Lost in Aggregation

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

Anonymous authors
Paper under double-blind review

Abstract

We define a generic class of functions that captures most conceivable aggregations for Message-Passing Graph Neural Networks (MP-GNNs), and prove that any 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 doubly-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 results state that MP-GNNs match full CR, however they concern a weak, 'non-uniform', notion of distinguishing-power where each graph size may required 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*, i.e. approximate, a function is to have the function's distinguishing-power i.e. to output different values for every two inputs on which the function differs. 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 is less relevant 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 straghitforward:

1. MP-GNNs with m layers do not match the distinguishing-power of CR for graphs of diameter $> m$.
2. A trivial sum-aggregation MP-GNN matches $\mathrm{CR}^{(1)}$ for diameter-1 graphs.
3. With no restriction on the aggregation function other than being computable and having a fixed output dimension, MP-GNNs with m layers can match the distinguishing-power of CR for graphs of diameter $\leq m$ 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.

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.

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 describe a general class of aggregation functions (Definition 3.1) which contains most conceivable aggregations, and 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 the whole graphs, by $N_{\text{dp}}(n)$ and $N_{\text{gdp}}(n)$ respectively, and similarly for $\text{CR}^{(2)}$ by $\text{CR}^{(2)}_{\text{dp}}(n)$, $\text{CR}^{(2)}_{\text{gdp}}(n)$, 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 3.7). Formally,

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

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

2. Observing a lower-bound for $\text{CR}^{(2)}_{\text{dp}}$, we add that not only the distinguishing-power of $\mathcal{N}$ is weaker, i.e. there are vertices distinguishable $\text{CR}^{(2)}$ and not by $\mathcal{N}$, but it gets infinitely weaker as the graph size grows (Corollary 3.9). Formally,

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

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 3.6).

2 Preliminaries

By $\mathbb{N}; \mathbb{Q}$ we denote the natural and rational numbers respectively. For $m \in \mathbb{N}$ we define $[m] \coloneqq \{i : i \in \mathbb{N}, 1 \leq i \leq m\}$. For $q \in \mathbb{Q}$ we denote its bit-length by $\lambda(q)$. For a vector $v \in \mathbb{Q}^d$ we define $\lambda(v) \coloneqq \Sigma_{i\in[d]}\lambda(v(i))$ the bit-length of v. For $d, k \in \mathbb{N}$ we define $\mathbb{Q}^d_k \coloneqq \{q : q \in \mathbb{Q}^d, \lambda(q) \leq k\}$ the

[1]Note that when referring to $\text{CR}^{(t)}$ as an upper bound - implying inclusion - we refer to the domain of graphs with diameter no greater than t: Obviously, when considering graphs of larger diameters, there are nodes distinguishable by a trivial MP-GNN with $t+1$ layers that are not distinguishable by $\text{CR}^{(t)}$.

dimension-d rational vectors of bit-length no greater than k. 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 $\left(\!\binom{S}{*}\!\right)$. 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 $\dim(W) \coloneqq (d_1, d_2)$.

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 define the *diameter* of a graph G, denoted $\text{dia}(G)$, to be the longest simple path between any two of its vertices. 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 the domain of graphs with trivial input-features, and denote it by $\mathcal{G}$, that is,

$$\mathcal{G} \coloneqq \{G \mid \forall v \in V(G)\ Z(G)(v) = 1\}$$

For $k, n \in \mathbb{N}$ we define $\mathcal{G}(n) \coloneqq \{G \in \mathcal{G} : |G| = n\}$ the graphs in $\mathcal{G}$ of size n, $\mathcal{G}^{(k)} \coloneqq \{G \in \mathcal{G} : \text{dia}(G) = k\}$ the graphs in $\mathcal{G}$ of diameter k, and $\mathcal{G}^{(k)}(n) \coloneqq \mathcal{G}(n) \cap \mathcal{G}^{(k)}$ those of size n and diameter k. 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 $\mathcal{G}_S$ be a graph domain, a mapping $f : \mathcal{G} \to \mathcal{Z}_*$ to new feature maps is called a *feature transformation*, and for $q \in \mathbb{N}$ a mapping $f : \mathcal{G}_S \to \mathbb{Q}^d$ to d-tuples 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 $f_F : \mathbb{Q}^{d_{in}} \to \mathbb{Q}^{d_{out}}$, denoted also by $F(x)$, such that

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

Message-Passing Graph Neural Network

A *Message Passing Graph Neural Network* (MP-GNN) of depth m and dimensions $r_0, \{p_i, q_i, r_i\}_{i \in [m]}$ $N = (\mathsf{mlp}_1, \text{agg}_1, \text{msg}_1), \ldots, (\mathsf{mlp}_m, \text{agg}_m, \text{msg}_m)$ is a sequence of m triplets, referred to as *layers*, such that for $i \in [m]$ layer i comprises a message function; an aggregation function; and an MLP,

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

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 dimension-wise 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,

$$\mathsf{cr}_G^{(0)}(v) \coloneqq Z(G)(v)$$

, then, for all $t > 0$ we define

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

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

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

3 Limited by Aggregation

We start with defining the aggregation classes that are our main focus.

Definition 3.1 (Sublinear; Logarithmic, Aggregation). *Let $d, d' \in \mathbb{N}$, $agg : \left(\!\!\binom{\mathbb{Q}^d}{*}\!\!\right) \to \mathbb{Q}^{d'}$ be a function from a multiset of rational vectors to a single rational vector. We denote by $S_{agg} : \mathbb{N}^2 \to \mathbb{N}$ the output complexity of agg depending on the number of vectors n and maximum bit-length k of any vector,*

$$S_{agg}(n,k) \coloneqq \max(\lambda(agg(M)) : M \in \left(\!\!\binom{\mathbb{Q}_k^d}{n}\!\!\right))$$

We say that agg is sublinear*, notated $\sigma(agg)$, if and only if for every $f : \mathbb{N} \to \mathbb{N}$ such that $f(n) = o(n)$ it holds that $S_{agg}(n, f(n)) = o(n)$.*

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

To our knowledge, the only architectures used in practice with potentially non-logarithmic aggregation are Graph Attention Networks Veličković et al. (2017) and Graph Transformers Dwivedi & Bresson (2020) which use *softmax* aggregation - involving exponentiation by the input. Most conceivable aggregation functions are logarithmic, see the following example. The reason for having also the sublinear definition is to provide a meaningful distinguishing-power bound for an even more general class of aggregation functions.

Example 3.2.
The functions sum; avg; max *are logarithmic.*

Proof. As these aggregations operate on vectors per-dimension, it is enough to show that they are logarithmic when operating on multisets of scalars.
Let $k : \mathbb{N} \to \mathbb{N}$ be a function such that $k(n) = O(\log n)$, and let $M = \{\{q_1, \ldots, q_n\}\}, \lambda(q_i) \leq k(n)$ be a multiset of n rationals, each of bit-length at most $k(n)$.

For sum aggregation, by definition

$$\lambda(\Sigma_{i\in[n]}q_i) \leq \log n2^{k(n)} = (\log n) + k(n)$$

Hence, by assumption on $k(n)$ it holds that

$$\max(\lambda(\Sigma_{i\in[n]}x_i) : \{\{x_i\}\}_{i\in[n]} \in \left(\!\!\binom{\mathbb{Q}}{k(n)}\!\!\right)) = O(\log n)$$

By similar argumentation we have that avg aggregation is also logarithmic i.e.

$$\max(\lambda(\frac{1}{n}\Sigma_{i\in[n]}x_i) : \{\{x_i\}\}_{i\in[n]} \in \left(\!\!\binom{\mathbb{Q}}{k(n)}\!\!\right)) = O(\log n)$$

Finally, for max aggregation, by definition

$$\lambda(\max_{i\in[n]} q_i) \leq \max_{i\in[n]}(\lambda(q_i)) \leq k(n)$$

Hence, by assumption on $k(n)$ it holds that

$$\max(\lambda(\max_{i\in[n]} x_i) : \{\{x_i\}\}_{i\in[n]} \in \left(\!\!\binom{\mathbb{Q}}{k(n)}\!\!\right)) = O(\log n)$$

□

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

Distinguishing-Power.

Let $\mathcal{G}' \subseteq \mathcal{G}$ be a graph domain, and let N be an MP-GNN, we define the *distinguishing-power* of N on $\mathcal{G}'$,

$$N_{\mathrm{dp}}(\mathcal{G}') := |\{N(G, v) : G \in \mathcal{G}', v \in V(G)\}|$$

to be the number of vertices equivalence-classes that N induces on $\mathcal{G}'$. Similarly, for $\mathrm{CR}^{(2)}$ we define

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

For distinguishing between graphs, we define

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

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

Main Result

Our fundamental result (Theorem 3.7) is that for any MP-GNN N comprising (only) logarithmic aggregations, the distinguishing-power of N is polynomial i.e.

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

, and even if the aggregations are only sublinear then the distinguishing-power is still sub-exponential i.e.

$$N_{\text{dp}}(\mathcal{G}(n)) = 2^{o(n)},\ N_{\text{gdp}}(\mathcal{G}(n)) = 2^{o(n)}$$

For simplicity, several of our definitions and statement to follow concern distinguishing between vertices, however it is relatively straightforward to adapt them to the setting of distinguishing between graphs.

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 $N_{\text{dp}}(\mathcal{G}(n)), N_{\text{gdp}}(\mathcal{G}(n)) \le 2^{L_N(n)}$. (Lemma 3.3)
2. We prove that for logarithmic-aggregations and sublinear-aggregations MP-GNNs, it holds that $L_N(n) = O(\log n)$ and $L_N(n) = o(n)$, respectively. (Lemma 3.5)
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 $N = (\mathsf{mlp}_1, \text{agg}_1, \text{msg}_1), \ldots, (\mathsf{mlp}_m, \text{agg}_m, \text{msg}_m)$ be an MP-GNN. For a graph $G \in \mathcal{G}$ and a vertex $v \in V(G)$ we denote by $I_N(G, v)$ the sequence of aggregation bits produced by the aggregations operating on G, v, that is,

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

We define the complexity of I_N, $L_N : \mathbb{N} \to \mathbb{N}$, to be the maximum bit-length of I_N per graph size, that is,

$$L_N(n) \coloneqq \max\left(\lambda(I_N(G, v)) : G \in \mathcal{G},\ |G| = n, v \in V(G)\right)$$

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

Lemma 3.3. *Let N be an MP-GNN, then $N_{\text{dp}}(\mathcal{G}(n)) \le 2^{L_N(n)}$.*

Proof. It is relatively straightforward, by induction, that

$$\forall G, G' \in \mathcal{G}\ \forall v \in V(G)\ \forall v' \in V(G')\quad I_N(G, v) = I_N(G', v') \Rightarrow N(G, v) = N(G, v')$$

, hence, the distinguishing-power of N is upper-bounded by the number of possible values of I_N, which in turn is upper-bounded exponentially by the maximum bit-length of I_N. □

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. We first show the output-complexity of a single MLP, before considering the complete MP-GNN's computation.

Lemma 3.4. *Let F be an MLP of input dimension d, and let $S_F : \mathbb{N} \to \mathbb{N}$ be the output-size copmlexity of F, that is, $S_F(n) \coloneqq \max(\lambda(F(x)) : x \in \mathbb{Q}^d, \lambda(x) = n)$, then $S_F(n) = O(n)$.*

Proof. 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} be the output-size complexity of a single layer l, we show that $S_{F_l}(n) = O(n)$. Assume w.l.o.g that $l = 1$. Let $x \in \mathbb{Q}^{d_1}$ be an input, define

$$w_{1_{max}} \coloneqq \max(\lambda(w_1(i,j)) : i \in [d_2], j \in [d_1])$$

the maximum bit-length of any weight, and

$$x_{max} \coloneqq \max(\lambda(x(j)) : j \in [d_1])$$

the maximum bit-length of any input element. Then,

$$\lambda(F(x)) = \lambda(\mathrm{ReLU}((w_1 \cdot x) + b_1)) \leq \lambda(b_1) + \Sigma_{i \in [d_2]} \lambda(\Sigma_{j \in [d_1]} w_1(i,j) x_j) \leq$$

$$\lambda(b_1) + d_2 d_1 \max_{i \in [d_2], j \in [d_1]} (\lambda(w_1(i,j) x_j)) = \lambda(b_1) + d_2 d_1 \max_{i \in [d_2], j \in [d_1]} (\lceil |\log(w_1(i,j) x_j)| \rceil) =$$

$$\lambda(b_1) + d_2 d_1 \max_{i \in [d_2], j \in [d_1]} (\lceil |\log w_1(i,j) + \log x_j| \rceil) \leq \lambda(b_1) + d_2 d_1 \max_{i \in [d_2], j \in [d_1]} (\lceil |\log w_1(i,j)| \rceil + \lceil |\log x_j| \rceil) \leq$$

$$\lambda(b_1) + d_2 d_1 (w_{1_{max}} + x_{max}) \leq \lambda(b_1) + d_2 d_1 (w_{1_{max}} + \Sigma_{i \in [d_1]} \lambda(x_i)) = \lambda(b_1) + d_2 d_1 w_{1_{max}} + d_2 d_1 \lambda(x) \Rightarrow$$

$$S_{F_1}(n) = O(n)$$

It is straightforward that the output-size complexity of a composition of a fixed number of functions of linear output-size complexity is linear, hence $S_F(n) = O(n)$. ☐

Lemma 3.5. *Let $N = (\mathsf{mlp}_1, agg_1, \mathrm{msg}_1), \ldots, (\mathsf{mlp}_m, agg_m, \mathrm{msg}_m)$ be an MP-GNN, then:*

1. *If all the aggregations are logarithmic then the complexity is logarithmic, formally*

$$\forall i \; \gamma(agg_i) \Rightarrow L_N(n) = O(\log n)$$

2. *Similarly, if all the aggregations are sublinear then the total-information complexity of N is sublinear, formally*

$$\forall i \; \sigma(agg_i) \Rightarrow L_N(n) = o(n)$$

Proof. We prove item (1), the proof of (2) is almost identical. 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 after msg_l, the output after agg_l, and the output of the layer i.e. after mlp_l. Formally,

$$S^{\mathrm{msg}}_{N^{(l)}}(n) \coloneqq \max\left(\lambda(\mathrm{msg}_l(N^{(l-1)}(G,v), N^{(l-1)}(G,w))) : G \in \mathcal{G},\ |G| = n,\ v \in V(G),\ w \in N_G(v)\right)$$

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

$$S_{N^{(l)}}(n) \coloneqq \max\left(\lambda(N^{(l)}(G,v)) : G \in \mathcal{G},\ |G| = n,\ v \in V(G)\right)$$

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

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

and the complexity of a sum of a fixed number of $O(\log n)$-complexity functions is $O(\log n)$, the induction's statement implies the lemma. For $l = 1$, by the initial features all being 1, clearly

$$\exists c \in \mathbb{N} : S^{\text{msg}}_{N^{(1)}}(n) = c$$

, hence $S^{\text{msg}}_{N^{(1)}}(n) = O(\log n)$ in a trivial manner. By the assumption on agg_1 we have $S^{\text{agg}}_{N^{(1)}}(n) = O(\log n)$. Finally, by Lemma 3.4 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 $S_{N^{(1)}}(n) = O(\log n)$. Assuming correctness for $l = k < m$ we prove for $l = k + 1$. By Lemma 3.4 and by the induction assumption, $S^{\text{msg}}_{N^{(k+1)}}(n)$ is the complexity of a composition of an $O(n)$-complexity function over the concatenation of two $O(\log n)$-complexity functions, which is $O(\log n)$. As for $S^{\text{agg}}_{N^{(k+1)}}$ and $S_{N^{(k+1)}}$, the argumentation is similar to that in the case of $l = 1$. □

Remark 3.6. *The line of proof of Lemma 3.5 works for every message and combination functions with output-size complexity $O(n)$ (with n being the function's input size) - not only for ReLU-activated MLPs. Hence, the guarantee that $L_N(n) = o(n)$ (with n being the input-graph size), and subsequently Theorem 3.7, hold for all MP-GNNs architectures that consist of such functions.*

Theorem 3.7. *Let $N = (\mathsf{mlp}_1, agg_1, \text{msg}_1), \ldots, (\mathsf{mlp}_m, agg_m, \text{msg}_m)$ be an MP-GNN, then:*

1. *If all aggregations are logarithmic then the distinguishing-power of N is polynomial. Formally,*

$$\forall i \; \gamma(agg_i) \Rightarrow N_{\text{dp}}(\mathcal{G}(n)) = \text{poly}(n)$$

2. *If all aggregations are sublinear then the distinguishing-power of N is sub-exponential. Formally,*

$$\forall i \; \sigma(agg_i) \Rightarrow N_{\text{dp}}(\mathcal{G}(n)) = 2^{o(n)}$$

3. *When N includes a graph readout, i.e. $N = (\mathsf{mlp}_i, agg_i, \text{msg}_i)_{i \in [m]}, (\mathsf{mlp}_R, agg_R)$, the above hold also for distinguishing graphs, that is,*

$$\forall i \; \gamma(agg_i) \wedge \gamma(agg_R) \Rightarrow N_{\text{gdp}}(\mathcal{G}(n)) = \text{poly}(n), \quad \forall i \; \sigma(agg_i) \wedge \sigma(agg_R) \Rightarrow N_{\text{gdp}}(\mathcal{G}(n)) = 2^{o(n)}$$

Proof.

1. By assumption, Lemma 3.5, and Lemma 3.3.
2. By assumption, Lemma 3.5, and Lemma 3.3.
3. The definition of $L_N(n)$; Lemma 3.3; and Lemma 3.5, can be adapted to include the output-bits of the readout aggregation, agg_R, and refer to graph embeddings rather than feature transformations. All relatively straightforward.

□

Comparison to Color Refinement

The absolute-terms upper bounds in Theorem 3.7 are meaningful on their own, considering that the total number of non-isomorphic graphs of size n, $|\mathcal{G}(n)|$, is doubly-exponential. In previous studies, the distinguishing-power of MP-GNNs has been compared to the distinguishing-power of 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 3.7 in its perspective. To be precise, we compare the distinguishing-power of (already) sublinear-aggregations MP-GNNs to the distinguishing-power of merely two iterations of CR, on diameter-2 graphs, i.e. to $\text{CR}^{(2)}_{\text{dp}}(\mathcal{G}^{(2)})$. We start with the following observation.

Lemma 3.8. *Let $n \in \mathbb{N}$, then*

$$CR^{(2)}_{\text{dp}}(\mathcal{G}^{(2)}(2n+1)) \geq \binom{n+n-1}{n-1}, \quad CR^{(2)}_{\text{gdp}}(\mathcal{G}^{(2)}(2n+1)) \geq \binom{n+n-1}{n-1}$$

Proof. We look at a two-level star graph of size $2n+1$, 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

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

all the possible choices, with repetition, of n elements from $n+1$ types. For $K \in \mathcal{K}_n$ we define G_K 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). Formally, $G_K \in \mathcal{G}$ is defined as follows:

$$V(G) = \{v\} \bigcup \{u_1, \dots, u_n\} \bigcup \{w_1, \dots, w_n\}$$

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

Finally, for $n \in \mathbb{N}$ we define

$$\mathcal{G}_{\mathcal{K}}(2n+1) := \{G_K : K \in \mathcal{K}_n,\ n \in \mathbb{N}\} \subset \mathcal{G}$$

the set of all graphs of the form above, of size $2n+1$. Observe that

$$|\mathcal{G}_{\mathcal{K}}(2n+1)| = \binom{n+n-1}{n-1}$$

, 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. Also, it is relatively straightforward that

$$\forall G, G' \in \mathcal{G}_{\mathcal{K}}(2n+1)\ \mathsf{cr}^{(2)}_G(v) \neq \mathsf{cr}^{(2)}_{G'}(v), \quad \forall G, G' \in \mathcal{G}_{\mathcal{K}}(2n+1)\ \mathsf{cr}^{(2)}(G) \neq \mathsf{cr}^{(2)}(G')$$

□

Combined with Lemma 3.3, we arrive at the following sufficient condition for an MP-GNN having weaker distinguishing-power than $\text{CR}^{(2)}_{\text{dp}}(\mathcal{G}^{(2)})$, and combined with Theorem 3.7 we have the following measure of the gap between sublinear-aggregations MP-GNNs and $\text{CR}^{(2)}_{\text{dp}}(\mathcal{G}^{(2)})$.

Corollary 3.9. *Let $N = (\mathsf{mlp}_1, agg_1, \text{msg}_1), \dots, (\mathsf{mlp}_m, agg_m, \text{msg}_m)$ be an MP-GNN, then:*

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

$$\exists n \in \mathbb{N} : L_N(2n+1) < \left\lceil \log \binom{2n-1}{n-1} \right\rceil \Rightarrow \exists G, G' \in \mathcal{G}^{(2)},\ v \in V(G),\ v' \in V(G') :$$

$$\mathsf{cr}^{(2)}_G(v) \neq \mathsf{cr}^{(2)}_{G'}(v') \wedge N(G, v) = N(G', v')$$

2. *Moreover, if $\forall i\ \sigma(agg_i)$ then the distinguishing-power of N is relatively infinitely weaker than that of $CR^{(2)}$. Formally,*

$$\forall i\ \sigma(agg_i) \Rightarrow \lim_{n \to \infty} \frac{N_{\text{dp}}(\mathcal{G}(n))}{CR^{(2)}_{\text{dp}}(\mathcal{G}^{(2)}(n))} = 0$$

3. *When N includes a graph readout, i.e.* $N = (\mathsf{mlp}_i, agg_i, \mathrm{msg}_i)_{i\in[m]}, (\mathsf{mlp}_R, agg_R)$*, the above hold also for distinguishing graphs, that is:*

   a. *Let LG_N be an adaptation of L_N that includes the bits of agg_R, then*

$$\exists n \in \mathbb{N} : LG_N(2n+1) < \left\lceil \log \binom{2n-1}{n-1} \right\rceil \Rightarrow$$

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

   b.

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

Remark 3.10. *In items (2);(3) of Corollary 3.9 we combined the result for $N_{\mathrm{dp}}(\mathcal{G}(n))$ and observation for $CR^{(2)}_{\mathrm{dp}}(\mathcal{G}^{(2)}(n))$ (and same for graph-embeddings) directly, that is, we compared the distinguishing-powers on different graph domains. This is not weaker than our general statement but, for what it's worth, stronger, as it is straightforward by definition that $N_{\mathrm{dp}}(\mathcal{G}(n)) \geq N_{\mathrm{dp}}(\mathcal{G}^{(2)}(n))$ and $CR^{(2)}_{\mathrm{dp}}(\mathcal{G}(n)) \geq CR^{(2)}_{\mathrm{dp}}(\mathcal{G}^{(2)}(n))$.*

4 Concluding Remarks

We have introduced an output-size complexity property for aggregation functions, satisfied by most conceivable aggregations, and shown that it has the effect of restricting MP-GNN models to distinguish merely a polynomial number of equivalence classes, or a sub-exponential number in case of an even more general property, both when distinguishing between vertices and when distinguishing between graphs. Given that the number of non-equivariant vertices or non-isomorphic graphs is doubly-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 even our more general class of MP-GNNs, making $\mathrm{CR}^{(1)}$ a tight bound [2], 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 unknown target function assumes 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. When the target function assumes an exponential number of values, it is impossible already for sublinear-aggregations MP-GNNs to make the required distinguishments.

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-size complexity $O(n)$.

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

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 or sublinear, 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*. There again, an exponentiation by the input is involved, potentially leading to an exponential rather than linear output-size complexity of the MLP, yet further analysis is required for a clear characterization.

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.

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.