k+1)}}(n) = O(\log n)$, we have $S^{\text{agg}}_{N^{(k+1)}}(n), S^{\text{cd,agg}}_{N^{(k+1)}}(n) = O(\log n)$. Finally, by the latter and by Lemma 3.5 we have $S_{N^{(k+1)}}(n), S^{\text{cd}}_{N^{(k+1)}}(n) = O(\log n)$.

For $L^{(E)}_N$, by definition $L^{(E)}_N(n) \coloneqq \max\left(\lambda_{\mathfrak{f}}(\text{agg}_R\{\{N(G,v)\}\}_{v\in V(G)}) : G \in \mathcal{G}_{\mathcal{B}},\ |G| = n\right)$, hence by $S_{N^{(m)}}(n), S^{\text{cd}}_{N^{(m)}}(n) = O(\log n)$, and by assumption on agg_R, we have $L^{(E)}_N(n) = O(\log n)$ □

Remark 3.7. *The line of proof of Lemma 3.6 works for every message and combination functions with output-size complexity, and outputs-common-denominator complexity, $O(n)$ (with n being the function's input size, as well as the outputs-multiset size), not only for ReLU-activated MLPs. Hence, the guarantee that $L_N(n) = O(\log n)$ (with n being the input-graph size), and subsequently Theorem 3.8, hold for all MP-GNNs architectures comprising message and combination functions that have these properties.*

Theorem 3.8. *Let $N = (\mathsf{mlp}_1, agg_1, \text{msg}_1), \ldots, (\mathsf{mlp}_m, agg_m, \text{msg}_m), (\mathsf{mlp}_R, agg_R)$ be an MP-GNN, possibly with a final-readout layer, then: If all aggregations are logarithmic then the distinguishing-power of N is polynomial. Formally,*

$$\forall i\ \gamma(agg_i) \Rightarrow N_{\text{dp}}(\mathcal{G}_{\mathcal{B}}(n)) = \text{poly}(n),\ \forall i\ \gamma(agg_i) \wedge \gamma(agg_R) \Rightarrow N_{\text{gdp}}(\mathcal{G}_{\mathcal{B}}(n)) = \text{poly}(n)$$

Proof. By assumption and Lemma 3.6 we have $L_N(n) = O(\log n),\ L^{(E)}_N(n) = O(\log n)$, hence by Lemma 3.4 we have $N_{\text{dp}}(\mathcal{G}_{\mathcal{B}}(n)) = 2^{O(\log n)} = \text{poly}(n),\ N_{\text{gdp}}(\mathcal{G}_{\mathcal{B}}(n)) = 2^{O(\log n)} = \text{poly}(n)$. □

Comparison to Color Refinement

The absolute-terms upper bounds in Theorem 3.8 are meaningful on their own, considering that the total number of non-isomorphic graphs of size n, $|\mathcal{G}_{\mathcal{B}}(n)|$, is super-exponential. In previous studies, the distinguishing-power of MP-GNNs has been compared to the distinguishing-power of Color Refinement (CR), where it was shown to either match it or not, depending on the setting, with no quantification given for the gap in the latter case. As CR is meaningful and well-studied, we proceed to put Theorem 3.8 in its perspective. We compare the distinguishing-power of logarithmic-aggregations MP-GNNs to the distinguishing-power of merely two iterations of CR, i.e. to $\text{CR}^{(2)}_{\text{dp}}(\mathcal{G}_{\mathcal{B}})$. We observe the following.

Lemma 3.9. *Let $n \in \mathbb{N}$, then*

$$CR^{(2)}_{\text{dp}}(\mathcal{G}_{\mathcal{B}}(2n+3)) \geq \binom{2n}{n},\ CR^{(2)}_{\text{gdp}}(\mathcal{G}_{\mathcal{B}}(2n+3)) \geq \binom{2n}{n}$$

Proof. We look at a two-level star graph of size $2n+3$, where we denote the center by v, the vertices of the first level by $u_i, i \in [n]$ and those of the second level by $w_i, i \in [n]$. We define

$$\mathcal{K}_n \coloneqq \{(k_0, \ldots k_n), k_i \in [0..n], \text{sum}(k_i) = n\}$$

all the possible choices, with repetition, of n elements from $n+1$ types. For $K \in \mathcal{K}_n$ we define $G_K \in \mathcal{G}_\mathcal{B}$ to be the graph where v is connected to all u's, and there are k_i of the u's that are connected to i of the w's. In other words v has k_i neighbors of degree i $(+1)$. In addition, v is connected to two vertices s_1, s_2 to make its degree higher than any of the u's and w's. Formally, G_K is defined as follows: $V(G) = \{v\}, \{s_1, s_2\}\{u_1, \ldots, u_n\}, \{w_1, \ldots, w_n\}$,

$$E(G) = \{vs_1, vs_2\}, \{vu_i : i \in [n]\}, \{u_i w_j : j \in [n], i > \Sigma_{h\in[j-1]} k_h\},\ \forall x \in V(G)\ Z(G)(x) = 1$$

Finally, for $n \in \mathbb{N}$ we define $\mathcal{G}_\mathcal{K}(2n+3) \coloneqq \{G_K : K \in \mathcal{K}_n\} \subset \mathcal{G}_\mathcal{B}$ the set of all graphs of the form above, of size $2n+3$. Observe that $|\mathcal{G}_\mathcal{K}(2n+3)| = \binom{2n}{n}$, as it is the number of options to choose with repetition n elements - the number of u vertices - out of n possible types - the possible number of neighbors. In addition, $\forall n \in \mathbb{N}\ \forall G, G' \in \mathcal{G}_\mathcal{K}(2n+3)\ \ \exists i : |\{j : |N_G(w_j)| = i\}| \neq |\{j : |N_{G'}(w_j)| = i\}|$. Hence, by definition of cr we have $\forall n \in \mathbb{N}\ \forall G, G' \in \mathcal{G}_\mathcal{K}(2n+3)\ \ \mathsf{cr}^{(2)}_G(v) \neq \mathsf{cr}^{(2)}_{G'}(v)$, hence $\mathrm{CR}^{(2)}_{\mathrm{dp}}(\mathcal{G}_\mathcal{B}(2n+3)) \geq \binom{2n}{n}$. Finally, the color of $v \in V(G)$ is unique not only compared to $v \in V(G')$ but also compared to all u's and w's in all graphs, as their degree is $\leq n+1$. Hence, by definition of cr we have $\forall n \in \mathbb{N}\ \forall G, G' \in \mathcal{G}_\mathcal{K}(2n+3)\ \ \mathsf{cr}^{(2)}(G) \neq \mathsf{cr}^{(2)}(G')$, hence $\mathrm{CR}^{(2)}_{\mathrm{gdp}}(\mathcal{G}_\mathcal{B}(2n+3)) \geq \binom{2n}{n}$ $\square$

Combined with Lemma 3.4, we arrive at the following sufficient condition for an MP-GNN having weaker distinguishing-power than $\mathrm{CR}^{(2)}$, and combined with Theorem 3.8 we have the following measure of the gap between the power of logarithmic-aggregations MP-GNNs and that of $\mathrm{CR}^{(2)}$.

Corollary 3.10. *Let $N = (\mathsf{mlp}_1, agg_1, \mathrm{msg}_1), \ldots, (\mathsf{mlp}_m, agg_m, \mathrm{msg}_m), (\mathsf{mlp}_R, agg_R)$ be an MP-GNN, possibly with a final-readout layer, then:*

1. *If there exists $n \in \mathbb{N}$ such that $L_N(2n+3) < \left\lceil \log \binom{2n-1}{n-1} \right\rceil$ then there are vertices that are distinguishable by $CR^{(2)}$ and not by N. Formally, $\exists n \in \mathbb{N} : L_N(2n+3) < \left\lceil \log \binom{2n-1}{n-1} \right\rceil \Rightarrow$*

$$\exists G, G' \in \mathcal{G}_\mathcal{B},\ v \in V(G),\ v' \in V(G') : \mathsf{cr}^{(2)}_G(v) \neq \mathsf{cr}^{(2)}_{G'}(v') \wedge N(G, v) = N(G', v')$$

   In particular, $\forall i\ \gamma(agg_i) \Rightarrow \lim_{n\to\infty} \frac{N_{\mathrm{dp}}(\mathcal{G}_\mathcal{B}(n))}{CR^{(2)}_{\mathrm{dp}}(\mathcal{G}_\mathcal{B}(n))} = 0$

2. *For distinguishing graphs,*

$$\exists n \in \mathbb{N} : L^{(E)}_N(2n{+}3) < \left\lceil \log \binom{2n-1}{n-1} \right\rceil \Rightarrow \exists G, G' \in \mathcal{G}_\mathcal{B} : \mathsf{cr}^{(2)}(G) \neq \mathsf{cr}^{(2)}(G') \wedge N(G) = N(G')$$

$$\forall i\ \gamma(agg_i) \wedge \gamma(agg_R) \Rightarrow \lim_{n\to\infty} \frac{N_{\mathrm{gdp}}(\mathcal{G}_\mathcal{B}(n))}{CR^{(2)}_{\mathrm{gdp}}(\mathcal{G}_\mathcal{B}(n))} = 0$$

4 Concluding Remarks

We have introduced an output-size complexity property for aggregation functions, satisfied by most of the reasonable aggregations that do not involve exponentiation or division by a graph-size-dependent value, and proved that it has the effect of restricting MP-GNN models to distinguish merely a polynomial number of equivalence classes. This applies both to distinguishing between vertices and distinguishing between graphs. Given that the number of non-equivariant vertices or non-isomorphic graphs is super-exponential, these bounds are significant.

We proceeded to take a familiar perspective and considered the well-studied distinguishing-power of the Color Refinement algorithm, already known to upper-bound all MP-GNNs, as a reference point. We have observed that $\mathrm{CR}^{(2)}$, i.e. merely 2 iterations of CR, is not only stronger than our general class of MP-GNNs, making $\mathrm{CR}^{(1)}$ a tight bound [3] for it, but is relatively infinitely stronger, as it is at least exponential. This is in stark contrast to non-uniform distinguishing-power results (Xu et al., 2019; Morris et al., 2019; Aamand et al., 2022), as well as to uniform results for recurrent MP-GNNs (Rosenbluth & Grohe, 2025).

A consequence of our results is that every function, in every function-class that is subsumed by logarithmic-aggregations MP-GNNs, does not distinguish more than a polynomial number of equivalence-classes.

To practice, an immediate implication of our results is that if the target function assumes (on the graph domain) a greater-than polynomial number of values in the graph size then it is simply impossible for a logarithmic-aggregations MP-GNN model to even distinguish between all vertices or graphs that are assigned a different value by the function, let alone assign them the specific function's values.

While we focus on MP-GNNs with ReLU-activated MLPs for message and combination functions, our results hold for all MP-GNNs comprising message and combination functions with output and outputs-common-denominator size-complexities of $O(n)$.

Our goal is to understand fundamental expressivity bounds of MP-GNNs - regardless of aggregations specifics. To that end, the following remain open for further research:

1. We have not addressed aggregations that have enough output bits to represent the number of all possible graphs. There, (full) CR distinguishing-power is potentially given for free - the aggregation function can simply implement CR, and the question to study is that of expressivity i.e. approximating a target function - computing a specific value for each vertex or graph. It is clear that the fixed number of MLP-runs in an MP-GNN with rational weights is a limiting factor, as a runtime-complexity upper bound of

$$O(n^2 \cdot \text{aggregation output-size complexity}) + \text{total aggregation-runtime}$$

   on the run of any MP-GNN should be relatively straightforward. However, a tighter bound, or perhaps one in terms other than runtime-complexity, for the expressivity of MP-GNNs with arbitrary computable aggregations, can be interesting.

2. We have not analyzed the output-size complexity of softmax aggregation. Potentially it is linear, rather than logarithmic , as it involves exponentiation by the input, however, a careful examination of the computation - taking into account the normalization and the actual algorithm for computing it - may prove otherwise.

3. We have not analyzed the output-size complexity of MLPs with non-ReLU activations such as *sigmoid* or *tanh*, which without considering computability have been shown to increase the distinguishing-power of MP-GNNs (Khalife & Basu, 2023). There again, an exponentiation by the input is involved, potentially leading to an exponential rather than linear output-size complexity of the MLP and higher distinguishing-power, yet further analysis is required for a clear characterization.

[3] when the graph diameter does not exceed the number of CR iterations.

References

Anders Aamand, Justin Chen, Piotr Indyk, Shyam Narayanan, Ronitt Rubinfeld, Nicholas Schiefer, Sandeep Silwal, and Tal Wagner. Exponentially improving the complexity of simulating the weisfeiler-lehman test with graph neural networks. *Advances in Neural Information Processing Systems*, 35:27333–27346, 2022.

Pablo Barceló, Egor V. Kostylev, Mikaël Monet, Jorge Pérez, Juan L. Reutter, and Juan Pablo Silva. The logical expressiveness of graph neural networks. In *8th International Conference on Learning Representations (ICLR 2020)*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=r1lZ7AEKvB`.

Christoph Berkholz, Paul Bonsma, and Martin Grohe. Tight lower and upper bounds for the complexity of canonical colour refinement. *Theory of Computing Systems*, 60(4):581–614, 2017.

Alain Cardon and Maxime Crochemore. Partitioning a graph in o (śaślog2śvś). *Theoretical Computer Science*, 19(1):85–98, 1982.

Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. *Advances in neural information processing systems*, 33:13260–13271, 2020.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

C. Gallicchio and A. Micheli. Graph echo state networks. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, 2010.

Thomas Gaudelet, Ben Day, Arian R Jamasb, Jyothish Soman, Cristian Regep, Gertrude Liu, Jeremy BR Hayter, Richard Vickers, Charles Roberts, Jian Tang, et al. Utilizing graph machine learning within drug discovery and development. *Briefings in bioinformatics*, 22(6), 2021.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

Martin Grohe. The logic of graph neural networks. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pp. 1–17. IEEE, 2021. doi: 10.1109/LICS52264.2021.9470677. URL `https://doi.org/10.1109/LICS52264.2021.9470677`.

Martin Grohe. The descriptive complexity of graph neural networks. In *Proceedings of the 38th Annual ACM/IEEE Symposium on Logic in Computer Science*, 2023. doi: 10.1109/LICS56636.2023.10175735.

Martin Grohe and Eran Rosenbluth. Are targeted messages more effective? In *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science*, pp. 1–14, 2024.

Sammy Khalife and Amitabh Basu. On the power of graph neural networks and the role of the activation function. *arXiv preprint arXiv:2307.04661*, 2023.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=SJU4ayYgl`.

H.L. Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of Chemical Documentation*, 5(2):107–113, 1965.

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 *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 4602–4609, 2019.

Robert Paige and Robert E Tarjan. Three partition refinement algorithms. *SIAM Journal on computing*, 16(6):973–989, 1987.

Eran Rosenbluth and Martin Grohe. Repetition makes perfect: Recurrent sum-gnns match message passing limit. *arXiv preprint arxiv:2505.00291*, 2025.

Eran Rosenbluth, Jan Toenshoff, and Martin Grohe. Some might say all you need is sum. *arXiv preprint arXiv:2302.11603*, 2023.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Jan Tönshoff and Martin Grohe. Learning from algorithm feedback: One-shot sat solver guidance with gnns. *arXiv preprint arXiv:2505.16053*, 2025.

Jan Tönshoff, Berke Kisin, Jakob Lindner, and Martin Grohe. One model, any csp: graph neural networks as fast global search heuristics for constraint satisfaction. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 4280–4288, 2023.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

B.Y. Weisfeiler and A.A. Leman. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series 2*, 1968. English translation by G. Ryabov available at `https://www.iti.zcu.cz/wl2018/pdf/wl_paper_translation.pdf`.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=ryGs6iA5Km`.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 535–546, 2021.

A Limited By Aggregation

Lemma A.1. *Let $p : \mathbb{Q} \to \mathbb{Q}$ be a rational polynomial, and let $c_1 \in \mathbb{N}$, then $\exists c_2 \in \mathbb{N} : \forall x \in \mathbb{Q}\ \lambda_{\mathfrak{f}}(x) \leq c_1 \log n \ \Rightarrow \lambda_{\mathfrak{f}}(p(x)) \leq c_2(\log n)$.*

Proof. Assume $p(x) = \Sigma_{i=0}^{|p|} \frac{a_i}{b_i} x^i$, and let $\frac{y}{z} \in \mathbb{Q}$. Then,

$$p(\frac{y}{z}) = \Sigma_{i=0}^{|p|} \frac{a_i y^i z^{|p|} \prod_{j=0}^{|p|} b_j}{z^i b_i z^{|p|} \prod_{j=0}^{|p|} b_j} =$$

$$\frac{1}{z^{|p|} \prod_{j=0}^{|p|} b_j} \Sigma_{i=0}^{|p|} a_i y^i z^{|p|-i} \prod_{j \neq i} b_j$$

Hence, $\lambda_{\mathfrak{f}}(p(\frac{y}{z})) \leq \log(z^{|p|} \prod_{j=0}^{|p|} b_j) + \log(\Sigma_{i=0}^{|p|} a_i y^i z^{|p|-i} \prod_{j \neq i} b_j) \leq \log(z^{|p|} \prod_{j=0}^{|p|} b_j) + \Sigma_{i=0}^{|p|} \log(a_i y^i z^{|p|-i} \prod_{j \neq i} b_j) = O(\log(z^{|p|})) + \Sigma_{i=0}^{|p|} O(\log(y^i z^{|p|-i})) \leq O(\log(z^{|p|})) + |p|^2 2c_1 O(\log n) \leq 3|p|^2 c_1 O(\log n) = O(\log n)$ □

Lemma A.2. *Let $c_1, c_2, n \in \mathbb{N}$, then $\exists c_3, c_4 \in \mathbb{N}$ such that $\forall M \in \left(\!\!\binom{\mathbb{Q}}{n}\!\!\right), \max_{x \in M}(\lambda_{\mathfrak{f}}(x)) \leq c_1 \log n, \lambda_{\mathfrak{f}}(\mathrm{cd}(M)) \leq c_2 \log n$ it holds that $\lambda_{\mathfrak{f}}(\Sigma_{x \in M} x) \leq c_3(\log n)$*

Proof. Let $n \in \mathbb{N},\ M \in \left(\!\!\binom{\mathbb{Q}}{n}\!\!\right),\ M = \{\{\frac{p_i}{q_i}\}\}_{i \in [n]}$, and define $q \coloneqq \mathrm{cd}(M)$, then

$$\lambda_{\mathfrak{f}}(\Sigma_{i \in [n]} \frac{p_i}{q_i}) = \lambda_{\mathfrak{f}}(\frac{1}{q} \Sigma_{i \in [n]} \frac{p_i q}{q_i}) \leq O(\log n) + O(\log(n 2^{O(\log n)} 2^{O(\log n)})) = O(\log n)$$

□

Theorem 3.2. *The following per-dimension aggregations $agg : \left(\!\!\binom{\mathbb{Q}^d}{*}\!\!\right) \to \mathbb{Q}^{d'}$ are logarithmic:*

1. *$agg(M) \coloneqq p_2(\Sigma_{x \in T} p_1(x)), T \subseteq M$, for rational polynomials $p_1, p_2 : \mathbb{Q} \to \mathbb{Q}$.*
2. *$agg(M) \coloneqq (agg_1, \ldots, agg_a), \gamma(agg_i), a \in \mathbb{N}$. That is, the concatenation of a fixed number of logarithmic aggregations.*

Proof. As these aggregations operate on vectors per-dimension, it is enough to show that they are logarithmic when operating on multisets of scalars.

1. Let $p_1 : \mathbb{Q} \to \mathbb{Q}$ be a rational polynomial and let c_1 be the constant factor upper bound guaranteed for it by Lemma A.1. Let $f, g : \mathbb{N} \to \mathbb{N}$ such that $f(n), g(n) = O(\log n)$ and let c_2 such that $\forall n\ f(n) \leq c_2 \log n, g(n) \leq c_2 \log n$. Let c_3 be the constant factor upper bound guaranteed by Lemma A.2 for the summation of elements whose bit-length is $\leq c_1 c_2 \log n$. Finally, let $p_2 : \mathbb{Q} \to \mathbb{Q}$ be a rational polynomial and let c_4 be the constant factor upper bound guaranteed for it by Lemma A.1 Let $M \in \left(\!\!\binom{\mathbb{Q}}{n}\!\!\right),\ M = \{\{x_1, \ldots, x_n\}\},\ \lambda_{\mathfrak{f}}(x_i) \leq f(n), \lambda_{\mathfrak{f}}(\mathrm{cd}(M)) \leq g(n)$ be a multiset of elements of bit-length at most $f(n)$ per-element, and a common denominator of bit-length at most $g(n)$. Then, by Lemma A.1 we have that $\forall i\ \lambda_{\mathfrak{f}}(p_1(x_i)) \leq c_1 f(n)$. Hence, by Lemma A.2 we have that $\lambda_{\mathfrak{f}}(\Sigma_{i \in [n]} p_1(x_i)) \leq c_3 c_1 c_2 \log n$.

Hence, by Lemma A.1 we have that $\lambda_{\mathfrak{f}}(p_2(\Sigma_{i\in[n]}p_1(x_i))) \leq c_4c_3c_1c_2\log n$. Hence, $S_{\text{agg}} = O(\log n)$.

For $S^{\text{cd}}_{\text{agg}}$, define $q_{\text{cd}} \coloneqq \text{cd}(M)$, assume $p_1(x) = \Sigma_{i=0}^{|p_1|}\frac{a_i}{b_i}x^i, p_2(x) = \Sigma_{i=0}^{|p_2|}\frac{r_i}{s_i}x^i$, define $q_{p_1} \coloneqq q_{cd}^{|p_1|}\prod_{i=0}^{|p_1|}b_i$, and define $q \coloneqq q_{p_1}^{|p_2|}\prod_{i=0}^{|p_2|}s_i$. Observe that:

1. Let $\frac{y}{z} \in \mathbb{Q}$ such that $\frac{q_{\text{cd}}}{z} \in \mathbb{N}$, then there exists $y' \in \mathbb{Z}$ such that $p_1(\frac{y}{z}) = \frac{y'}{q_{p_1}}$.
2. Let $\frac{y}{z} \in \mathbb{Q}$ such that $\frac{q_{p_1}}{z} \in \mathbb{N}$, then there exists $y' \in \mathbb{Z}$ such that $p_2(\frac{y}{z}) = \frac{y'}{q}$.

By (1) we have that $\forall T \subseteq M \quad \frac{\Sigma_{x\in T}p_1(x)}{q_{p_1}} \in \mathbb{Z}$. Then, by (2) we have that $\forall T \subseteq M \quad \frac{p_2(\Sigma_{x\in T}p_1(x))}{q} \in \mathbb{Z}$. Hence, we have that $\forall U \subseteq \left(\!\!\binom{M}{*}\!\!\right)$ $\text{cd}(\{\{p_2(\Sigma_{x\in T}p_1(x))\}\}_{T\in U}) \leq q$. Hence, $\lambda_{\mathfrak{f}}(\text{cd}(\{\{p_2(\Sigma_{x\in T}p_1(x))\}\}_{T\in U})) \leq \lambda_{\mathfrak{f}}(q) = O(\log(q_{p_1}^{|p_2|}\prod_{i=0}^{|p_2|}s_i) = O(\log(q_{p_1}^{|p_2|})) = O(\log(q_{cd}^{|p_1|}\prod_{i=0}^{|p_1|}b_i)) = O(\log(q_{cd}^{|p_1|})) = O(|p_1|\log(q_{cd})) = O(\log n)$. Hence, $S^{\text{cd}}_{\text{agg}} = O(\log n)$ [2.] $\lambda_{\mathfrak{f}}(\text{agg}) = \Sigma_{i\in[a]}\lambda_{\mathfrak{f}}(\text{agg}_i) = aO(\log n) = O(\log n)$. As for $S^{\text{cd}}_{\text{agg}}$, Let $U \subseteq \left(\!\!\binom{M}{*}\!\!\right)$, define $\text{cd}_i \coloneqq \text{cd}(\{\{\text{agg}_i(T)\}\}_{T\in U})$, and define $\text{cd}_{\text{agg}} \coloneqq \text{cd}(\{\{\text{agg}(T)\}\}_{T\in U}))$. Then $\text{cd}_{\text{agg}} \leq \prod_{i\in[a]}\text{cd}_i$, hence $\lambda_{\mathfrak{f}}(\text{cd}_{\text{agg}}) \leq \lambda_{\mathfrak{f}}(\prod_{i\in[a]}\text{cd}_i) = O(\log(\prod_{i\in[a]}\text{cd}_i)) = O(aO(\log n)) = O(\log n)$.

□

Example 3.3. *The following common aggregations are logarithmic:*

1. *sum.*
2. *Selection of k elements, for a fixed $k \in \mathbb{N}$, by any criteria e.g. highest; lowest; quintile.*
3. *k-bins agg bin-aggregation, for a fixed $k \in \mathbb{N}$ and logarithmic aggregation agg.*

Proof.

1. By Theorem 3.2(1), setting $T = M,\ p_1(x) = x,\ p_2(x) = x$, we have that sum is logarithmic.
2. Defining agg_i to be the selection of the $i^{\text{th}}, i \in [k]$ element (by whichever criteria) we have that agg_i is logarithmic. Then, by Theorem 3.2(2) we have that $(\text{agg}_1, \ldots, \text{agg}_k)$ is logarithmic i.e. the selection of the k elements is logarithmic.
3. Note that agg being logarithmic when applied to M it is logarithmic when applied to any $T \subseteq M$. Defining $T_i, i \in [k]$ to be the elements in bin k, and defining $\text{agg}_i \coloneqq \text{agg}(T_i)$ we have that agg_i is logarithmic and by Theorem 3.2(2) $(\text{agg}_1, \ldots, \text{agg}_k)$ is logarithmic i.e. the sequence of aggregated bins is logarithmic.

□

Lemma 3.5. *Let F be an MLP of input dimension d, we define $S_F : \mathbb{N} \to \mathbb{N}$ be the output-size complexity of F, that is,*

$$S_F(k) \coloneqq \max(\lambda_{\mathfrak{f}}(F(x)) : x \in \mathbb{Q}^d, \lambda_{\mathfrak{f}}(x) = k)$$

In addition, we denote by $S^{\text{cd}}_{agg} : \mathbb{N}^3 \to \mathbb{N}$ the complexity of the common denominator of n applications of F on elements of a multiset of n vectors, bit-length k per vector, and multiset-common-denominator of bit-length ℓ, that is,

$$S^{\text{cd}}_F(n,k,\ell) \coloneqq \max\left(\lambda_{\mathfrak{f}}(\text{cd}(M)) : M = \{\{F(x_i)\}\}_{i\in[m]}, x_i \in M', M' \in \left(\!\!\binom{\mathbb{Q}_k^d}{n}\!\!\right), \lambda_{\mathfrak{f}}(\text{cd}(M')) \leq \ell\right)$$

Then,

1. $S_F(k) = O(k)$.
2. *For every* $f : \mathbb{N} \to \mathbb{N}$ *such that* $f(n) = O(\log n)$ *it holds that* $S_F^{\mathrm{cd}}(n, k, f(n)) = O(\log n)$.

Proof. 1. Assume $F = (l_1, \ldots, l_m), l_i = (w_i, b_i)$, $\dim(w_i) = (d_{i+1}, d_i)$. Let S_{F_l} to be the output-size complexity of a single layer l, we show that $S_{F_l}(n) = O(n)$. As it is straightforward that the output-size complexity of a composition of a fixed number of functions of linear output-size complexity is linear, proving $\forall\ l\ S_{F_l}(n) = O(n)$ will prove $S_F(n) = O(n)$.

Assume w.l.o.g that $l = l_1$. All fractions in the proceeding argumentation are in reduced form. For $i \in [d_2], j \in [d_1]$ let $\frac{p_{i,j}^{(w)}}{q_{i,j}^{(w)}} = w_1(i, j)$, and for $i \in [d_2]$ let $\frac{p_i^{(b)}}{q_i^{(b)}} = b_1(i)$. Define

$$q_c^{(w)} \coloneqq \prod_{i \in [d_2], j \in [d_1]} q_{i,j}^{(w)} \prod_{i \in [d_2]} q_i^{(b)}$$

a common denominator of all the parameters. For $i \in [d_1]$ let $\frac{p_i^{(x)}}{q_i^{(x)}} = x(i)$ and define $q_c^{(x)} \coloneqq \prod_{i \in [d_1]} q_i^{(x)}$ a common denominator of the input values. Let $|(w_1 x)(i) + b_1(i)| = \frac{p_i^{(r)}}{q_i^{(r)}}$. Remember that for $n \in \mathbb{N}$ $\lambda_{\mathfrak{f}}(n) = \log n$, then

$$\lambda_{\mathfrak{f}}(q_i^{(r)}) \leq \lambda_{\mathfrak{f}}(q_c^{(w)} \cdot q_c^{(x)}) = O(\lambda_{\mathfrak{f}}(q_c^{(w)}) + \lambda_{\mathfrak{f}}(q_c^{(x)})) = O(\lambda_{\mathfrak{f}}(q_c^{(w)}) + \Sigma_{i \in [d_1]} \lambda_{\mathfrak{f}}(q_i^{(x)})) = O(\lambda_{\mathfrak{f}}(x)) \quad \text{(A.1)}$$

Define

$$p_{\max}^{(w)} \coloneqq \max(\left|p_{i,j}^{(w)}\right| : i \in [d_2], j \in [d_1])$$

, then

$$\begin{aligned}\lambda_{\mathfrak{f}}(p_i^{(r)}) \leq \lambda_{\mathfrak{f}} q_i^{(r)}(p_i^{(b)} + \Sigma_{j \in [d_1]} p_{i,j}^{(w)} p_j^{(x)}) = \\ O(\lambda_{\mathfrak{f}} q_i^{(r)} + \lambda_{\mathfrak{f}}(\left|p_i^{(b)}\right| + p_{\max}^{(w)} \Sigma_{j \in [d_1]} \left|p_j^{(x)}\right|)) = \\ O(\lambda_{\mathfrak{f}}(p_{\max}^{(w)} \Sigma_{j \in [d_1]} \left|p_j^{(x)}\right|)) = \\ O(\lambda_{\mathfrak{f}}(p_{\max}^{(w)}) + \lambda_{\mathfrak{f}}(\Sigma_{j \in [d_1]} \left|p_j^{(x)}\right|)) = O(\lambda_{\mathfrak{f}}(x)) \quad \text{(A.2)}\end{aligned}$$

For fractional representation we have

$$\lambda_{\mathfrak{f}}(w_1 x + b_1) = O(\Sigma_{i \in [d_2]}(\lambda_{\mathfrak{f}}(p_i^{(r)}) + \lambda_{\mathfrak{f}}(q_i^{(r)})))$$

Hence, by Equations (A.1) and (A.2) we have

$$\lambda_{\mathfrak{f}}(w_1 x + b_1) = O(\lambda_{\mathfrak{f}}(x))$$

, concluding the induction proof.

2. Define $q_c^F \coloneqq \mathrm{cd}(\{\{w_i, b_i : i \in [m]\}\})$ the common denominator of all the parameters in F and note that in relation to the input bit-length $q_c^F = O(1)$. Let

$$M = \{\{x_1, \ldots, x_m\}\}, M \in \left(\!\!\binom{\mathbb{Q}_k^d}{n}\!\!\right)$$

such that $\lambda_{\mathfrak{f}}(\mathrm{cd}(M)) = k$, then $q_c^{(F)} \cdot \mathrm{cd}(M) = O(k)$. Hence $S_F^{\mathrm{cd}}(k) = O(k)$. $\square$