How hard is graph isomorphism for graph neural networks?

Andreas Loukas
École Polytechnique Fédérale Lausanne
andreas.loukas@epfl.ch

Abstract

A hallmark of graph neural networks is their ability to distinguish the isomorphism class of their inputs. This study derives the first hardness results for graph isomorphism in the message-passing model (MPNN). MPNN encompasses the majority of graph neural networks used today and is universal in the limit when nodes are given unique features. The analysis relies on the introduced measure of *communication capacity*. Capacity measures how much information the nodes of a network can exchange during the forward pass and depends on the depth, message-size, global state, and width of the architecture. It is shown that the capacity of MPNN needs to grow *linearly* with the number of nodes so that a network can distinguish *trees* and *quadratically* for general *connected graphs*. Crucially, the derived bounds are applicable not only to worst-case instances but over a portion of all inputs. An empirical study involving 12 tasks of varying difficulty and 420 networks reveals strong alignment between actual performance and theoretical predictions.

1 Introduction

The study of the expressive power of neural networks has historically focused on vectors and sequences [Cybenko, 1989, Hornik et al., 1989, Mhaskar and Poggio, 2016, Hanin and Sellke, 2017, Lin et al., 2017, Lu et al., 2017, Neto et al., 1997, Pérez et al., 2019].

Recently, we have also seen the first theoretical investigations of graph neural networks [Maron et al., 2019, Keriven and Peyré, 2019, Xu et al., 2018]. Therein, one of the most intensely studied models is that of message-passing neural networks (MPNN). Since its inception by Scarselli et al. [2008], MPNN has been extended to include edge [Gilmer et al., 2017] and global features [Battaglia et al., 2018]. The model also encompasses many of the popular graph neural network architectures used today [Kipf and Welling, 2016, Xu et al., 2018, Hamilton et al., 2017, Li et al., 2015, Duvenaud et al., 2015, Battaglia et al., 2016, Kearnes et al., 2016, Simonovsky and Komodakis, 2017].

1.1 Prior work

Two types of analyses of MPNN may be distinguished.

The first bound the expressive power of anonymous MPNN, i.e., those in which nodes do not have any access to discriminative features (also known as labels or attributes) and that are permutation (in)equivariant by design. Xu et al. [2018] and Morris et al. [2019] established the equivalence of anonymous MPNN to the 1st-order Weisfeiler-Lehman graph isomorphism test. A consequence of this connection is that anonymous MPNN cannot distinguish between regular graphs with the same number of nodes. Other notable findings include the derivation of approximability results for NP-hard problems [Sato et al., 2019], the connection to 1st-order logic [Barceló et al., 2019], the observation that MPNN cannot count simple subgraphs [Chen et al., 2020], as well as the analysis of the power of

particular architectures to compute graph properties [Dehmamy et al., 2019, Garg et al., 2020] and to distinguish graphons [Magner et al., 2020]—see also overview papers [Geerts et al., 2020, Sato, 2020].

The second focus on the non-anonymous case [Murphy et al., 2019, Loukas, 2020, Dasoulas et al., 2019, Sato et al., 2020]. With node features acting as identifiers, MPNN were shown to become universal in the limit [Loukas, 2020] and permutation (in)equivariance needs to be learned. The node features may correspond to a simple one-hot encoding of the nodes [Kipf and Welling, 2016, Berg et al., 2017, Murphy et al., 2019] or a random node coloring [Dasoulas et al., 2019, Sato et al., 2020]. On the other hand, in the non-asymptotic regime, there is evidence that the power of MPNN grows as a function of depth and width. Specifically, Loukas [2020] derived several hardness results for decision, optimization, and estimation graph problems. The key insight was that networks cannot solve many tasks when the product of their depth and width does not exceed a polynomial of the number of nodes.

1.2 New insights: communication capacity and graph isomorphism

Current results for the non-anonymous MPNN leave two important questions unanswered: First, it is unclear whether a depth-vs-width dependency is indicative only of worst-case distributions as proven in previous studies [Loukas, 2020] or if it constitutes a more general phenomenon. In addition, it is unknown whether a similar dependency holds for graph isomorphism—a problem of particular significance to graph neural networks [Xu et al., 2018, Morris et al., 2019, Chen et al., 2020].

To address these questions, this paper defines and characterizes the *communication capacity* of MPNN, a measure of the amount of information that the nodes can exchange during the forward pass. In Section 2 it is shown that the capacity of MPNN depends on the network's depth, width, and message-size, as well as on the cut-structure of the input graph. In essence, communication capacity is an effective generalization of the previously considered product between depth and width [Loukas, 2020] that takes into account more involved properties and also holds for MPNN with a global state [Gilmer et al., 2017, Battaglia et al., 2018, Ishiguro et al., 2019].

The paper then delves into the *communication complexity* of graph isomorphism. The theory of communication complexity compliments communication capacity as it provides a convenient mathematical framework to study how much information needs to be exchanged by parties that jointly compute a function [Rao and Yehudayoff, 2020]. In this setting, Section 3 derives the first hardness results for graph and tree isomorphism. It is shown that the communication capacity of MPNN needs to grow at least linearly with the number of nodes so that the network can learn to distinguish trees, and quadratically to distinguish between all connected graphs. This stands out from previous relevant works that have studied subcases of isomorphism, such as subgraph freeness [Even et al., 2017, Gonen and Oshman, 2018] or considered models that cannot solve isomorphism [Xu et al., 2018, Morris et al., 2019, Chen et al., 2020, Dehmamy et al., 2019, Magner et al., 2020]. In addition, the derived lower bounds rely on a newly developed mathematical technique which renders them applicable not only to worst-case instances [Loukas, 2020], but probabilistically over a portion of all inputs.

A large-scale empirical study reveals strong qualitative and quantitative agreement between the MPNN test accuracy and theoretical predictions. In the 12 graph and tree isomorphism tasks considered, the performance of the 420 networks trained was found to depend strongly on their communication capacity. In addition, the proposed theory could consistently predict which networks would exhibit poor classification accuracy as a function of their capacity and the type of task in question.

2 The communication capacity of message-passing networks

Suppose that a learner is given a graph $G = (\mathcal{V}, \mathcal{E}, a)$ sampled based on a distribution $\mathbb{D}$ over a finite universe of graphs $\mathcal{X}$. Throughout this paper, $\mathcal{V}$ will be used to denote the set of nodes of cardinality n, $\mathcal{E}$ the set of edges, and a encodes any node and edge features of interest. With G as input, the learner needs to predict the output of function $f : \mathcal{X} \to \mathcal{Y}$. This work focuses on graph classification, in which case f assigns a class $y \in \mathcal{Y}$ (e.g., its isomorphism class) to each graph in the universe.

A message-passing neural network N is a learner that operates as follows:

Set $x_i^{(0)} = a_i$ for all $v_i \in \mathcal{V}$.
for layer $\ell = 1, \ldots, d$ **do**
  for every edge $e_{ij} \in \mathcal{E}$ (in parallel) **do**

$$\text{msg}_{ij}^{(\ell)} = \textsc{Message}_\ell \left(x_j^{(\ell-1)},\, a_j,\, a_{ij} \right)$$

  for every node $v_i \in \mathcal{V}$ (in parallel) **do**

$$x_i^{(\ell)} = \textsc{Update}_\ell \Big(x_i^{(\ell-1)},\ \Big\{ \text{msg}_{ij}^{(\ell)} \ :\ e_{ij} \in \mathcal{E} \Big\} \Big)$$

return $\textsc{ReadOut}\left(\left\{ x_i^{(d)} : v_i \in \mathcal{V} \right\} \right)$.

In its essence, the message-passing model dictates that the node representations $x_i^{(\ell)}$ should be progressively updated by exchanging information along the edges of the graph. Each message $\text{msg}_{ij}^{(\ell)}$ contains some information that is sent to from node v_j to v_i. Each neuron in a network utilizes some alphabet $\mathcal{S}$ of cardinality $s = |\mathcal{S}|$ to encode its state. For this reason, $x_i^{(\ell)}$ and $\text{msg}_{ij}^{(\ell)}$ are selected from some finite sets $\mathcal{S}^{w_\ell}$ and $\mathcal{S}^{m_\ell}$, where w_ℓ and m_ℓ are the width and the message-size of the ℓ-th layer. For instance, to represent whether a neuron is activated it suffices to choose a binary alphabet, whereas a more general computer could use the set of numbers represented by 32-bits in floating point arithmetic.

The logic of the network is then determined by the *message*, *update*, and *readout* functions.

- In general, $\textsc{Message}_\ell$ and $\textsc{Update}_\ell$ are layer-dependent functions whose parameters are selected based on some optimization procedure. It is common to parametrize these functions by feed-forward neural networks [Scarselli et al., 2008, Li et al., 2015, Battaglia et al., 2018]. The rational is that, by the universal approximation theorem and its variants [Cybenko, 1989, Hornik et al., 1989, Lu et al., 2017], these networks can approximate any smooth function that maps vectors onto vectors.

- Function $\textsc{ReadOut}$ allows us to recover the final output from the node representations of the last layer. Two cases may be distinguished depending on the task at hand: (i) If the network's output is required to be invariant on the number of nodes, $\textsc{ReadOut}$ aggregates the decisions of individual nodes. (ii) When $f(G)$ assigns some class to every node, $\textsc{ReadOut}$ is the identity function.

In the pseudo-code above, all message exchange needs to occur along graph edges. However, one may also easily incorporate a *global state* (or external memory) to the model above by instantiating a special node v_0 and extending the edge set to contain edges from every other node to it. Global state is useful for incorporating graph features to the decision making [Battaglia et al., 2018] and there is some evidence that it can facilitate logical reasoning [Barceló et al., 2019]. Here, I will suppose that $x_0^{(\ell)}$ belongs to the set $\mathcal{S}^{g_\ell}$, with g_ℓ possibly being larger than w_ℓ.

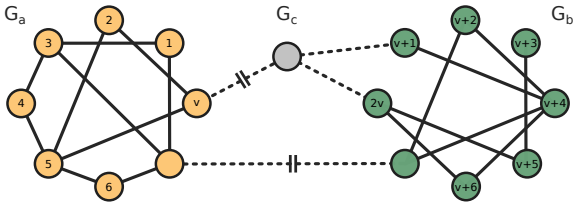


Figure 1: A visual depiction of a graph $G = (\mathcal{V}, \mathcal{E})$ chosen from $\mathcal{X}$. G_a (in yellow) and G_b (in green) are chosen from families $\mathcal{X}_a$ and $\mathcal{X}_b$ of graphs with v nodes. The edges of G_c (dashed lines) may connect to any node but should induce a $(\mathcal{V}_a, \mathcal{V}_b)$-cut of at most τ.

2.1 Communication capacity

Networks that rely on message-passing can exchange a bounded amount of information. To illustrate this phenomenon, imagine that there are two node- and edge-disjoint subgraphs $G_a = (\mathcal{V}_a, \mathcal{E}_a)$ and $G_b = (\mathcal{V}_b, \mathcal{E}_b)$ of G that are controlled by two parties: Alice and Bob. By construction, when Alice needs to send information to Bob, she does so by sending messages across some path that crosses between $\mathcal{V}_a$ and $\mathcal{V}_b$. Bob does the same. From this elementary observation, it can be deduced that the number of symbols that can be sent during the network's forward pass is bounded by the cut between the two parties:

Lemma 2.1 (Communication capacity)**.** *Let N be an MPNN of d layers, where each layer ℓ has width w_ℓ, exchanges messages of size m_ℓ, and maintains a global state of size g_ℓ. For any partitioning of G into $G_a = (\mathcal{V}_a, \mathcal{E}_a)$ and $G_b = (\mathcal{V}_b, \mathcal{E}_b)$ with $\mathcal{V}_a \cap \mathcal{V}_b = \mathcal{E}_a \cap \mathcal{E}_b = \varnothing$, the number of symbols c_N that can be transmitted from Alice to Bob (or from Bob and to Alice) is at most*

$$c_N \leq \mathit{cut}(\mathcal{V}_a, \mathcal{V}_b) \sum_{\ell=1}^{d} \min\{m_\ell, w_\ell\} + \sum_{\ell=1}^{d} g_\ell,$$

with $\mathit{cut}(\mathcal{V}_a, \mathcal{V}_b)$ being the size of the smallest cut that separates $\mathcal{V}_a$ from $\mathcal{V}_b$ in G.

Hence, any network N of finite size has bounded communication capacity. In Section 3 this limitation will be exploited in order to characterize what N cannot compute.

3 The communication complexity of graph isomorphism

This section derives necessary conditions for the communication capacity of a network that solves the graph isomorphism problem. Graph isomorphism entails finding a mapping $f_{\text{isom}} : \mathcal{X} \to \mathcal{Y}$ from a universe of labeled graphs to their corresponding graph isomorphism classes. Crucially, though the nodes of labeled graph G are assigned some predefined order (which constitutes their label in graph-theory nomenclature), the class $f_{\text{isom}}(G)$ should be invariant to this ordering.

The analysis follows a communication complexity-theoretic argument. Specifically, I consider the universe $\mathcal{X}$ of all labeled graphs $G = (\mathcal{V}, \mathcal{E})$ admitting to the following $(\mathcal{X}_a, \mathcal{X}_b, \tau)$ decomposition:

1. Subgraph $G_a = (\mathcal{V}_a, \mathcal{E}_a)$ induced by labels $\mathcal{V}_a = (1, 2, \cdots, v)$ belongs to $\mathcal{X}_a$.
2. Subgraph $G_b = (\mathcal{V}_b, \mathcal{E}_b)$ induced by labels $\mathcal{V}_b = (v+1, v+2, \cdots, 2v)$ belongs to $\mathcal{X}_b$.
3. Subgraph $G_c = (\mathcal{V}, \mathcal{E} \setminus (\mathcal{E}_a \cup \mathcal{E}_b))$ yields $\mathit{cut}(\mathcal{V}_a, \mathcal{V}_b) \leq \tau$.

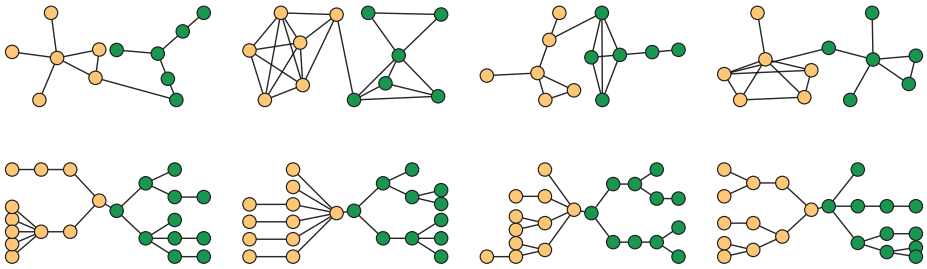


Figure 2: Example graphs sampled from two $(\mathcal{X}_a, \mathcal{X}_b, 1)$ decompositions. Top: $\mathcal{X}_a$ and $\mathcal{X}_b$ contain all connected graphs on $v = 6$ nodes (special case of Theorems 3.1 and 3.2). Bottom: $\mathcal{X}_a$ and $\mathcal{X}_b$ contain all trees on 11 nodes (special case of Theorem 3.3). In both cases, there exists a $\tau = 1$ cut between the nodes $\mathcal{V}_a$ controlled by Alice (in yellow) and nodes $\mathcal{V}_b$ controlled by Bob (in green).

A $(\mathcal{X}_a, \mathcal{X}_b, \tau)$ decomposition is fairly general: the main restriction placed is that the cut between $\mathcal{V}_a$ and $\mathcal{V}_b$ is bounded by τ. Families $\mathcal{X}_a$ and $\mathcal{X}_b$ can be chosen to contain relevant families of graphs (e.g., all connected graphs or all trees), whereas the nodes and edges of G_c may be selected arbitrarily.

To derive lower bounds, it will be imagined that $G_a = (\mathcal{V}_a, \mathcal{E}_a)$ and $G_b = (\mathcal{V}_b, \mathcal{E}_b)$ are known to Alice and Bob, respectively. The goal of the two parties is to determine $f_{\text{isom}}(G) = f_{\text{isom}}(G_a, G_b, G_c)$ by exchanging as little information as possible — see Section 3.1 for precise definitions and Appendix B for more information. Two main results will be proven: Section 3.2 will show that, when $\mathcal{X}_a$ and $\mathcal{X}_b$ contain all labeled connected graphs on v nodes, the number of symbols Alice and Bob must exchange grows quadratically with v. Moreover, Section 3.3 will show that, when $\mathcal{X}_a$ and $\mathcal{X}_b$ contain only trees, the dependence on v is linear.

With these results in place, Section 3.4 will argue that MPNN needs to have capacity that is at least half the communication complexity to compute f_{isom}. Since the derived impossibility results extend also to any universe $\mathcal{X}'$ that is a superset of a universe $\mathcal{X}$ that admits to a $(\mathcal{X}_a, \mathcal{X}_b, \tau)$ decomposition, the impossibility results also hold for the general problems of graph and tree isomorphism.

3.1 Communication complexity

Suppose that Alice and Bob wish to jointly compute a function $f : \mathcal{X}_a \times \mathcal{X}_b \to \mathcal{Y}$ that depends on both their inputs. Alice's input is an element $x_a \in \mathcal{X}_a$ and Bob sees an element $x_b \in \mathcal{X}_b$. In the graph isomorphism problem, x_a and x_b correspond to subgraphs G_a and G_b, respectively. To learn $f(x_a, x_b)$, the two parties need to exchange information based on some communication *protocol* π.

Worst-case complexity. The focus of classical theory is on the worst-case input. Denote by $\|\pi(x_a, x_b)\|_m$ the number of symbols that Alice and Bob need to exchange in order to compute $f(x_a, x_b)$ using protocol π. Subscript $m \in \{one, both\}$ indicates whether "successful computation" entails one or both parties figuring out $f(x_a, x_b)$ at the end of the exchange. The *communication complexity* [Rao and Yehudayoff, 2020] of f is defined as

$$c_f^m := \min_{\pi} \max_{(x_a, x_b) \in \mathcal{X}_a \times \mathcal{X}_b} \|\pi(x_a, x_b)\|_m \tag{1}$$

and corresponds to the minimum worst-case length of any protocol that computes f.

Expected complexity. In machine learning, rather than the worst-possible case, one usually cares about the expected behavior of a learner when its input is sampled from a distribution. Concretely,

let (X_a, X_b) be random variables sampled from a distribution $\mathbb{D}$ with domain $\mathcal{X}_a \times \mathcal{X}_b$. The expected length of a protocol π is

$$\mathrm{E}_{\mathbb{D}}\big[c_f^m(\pi)\big] \coloneqq \sum_{(x_a,x_b)\in\mathcal{X}_a\times\mathcal{X}_b} \|\pi(x_a,x_b)\|_m \cdot \mathrm{P}(X_a = x_a, X_b = x_b)\,, \tag{2}$$

where now the protocol length $\|\pi(x_a,x_b)\|_m$ is weighted according to the probability of each input. With this in place, I define the *expected communication complexity* of f as

$$c_f^m(\mathbb{D}) \coloneqq \min_{\pi} \mathrm{E}_{\mathbb{D}}\big[c_f^m(\pi)\big]\,, \tag{3}$$

corresponding to the minimum expected length of any protocol that computes f.

For an overview of the classical theory of communication complexity pertaining to the worst-case and an analysis of the newly-defined expected complexity, the reader may refer to Appendix B.

3.2 Graph isomorphism

I first focus on the case where $\mathcal{X}_a$ and $\mathcal{X}_b$ contain all connected graphs on v nodes. Some examples of graphs arising from a $(\mathcal{X}_a, \mathcal{X}_b, 1)$ decomposition can be found in Figure 2.

The first result is distribution agnostic:

Theorem 3.1 (Graph isomorphism)**.** *When $\mathcal{X}_a$ and $\mathcal{X}_b$ each contain the set of all connected graphs on v nodes, the worst-case communication complexity of f_{isom} is at least*

$$c^{both}_{f_{isom}} \geq \frac{v^2}{\log_2 s} - 2v\log_s\left(\frac{v\sqrt{2}}{e}\right) - \log_s\left(2ve^2\right) + o(1) = \beta \quad \textit{and} \quad c^{one}_{f_{isom}} \geq \frac{\beta - (\log_2 s)^{-1}}{2}.$$

The proposed bound is asymptotically tight: the two parties should exchange $\Theta(v^2/\log_2 s)$ symbols in the worst case. The tightness is a consequence of the following elementary upper bound: to compute $f_{\mathrm{isom}}(G)$, Bob and Alice can simply send their entire edge-sets to each other and proceed to compute $f(G_a, G_b, G_c)$ independently. Then, since the number of edges of a graph v nodes are $|\mathcal{E}_a|, |\mathcal{E}_b| \leq v(v-1)/2$, it suffices to exchange $c_{f_{\mathrm{isom}}} \leq v(v-1)/\log_2 s$ symbols.

As it turns out, a similar bound holds also in the *random graph model* $\mathbb{G}_{v,p}$. In $\mathbb{G}_{v,p}$, every graph with v nodes and k edges is sampled with probability

$$\mathrm{P}(G \sim \mathbb{G}_{v,p}) = p^k(1-p)^{\binom{v}{2}-k}.$$

Effectively, this means the probability of choosing each graph depends only on the number of edges it contains. Moreover, for $p = 0.5$ each graph is sampled uniformly at random from the set of all possible graphs. The following theorem bounds the expected communication complexity when the subgraphs known to Alice and Bob are sampled from $\mathbb{G}_{v,p}$:

Theorem 3.2 (Random graph isomorphism)**.** *Let G_a and G_b be sampled independently from $\mathbb{G}_{v,p}$, with $p > \log v/v$ and $cut(\mathcal{V}_a, \mathcal{V}\setminus\mathcal{V}_a) = cut(\mathcal{V}_b, \mathcal{V}\setminus\mathcal{V}_b) = 1$. Denote by $\mathbb{B}_{v,p}$ the resulting distribution. With high probability,*

$$c^{both}_{f_{isom}}(\mathbb{B}_{v,p}) \geq v^2\,\mathrm{H}_s(p) - v\left(2\log_s\left(\frac{v}{e}\right) + \mathrm{H}_s(p)\right) - \log_s\left(2ve^2\right) = \beta$$

and

$$c^{one}_{f_{isom}}(\mathbb{B}_{v,p}) \geq \frac{\beta}{2} - \frac{v^2 - v(1-\mathrm{H}_2(p)) + 1}{2\log_2 s},$$

where $\mathrm{H}_s(p) = -(1-p)\log_s(1-p) - p\log_s p$ is the binary entropy function (base s).

The expected complexity, therefore, grows asymptotically with $\Omega(v^2\,\mathrm{H}_s(p))$ and is maximized when every graph in the universe is sampled with equal probability (i.e., for $p = 0.5$). Interestingly, in this setting, the bounds of Theorems 3.1 and Theorem 3.2 match. This implies that, unless there is some strong isomorphism class imbalance in the dataset, the communication complexity lower bound posed by Theorem 3.1 does not only concern rare worst-case inputs, but should be met on average.

Note also that in the theorem it is asserted that $p > \log v/v$. The latter condition suffices to guarantee that every $G \sim \mathbb{B}_{v,p}$ will be connected with high probability.

3.3 Tree isomorphism

I also bound the communication complexity of tree isomorphism. In this case, $\mathcal{X}_a$ and $\mathcal{X}_b$ contain all trees on v nodes. Some examples of trees arising from a $(\mathcal{X}_a, \mathcal{X}_b, 1)$ decomposition can be found in Figure 2.

The following is proved:

Theorem 3.3 (Tree isomorphism)**.** *Suppose that G_a and G_b are sampled independently from the set of all trees on v nodes. Denote by $\mathbb{T}_v$ the resulting distribution. The communication complexity of f_{isom} is at least*

$$c^{both}_{f_{isom}} \geq c^{both}_{f_{isom}}(\mathbb{T}_v) \gtrsim 2v\log_s \alpha - 5\log_s v + \log_s 7 = \beta \quad \textit{and} \quad c^{one}_{f_{isom}} \geq c^{one}_{f_{isom}}(\mathbb{T}_v) \gtrsim \frac{\beta + \log_s 2}{2},$$

where $\alpha \approx 2.9557652$ and $f(n) \gtrsim g(n)$ means $f(n) \geq g(n)$ as n grows.

Perhaps as expected, discriminating trees is significantly easier. For trees, the communication complexity grows asymptotically with $\Theta(v)$, rather than quadratically as in Theorems 3.1 and 3.2.

Further, akin to the general case, the expected and worst-case complexities match when every tree is sampled with equal probability. Since a distribution over trees cannot be meaningfully parametrized based a connection probability p (trees always have the same number of edges), by default in $\mathbb{T}_v$ every $G \in \mathcal{X}$ is sampled with equal probability.

3.4 Consequences for message-passing neural networks

Two types of networks are distinguished depending on how the readout function operates:

1. ReadOut performs *majority-voting*. Specifically, for N to compute $f_{isom}(G)$ there should exist a function $g : \mathcal{S}^{w_d} \to \mathcal{Y}$ and a set of nodes $\mathcal{M}_G \subseteq \mathcal{V}$ possibly dependent on G and of cardinality at least $|\mathcal{M}_G| \geq \mu = O(1)$, such that $g(x_i^{(d)}) = f_{isom}(G)$ for every $v_i \in \mathcal{M}_G$.
2. ReadOut performs *consensus*. This is akin to a majority-voting, with the distinction that $\mathcal{M}_G$ should contain at least $|\mathcal{M}_G| \geq n - \mu = \Omega(n)$ nodes.

The next result makes the connection between communication complexity and MPNN explicit:

Proposition 3.1. *Let $\mathbb{D}$ be a distribution over graphs that is densely supported on a universe $\mathcal{X}$ admitting to a $(\mathcal{X}_a, \mathcal{X}_b, \tau)$ decomposition. Further, suppose that N is an MPNN whose communication capacity is at most c_N. The following statements hold:*

- *If $2c_N < c^m_{f_{isom}}$, then N cannot compute $f_{isom}(G)$ for at least one $G \in \mathcal{X}$.*
- *If $\mathcal{X} \subset \mathcal{X}'$ and $2c_N < c^m_{f_{isom}}$, then N cannot compute $f_{isom}(G)$ for at least one $G \in \mathcal{X}'$.*

- *If* $2c_N < \delta\, c^m_{f_{isom}}(\mathbb{D})$ *for some* $\delta \in [0,1]$*, then* N *cannot compute* $f_{isom}(G)$ *with probability at least* $\frac{1-\delta}{(c^m_{f_{isom}}(\mathbb{G})/\beta_m)-\delta}$.

Above, with majority-voting one should set $m = one$*,* $\beta_{one} \leq \log_s(\min\{|\mathcal{X}_a|, |\mathcal{X}_b|\})$ *and* $v > (n-\mu)/2$*, whereas with consensus* $m = both$*,* $\beta_{both} \leq \log_s(|\mathcal{X}_a| + |\mathcal{X}_b|)$*, and* $v > \mu$*.*

It is then a direct corollary of Proposition 3.1 together with Lemma 2.1, Theorems 3.1, 3.2 and 3.3 that an MPNN of sub-quadratic and sub-linear capacity cannot compute the isomorphism class of connected graphs and trees, respectively.

4 Empirical results

This section tests the developed theory on 12 graph and tree isomorphism tasks of varying difficulty. In the 420 neural networks tested, the bounds are found to consistently predict when each network can solve a given task as a function of its capacity.

4.1 Experimental setting

In the considered experiments, MPNN of different capacities were tasked with learning the mapping between a universe of graphs their corresponding isomorphism classes.

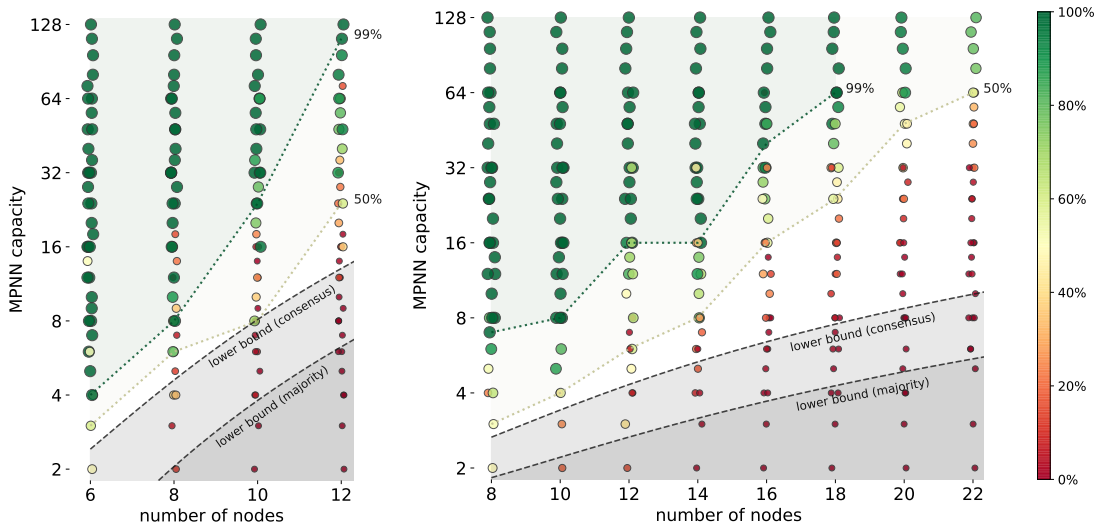
Datasets. A total of 12 universes were constructed: graph universes $\mathcal{X}^n_{\text{graph}}$ for $n = (6, 8, 10, 12)$ and tree universes $\mathcal{X}^n_{\text{tree}}$ for $n = (8, 10, \ldots, 22)$. Examples of graphs from $\mathcal{X}^{12}_{\text{graph}}$ and $\mathcal{X}^{22}_{\text{tree}}$ can are shown in Figure 2. Each $\mathcal{X}^n_{\text{graph}}$ was built in two steps: First, `geng` [McKay and Piperno, 2014] was used to populate $\mathcal{X}_a$ and $\mathcal{X}_b$ with all possible connected graphs on $v = n/2$ nodes. Then, each $G \in \mathcal{X}^n_{\text{graph}}$ was generated by selecting G_a and G_b from $\mathcal{X}_a$ and $\mathcal{X}_b$ and connecting them with an edge, such that $\tau = 1$. The labels added to the nodes of G were the one-hot encoding of a random permutation of $(1, \ldots, v)$ and $(v+1, \ldots, n)$. The construction of $\mathcal{X}^n_{\text{tree}}$ differed only in that $\mathcal{X}_a$ and $\mathcal{X}_b$ contained all trees on $v = n/2$ nodes. Each dataset was split into a training, a validation, and a test set (covering 90%, 5%, and 5% of the dataset, respectively). Additional details are provided in Appendix A.

Architecture and training. The networks combined multiple GIN0 [Xu et al., 2018] layers with batch normalization and a sum readout function. Their depth and width varied in $d \in (2, 3, 4, 5, 6, 7, 8)$ and $w \in (1, 2, 4, 8, 16)$, respectively, the message-size was set equal to w, and no global state was used. Each network was trained using Adam with a decaying learning rate. Early stopping was employed when the validation accuracy reached 100%.

4.2 Findings

Let me begin by stating that networks of sufficient size could solve nearly every isomorphism task up to 100% test accuracy (Table 2 in Appendix A), which corroborates previous theoretical findings that (non-anonymous) MPNN are more powerful than their anonymous counterparts and that they can learn to be permutation invariant [Murphy et al., 2019, Loukas, 2020, Dasoulas et al., 2019].

Figures 3a and 3b summarize the neural network performance for all graph- and tree-isomorphism tasks considered. The achieved accuracy strongly correlated with communication capacity (computed based on Lemma 2.1) with larger-capacity networks performing consistently better. Moreover, in qualitative agreement with the analysis, solving a task can be seen to necessitate larger capacity when the number of nodes is increased. A case in point, whereas a capacity of 4 suffices to classify 99% of



(a) graph isomorphism *(b) tree isomorphism*

Figure 3: Test accuracy in terms of communication capacity and the number of nodes for 4 graph (left) and 8 tree isomorphism tasks (right). Each marker corresponds to a trained network. Networks of high (low) accuracy as plotted with large green (small red) markers. The two dashed colored lines connect the smallest-capacity networks that attain 50% and 99% accuracy, respectively. The two gray regions at the bottom of the figure correspond to the proposed distribution-dependent lower bounds for a majority and consensus readout function. Best seen in color.

graphs of 6 nodes correctly, for 8, 10, and 12 nodes the required capacity increases to 8, 24, and 112, respectively. This identified correlation between capacity and accuracy could not be explained by the depth or width of the network alone, as, in most instances, tasks that could not be solved by wide and shallow networks could also not be solved by deep networks of the same capacity. The only exception was when the depth was too small, in which case the receptive field did not cover the entire graph (see Figures 5a and 5b in Appendix A).

The gray regions at the bottom of each figure indicate the proposed expected communication complexity lower bounds. Here, $|\mathcal{S}| = 2$ based on the interpretation that each neuron can be either in an activated state or not. There are also two lower bounds plotted since a network that sums the final layer's node representations can learn to differentiably approximate both a majority-voting and a consensus function. The analysis asserts that a network with capacity below the gray dashed lines should not be able to solve the isomorphism problem for a significant fraction of all inputs (see precise statement in Proposition 3.1). Indeed, networks in the gray region consistently perform poorly. The empirical accuracy appears to match closely the consensus bound, though it remains inconclusive if the network is actually learning to do consensus. A closer inspection of the results (see Figures 4a and 4b in Appendix A) also reveals that the poor performance of the networks in the gray region is not an issue of generalization. In agreement with the theory, networks of insufficient communication capacity do not possess the expressive power to map a fraction of all inputs to the right isomorphism class, irrespective of whether these graphs appear in the training or test set.

5 Conclusion

This work proposed a hardness-result for graph isomorphism in the MPNN model by characterizing the amount of information the nodes can exchange during their forward pass (termed communication capacity). The developed proof techniques may be of independent interest: they innovate upon classical communication complexity arguments by considering the expected as well as worst-case complexity. A closer examination of the proofs also reveals connections between hardness and the entropy of the classification function (see Lemmas B.2 and C.1) that could possibly be exploited for other graph problems. The exploration of these directions is left for future work.

From a practical perspective, this work provided evidence that, if the amount of training data is not an issue, graph isomorphism is hard but not impossible for MPNN. Specifically, it was argued that the number of parameters needs to increase quadratically with the number of nodes. The implication is that, in the most general case, networks of practical size should be able to solve the isomorphism problem for graphs with at most a few dozen nodes, but will encounter issues otherwise.

Acknowledgements. I would like to thank Nikolaos Karalias and Giovanni Cherubin for their helpful discussions, as well as the Swiss National Science Foundation for supporting this work in the context of the project "*Deep Learning for Graph-Structured Data*" (grant number PZ00P2 179981).

References

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Hrushikesh N Mhaskar and Tomaso Poggio. Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, 14(06):829–848, 2016.

Boris Hanin and Mark Sellke. Approximating continuous functions by relu nets of minimal width. *arXiv preprint arXiv:1710.11278*, 2017.

Henry W Lin, Max Tegmark, and David Rolnick. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6):1223–1247, 2017.

Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6231–6239. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7203-the-expressive-power-of-neural-networks-a-view-from-the-width.pdf`.

J Pedro Neto, Hava T Siegelmann, J Félix Costa, and CP Suárez Araujo. Turing universality of neural nets (revisited). In *International Conference on Computer Aided Systems Theory*, pages 361–366. Springer, 1997.

Jorge Pérez, Javier Marinković, and Pablo Barceló. On the turing completeness of modern neural network architectures. *arXiv preprint arXiv:1901.03429*, 2019.

Haggai Maron, Ethan Fetaya, Nimrod Segol, and Yaron Lipman. On the universality of invariant networks. *arXiv preprint arXiv:1901.09342*, 2019.

Nicolas Keriven and Gabriel Peyré. Universal invariant and equivariant graph neural networks. *arXiv preprint arXiv:1905.04943*, 2019.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

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.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in neural information processing systems*, pages 4502–4510, 2016.

Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3693–3702, 2017.

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, pages 4602–4609, 2019.

Ryoma Sato, Makoto Yamada, and Hisashi Kashima. Approximation ratios of graph neural networks for combinatorial problems. In *Advances in Neural Information Processing Systems*, 2019.

Pablo Barceló, Egor V Kostylev, Mikael Monet, Jorge Pérez, Juan Reutter, and Juan Pablo Silva. The logical expressiveness of graph neural networks. In *International Conference on Learning Representations*, 2019.

Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. Can graph neural networks count substructures? *arXiv preprint arXiv:2002.04025*, 2020.

Nima Dehmamy, Albert-László Barabási, and Rose Yu. Understanding the representation power of graph neural networks in learning graph topology. *arXiv preprint arXiv:1907.05008*, 2019.

Vikas K Garg, Stefanie Jegelka, and Tommi Jaakkola. Generalization and representational limits of graph neural networks. *arXiv preprint arXiv:2002.06157*, 2020.

Abram Magner, Mayank Baranwal, and Alfred O Hero III. The power of graph convolutional networks to distinguish random graph models: Short version. *arXiv preprint arXiv:2002.05678*, 2020.

Floris Geerts, Filip Mazowiecki, and Guillermo A Pérez. Let's agree to degree: Comparing graph convolutional networks in the message-passing framework. *arXiv preprint arXiv:2004.02593*, 2020.

Ryoma Sato. A survey on the expressive power of graph neural networks. *ArXiv*, abs/2003.04078, 2020.

R Murphy, B Srinivasan, V Rao, and B Riberio. Relational pooling for graph representations. In *International Conference on Machine Learning (ICML 2019)*, 2019.

Andreas Loukas. What graph neural networks cannot learn: depth vs width. In *International Conference on Learning Representations*, ICLR, 2020.

George Dasoulas, Ludovic Dos Santos, Kevin Scaman, and Aladin Virmaux. Coloring graph neural networks for node disambiguation. *arXiv preprint arXiv:1912.06058*, 2019.

Ryoma Sato, Makoto Yamada, and Hisashi Kashima. Random features strengthen graph neural networks. *arXiv preprint arXiv:2002.03155*, 2020.

Rianne van den Berg, Thomas N Kipf, and Max Welling. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2017.

Katsuhiko Ishiguro, Shin-ichi Maeda, and Masanori Koyama. Graph warp module: an auxiliary module for boosting the power of graph neural networks. *arXiv preprint arXiv:1902.01020*, 2019.

Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020. doi: 10.1017/9781108671644.

Guy Even, Orr Fischer, Pierre Fraigniaud, Tzlil Gonen, Reut Levi, Moti Medina, Pedro Montealegre, Dennis Olivetti, Rotem Oshman, Ivan Rapaport, et al. Three notes on distributed property testing. In *31st International Symposium on Distributed Computing (DISC 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

Tzlil Gonen and Rotem Oshman. Lower bounds for subgraph detection in the congest model. In *21st International Conference on Principles of Distributed Systems (OPODIS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, {II}. *Journal of Symbolic Computation*, 60(0):94 – 112, 2014. ISSN 0747-7171. doi: http://dx.doi.org/10.1016/j.jsc.2013.09.003. URL `http://www.sciencedirect.com/science/article/pii/S0747717113001193`.

Philippe Flajolet and Robert Sedgewick. *Analytic combinatorics*. cambridge University press, 2009.

Alan Frieze and Michał Karoński. *Introduction to random graphs*. Cambridge University Press, 2016.

Richard Otter. The number of trees. *Annals of Mathematics*, pages 583–599, 1948.

A Additional empirical results

This section presents the empirical results more comprehensively.

First, Table 1 provides summary statistics for each of the 12 isomorphism tasks considered:

	$\mathcal{X}^{6}_{\text{graph}}$	$\mathcal{X}^{8}_{\text{graph}}$	$\mathcal{X}^{10}_{\text{graph}}$	$\mathcal{X}^{12}_{\text{graph}}$	$\mathcal{X}^{8}_{\text{tree}}$	$\mathcal{X}^{10}_{\text{tree}}$	$\mathcal{X}^{12}_{\text{tree}}$	$\mathcal{X}^{14}_{\text{tree}}$	$\mathcal{X}^{16}_{\text{tree}}$	$\mathcal{X}^{18}_{\text{tree}}$	$\mathcal{X}^{20}_{\text{tree}}$	$\mathcal{X}^{22}_{\text{tree}}$
classes	3	21	231	6328	3	6	21	66	276	1128	5671	22730
degree (avg.)	4.0	4.7	5.4	6.0	3.5	3.6	3.7	3.7	3.8	3.8	3.8	3.8
diameter (avg.)	3.7	4.5	5.0	5.4	4.0	4.3	5.0	5.4	6.0	6.4	6.9	7.3
dataset size	10k	10k	40k	100k	10k	10k	40k	40k	40k	40k	40k	100k

Table 1: Details relevant to the 4 graph and 8 tree isomorphism tasks.

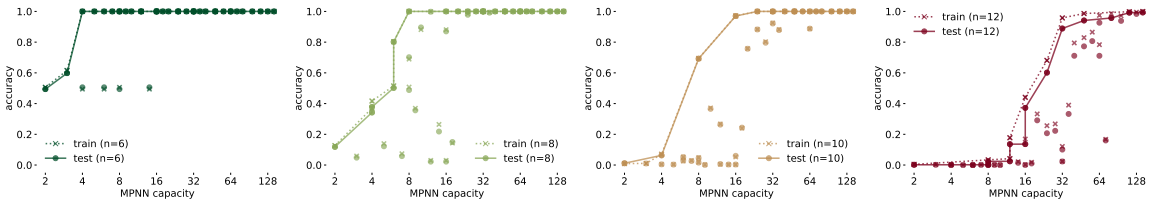
Table 2 provides empirical evidence that, with a one-hot encoding of the node-ordering given as features and a sufficiently large training set, MPNN of sufficient capacity can solve graph isomorphism. In the

current experiment, a large network (depth = 10 and width = 32) is seen to solve most isomorphism instances. The network did not achieve perfect classification for larger graphs, but better results can be achieved with more training data.

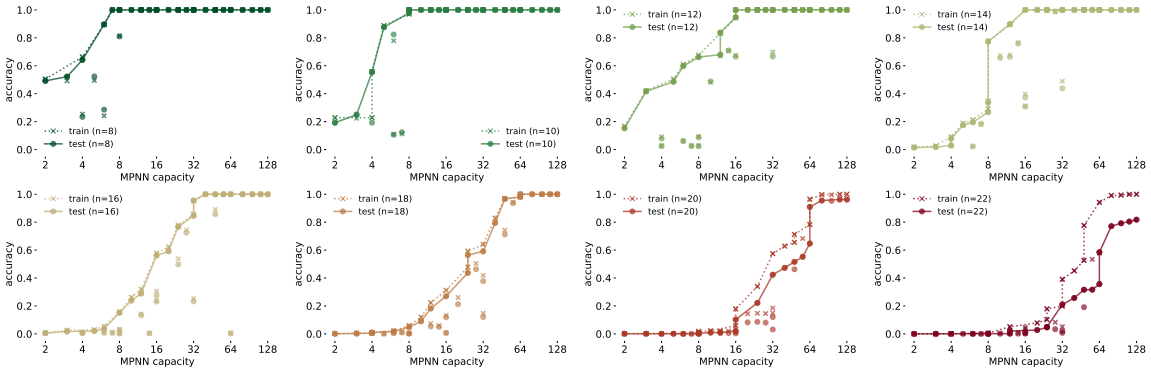
accuracy	$\mathcal{X}^{6}_{\text{graph}}$	$\mathcal{X}^{8}_{\text{graph}}$	$\mathcal{X}^{10}_{\text{graph}}$	$\mathcal{X}^{12}_{\text{graph}}$	$\mathcal{X}^{8}_{\text{tree}}$	$\mathcal{X}^{10}_{\text{tree}}$	$\mathcal{X}^{12}_{\text{tree}}$	$\mathcal{X}^{14}_{\text{tree}}$	$\mathcal{X}^{16}_{\text{tree}}$	$\mathcal{X}^{18}_{\text{tree}}$	$\mathcal{X}^{20}_{\text{tree}}$	$\mathcal{X}^{22}_{\text{tree}}$
training	100%	100%	100%	99.997%	100%	100%	100%	100%	100%	100%	100%	100%
validation	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	97.45%	82.82%
test	100%	100%	100%	99.96%	100%	100%	100%	100%	100%	100%	97.35%	82.92%

Table 2: The performance of a large-capacity MPNN.

The achieved accuracy of all networks considered is shown in Figures 4a and 4b for graph and tree isomorphism tasks, respectively. In contrast to the figures of Section 4, these plots: depict the training as well as testing accuracy. For the majority of tasks the test and training accuracy is almost identical. Overfitting can be a problem for larger graphs (e.g., trees of at least 20 nodes). The problem can be mitigated by increasing the size of the training set.



(a) graph isomorphism

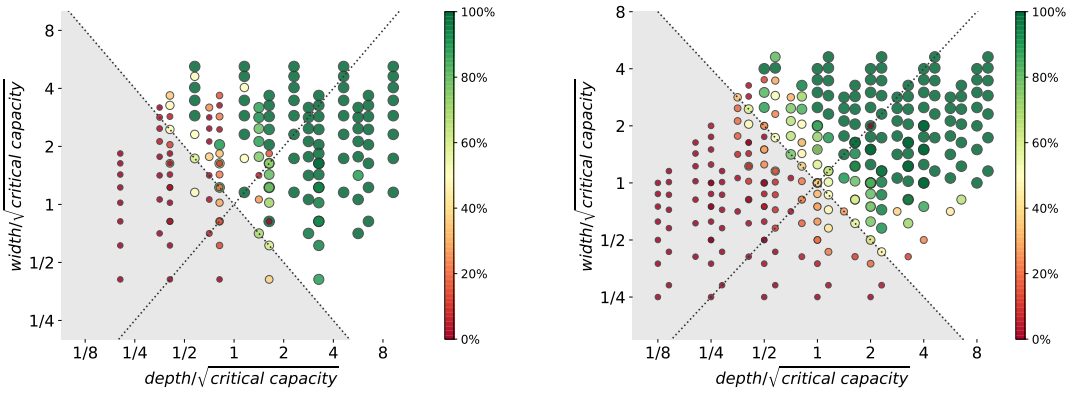


(b) tree isomorphism

Figure 4: Training and test accuracy as a function of communication capacity.

Finally, Figures 5a and 5b demonstrate that depth and width are partially exchangeable for graph and tree isomorphism. This implies that the correlation between capacity and accuracy (see Figures 3a and 3b) cannot be explained by only looking at the depth or width of a network. Here, the two figures depict the empirical test accuracy (by the marker color and size) as a function of depth and width for all graph and tree isomorphism tasks. For each task, the depth and width have been normalized by the square root of the critical capacity, corresponding to the smallest communication capacity of any network that could achieve at least 50% accuracy. As a consequence of the normalization, all networks in the top-right region (in white) possess sufficient capacity for the task at hand. Moreover, networks

plotted below (above) the main diagonal are deeper than they are wide (wider than they are deep). As seen, the isomorphism task can be solved by both wide and deep networks of super-critical capacity, as long as the networks are not too shallow. Indeed, networks of very small depth cannot see the entire graph and thus have poor accuracy.



(a) graph isomorphism *(b) tree isomorphism*

Figure 5: Accuracy as a function of capacity-normalized depth and width. Depth and width are partially exchangeable for graph and tree isomorphism.

B Communication complexity: basics and beyond

B.1 Basic theory: protocols

Let's start by denoting by $\mathcal{S}$ the common set of symbols[1] Alice and Bob use to communicate and denote by $s = |\mathcal{S}|$ its cardinality. A protocol π is described in terms of a rooted s-ary tree, i.e., a tree with a clearly defined root and in which every internal node has exactly s children. In addition, every internal node v is owned by either Alice or Bob and each one of the node's children symbolizes a symbol sent by its owner. Specifically, the protocol associates v with a function π_v that maps the input of v's owner to $\mathcal{S}$ (or equivalently to one of v's children). The protocol operates as follows: first, both parties set the *current* vertex to be the root of the tree. Say that the current vertex is v. If the owner of v is Alice then she announces symbol $\pi_v(x)$ and otherwise Bob announces $\pi_v(y)$. Both parties then update the current vertex to point to the child of v indicated by the value of π_v. This procedure is repeated until a leaf is found and said leaf becomes the protocol's output.

It can be seen that the number of symbols $\|\pi(x_a, x_b)\|_m$ Alice and Bob need to send in order to jointly compute $f(x_a, x_b)$ using protocol π equals the length of the path from the root to the leaf $\pi(x_a, x_b)$. Moreover, the number of symbols sent by a protocol in the worst case (i.e., for any input) is at most equal to the depth of the protocol tree (Fact 1.1 in [Rao and Yehudayoff, 2020]).

[1] Though usually it is assumed that the parties communicate using binary symbols, i.e., $\mathcal{S} = \{0, 1\}$, the set could also be defined more abstractly to contain s symbols.

B.2 Basic theory: monochromatic rectangles

To understand how protocols operate, one needs to consider the concept of rectangles. A *rectangle* is a subset of $\mathcal{X}_a \times \mathcal{X}_b$ that can be expressed as $\mathcal{X}'_a \times \mathcal{X}'_b$ for some $\mathcal{X}'_a \subset \mathcal{X}_a$ and $\mathcal{X}'_b \subset \mathcal{X}_b$.

As it turns out, every protocol can be described in terms of rectangles. Let $\mathcal{R}^v \subseteq \mathcal{X}_a \times \mathcal{X}_b$ be the set of inputs leading to a path that crosses a node $v \in \pi$. Moreover, define the following sets:

$$\mathcal{X}_a^v = \{x \in \mathcal{X}_a : \exists y \in \mathcal{X}_b \text{ such that } (x_a, x_b) \in \mathcal{R}^v\}$$
$$\mathcal{X}_b^v = \{y \in \mathcal{X}_b : \exists x \in \mathcal{X}_a \text{ such that } (x_a, x_b) \in \mathcal{R}^v\}$$

The following result clarifies the connection between protocols and rectangles.

Lemma B.1 (Lemma 1.4 in [Rao and Yehudayoff, 2020])**.** *For every protocol π and vertex v, $\mathcal{R}^v$ is a rectangle with $\mathcal{R}^v = \mathcal{X}_a^v \times \mathcal{X}_b^v$. Further, the rectangles $\mathcal{R}^\ell$ given by the leafs $\ell \in \mathcal{L}_\pi$ of the protocol tree form a partition of $\mathcal{X}_a \times \mathcal{X}_b$.*

It is not hard to realize that, for every leaf $\ell \in \mathcal{L}_\pi$, the function f should always take the same value at every $(x_a, x_b) \in \mathcal{R}^\ell$ in order for both parties to be able to compute the output from $\pi(x_a, x_b)$. Such rectangles are referred to as *monochromatic*: concretely, a rectangle $\mathcal{R} \subset \mathcal{X}_a \times \mathcal{X}_b$ is monochromatic if $f(x_a, x_b) = f(x'_a, x'_b)$ for every $(x_a, x_b), (x'_a, x'_b) \in \mathcal{R}$. Indeed, if leaf rectangles were not monochromatic, Alice and Bob would not be able to identify the output of f based on $\mathcal{R}^\ell$.

The following theorem is obtained by combining Lemma B.1 with the fact that the minimum depth of any s-ary tree with $s^{c_f^{both}}$ leafs is c_f^{both}.

Theorem B.1 (Theorem 1.6 by Rao and Yehudayoff [2020])**.** *If the communication complexity of $f : \mathcal{X}_a \times \mathcal{X}_b \to \mathcal{Y}$ is c_f^{both}, then $\mathcal{X}_a \times \mathcal{X}_b$ can be partitioned into at most $s^{c_f^{both}}$ monochromatic rectangles.*

The following is a direct corollary:

Corollary B.1 (Rao and Yehudayoff [2020])**.** *If $\mathcal{X}_a \times \mathcal{X}_b$ cannot be partitioned into s^c monochromatic rectangles, then $c_f^{both} \geq c$.*

A simple way to satisfy the requirement of the corollary is to prove that no large monochromatic rectangle exists. For instance, if it is shown that all monochromatic rectangles have size bounded by k^2 then every monochromatic partitioning must contain at least $|\mathcal{X}_a \times \mathcal{X}_b|/k^2$ rectangles and the complexity is at least $c_f^{both} \geq \log_s \left(|\mathcal{X}_a \times \mathcal{X}_b|/k^2\right)$. I will rely on this method in the following to derive lower bounds on the worst-case communication complexity of different functions.

B.3 Beyond the basics: expected communication complexity

The following lemma connects $\mathrm{E}_{\mathbb{D}}[c_f(\pi)]$ to the entropy of the categorical distribution induced by the leafs of the protocol tree.

Lemma B.2. *Let the random variables $X = (X_a, X_b) \sim \mathbb{D}$ be sampled from some distribution $\mathbb{D}$ and, moreover, let random variable L_π be the leaf for a protocol π that computes $f(X_a, X_b)$. The expected communication complexity of f is*

$$\min_\pi \mathrm{H}_s(L_\pi) \leq c_f^m(\mathbb{D}) \leq \min_\pi \mathrm{H}_s(L_\pi) + 1,$$

where $\mathrm{H}_s(L_\pi)$ is the Shannon entropy (base s) of L_π under $\mathbb{D}$.

Proof. The expected length of a protocol π is

$$\begin{aligned}\mathrm{E}_{\mathbb{D}}\big[c_f^m(\pi)\big] &= \sum_{x_a,x_b} \|\pi(x_a,x_b)\|_m \cdot \mathrm{P}(X_a = x_a, X_b = x_b) \\ &= \sum_{\ell\in\mathcal{L}_\pi} \mathrm{depth}(t) \cdot \mathrm{P}(L_\pi = \ell) \\ &= \mathrm{E}_{\mathbb{D}}[\mathrm{depth}(L_\pi)]\,.\end{aligned}$$

Note that the set $\mathcal{L}_\pi$ contains the leafs of the protocol tree and L_π is a categorical random variable over leafs with

$$\mathrm{P}(L_\pi = \ell) = \sum_{x,y\ :\ \pi(x_a,x_b)=\ell} \mathrm{P}(X_a = x_a, X_b = x_b)\,,$$

which is also equal to the probability $\mathrm{P}\big((X_a,X_b)\in\mathcal{R}^\ell\big)$ that a randomly drawn input belongs to $\mathcal{R}^\ell$.

To understand $c_f^m(\mathbb{D})$ it helps to realize the connection between protocols and coding theory: rather than sending information between Alice and Bob, one may think of sending the leafs over a channel by using a codebook. In this analogy, each leaf corresponds to a code and the path from the root of the protocol tree to every internal node at depth t corresponds to code prefix of length t. Furthermore, the probability of encountering the leaf is $P(L_\pi = t)$ and the depth of the protocol tree for every input $(x_a,x_b)\in\mathcal{R}^\ell$ is equal to the length of the code required to send the associated symbol.

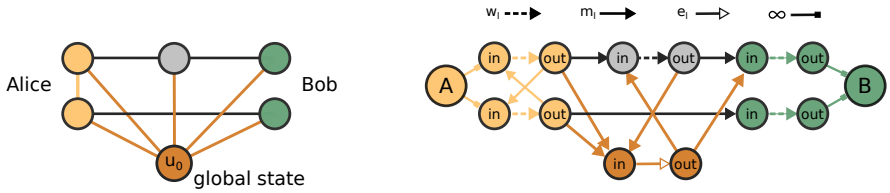
From the above it follows that the act of designing a protocol with minimal $c_f^m(\pi)$ is equivalent to finding a tree with minimum expected path length from the root to the leafs, which in turn is equivalent to minimizing the length of the expected code length for a categorical distribution L_π. Therefore, based on Shannon's source coding theorem we have that

$$\min_\pi \mathrm{H}_s(L_\pi) \le c_f^m(\mathbb{D}) \le \min_\pi \mathrm{H}_s(L_\pi) + 1,$$

matching the lemma statement. □

C Deferred proofs

C.1 Proof of Lemma 2.1



(a) Example partitioning *(b) Maximum flow reduction*

Figure 6: An example of the reduction employed in the proof of Lemma 2.1. The yellow and green subgraphs correspond respectively to G_a and G_b. The global state (external memory) is shown in orange. Each edge is annotated based on its capacity in the maximum flow reduction.

The number of symbols that can be transmitted from Alice to Bob in layer ℓ is bounded by the maximum flow of the following *multi-source multi-sink maximum flow problem with node capacities*:

- The nodes $\mathcal{V}_a$ are the senders and the nodes $\mathcal{V}_b$ are the sinks.

- Each edge has capacity m_ℓ.
- Each node in $\mathcal{V}$ has capacity w_ℓ, whereas v_0 has capacity g_ℓ.

This problem can be reduced to a simple maximum flow problem (single source single-sink without node capacities) in three steps:

1. All nodes in $\mathcal{V}_a$ (resp. $\mathcal{V}_b$) are connected to a new node A (resp. B) with edges of infinite capacity.
2. Each node v_i (with the exception of A, B and v_0) is split into two nodes in_i and out_i connected by an edge of capacity w_ℓ. Incoming edges to v_i are connected to in_i and outgoing edges are connected to out_i.
3. The same splitting procedure is performed for node v_0, but now the internal edge has capacity g_ℓ.

Consider the transformed flow network as shown in Figure 6b. By the max-flow min-cut theorem, the maximum value of the flow is equal to the minimum capacity over all cuts that separate $\mathcal{V}_a \cup A$ from $\mathcal{V}_b \cup B$. The latter however can always be bounded by $\text{cut}(A, B) + g_\ell$. The first term of this equation gives the weight of the smallest cut separating A and B in the reduced graph, excluding those (orange) edges that touch v_0: since the edges from A to $\mathcal{V}_a$ have infinite capacity (resp. from B to $\mathcal{V}_b$), every such cut also separates $\mathcal{V}_a$ and $\mathcal{V}_b$. Notice also that every path from A to B includes at least one internal edge of capacity w_ℓ and one normal edge of capacity m_ℓ. Combining the previous observations one finds that $\text{cut}(A, B) \leq \text{cut}(\mathcal{V}_a, \mathcal{V}_b) \min\{w_\ell, m_\ell\}$, where $\text{cut}(\mathcal{V}_a, \mathcal{V}_b)$ is the size of the smallest cut that separates $\mathcal{V}_a$ and $\mathcal{V}_b$ on G (the undirected and unweighted graph prior to the reduction). The internal edge capacity of v_0 in accounted by term g_ℓ. The final bound is obtained by summing the cuts over all d layers.

C.2 Proof of Theorem 3.1

The proof consists of two main steps. First, the number of monochromatic rectangles of f_{isom} will be controlled using the number of graph isomorphism classes in $\mathcal{X}$. Then, invoking Corollary B.1 will result in a bound for $c^{both}_{f_{\text{isom}}}$. Second, the identified lower bound will be translated to a bound regarding $c^{one}_{f_{\text{isom}}}$ based on Lemma D.1.

There are $2^{\binom{v}{2}}$ labeled graphs on v nodes (i.e., counting orderings), the overwhelming majority of which are connected. The number of connected labeled graphs on v nodes is

$$|\mathcal{X}_a| = |\mathcal{X}_b| = 2^{\binom{v}{2}} \left(1 - \frac{2v}{2^v} + o\left(\frac{1}{2^v}\right)\right) = 2^{\binom{v}{2}} \left(1 - O\left(\frac{v}{2^v}\right)\right),$$

which, for sufficiently large v, is very close to $2^{\binom{v}{2}}$ [Flajolet and Sedgewick, 2009, p. 138]. Specifically, one may write

$$\begin{aligned}
\log_2 |\mathcal{X}_a| = \log_2 |\mathcal{X}_b| &= \log_2 \left(2^{\binom{v}{2}} \left(1 - O\left(\frac{v}{2^v}\right)\right)\right) \\
&= \binom{v}{2} \log_2 2 + 2 \log_2 \left(1 - O\left(\frac{v}{2^v}\right)\right) \\
&\geq \frac{v(v-1)}{2} - 2\, O\left(\frac{v}{2^v}\right) \qquad (\log(1-x) \geq -O(1)x \text{ for } x \in [0,1]) \\
&= \frac{v(v-1)}{2} + o(1)
\end{aligned}$$

and, similarly, $\log_2 |\mathcal{X}_a| = \log_2 |\mathcal{X}_b| \leq \frac{v(v-1)}{2}$. The number of permutations on v nodes is $v!$, which implies that the number $g(v)$ of isomorphism classes of v-node graphs is bounded by

$$\begin{aligned}
\log_2 g(v) &\geq \log_2 \left(\frac{|\mathcal{X}_a|}{v!} \right) && (4)\\
&= \frac{v(v-1)}{2} - \log_2 (v!) + o(1) && (5)\\
&\geq \frac{v(v-1)}{2} - v \log_2 \left(\frac{v}{e} \right) - \log_2 \left(\sqrt{ve^2} \right) + o(1) && (\text{since } x! \leq \sqrt{xe^2}\,(x/e)^x)\\
&= \frac{v^2}{2} - v \log_2 \left(\frac{v\sqrt{2}}{e} \right) - \log_2 \left(\sqrt{ve^2} \right) + o(1) && (6)
\end{aligned}$$

By construction, $\mathcal{X}$ contains at least $g(v)(1+g(v))/2$ classes. To obtain this bound, one assumes that there do not exist any classes that differ only w.r.t. G_c and then notes that each unique class of $\mathcal{X}$ may be build either by gluing two distinct or identical classes on v nodes (corresponding to graphs in $\mathcal{X}_a$ and $\mathcal{X}_b$). The bound then follows by counting all pairs of elements (there are $g(v)$ of those) with repetitions (e.g., for $\{a, b, c\}$ the set of possible pairs are $\{(aa), (ab), (ac), (bb), (bc), (cc)\}$).

The number of monochromatic rectangles of f_{isom} is at least the number of classes and thus Corollary B.1 asserts:

$$\begin{aligned}
c^{both}_{f_{\text{isom}}} \log_2 s &= \log_2 \left(\left\{ \begin{array}{c} \text{minimum number of} \\ \text{monochromatic} \\ \text{rectangles} \end{array} \right\} \right)\\
&\geq \log_2 \left(\frac{g(v)(g(v)+1)}{2} \right)\\
&= 2 \log_2 g(v) + \log_2 \left(1 + \frac{1}{g(v)} \right) - 1 \geq 2 \log_2 g(v) - 1 && (7)
\end{aligned}$$

Substituting (6) into (7) gives:

$$\begin{aligned}
c^{both}_{f_{\text{isom}}} \log_2 s &\geq v^2 - 2v \log_2 \left(\frac{v\sqrt{2}}{e} \right) - 2 \log_2 \left(\sqrt{ve^2} \right) - 1 + o(1)\\
&= v^2 - 2v \log_2 \left(\frac{v\sqrt{2}}{e} \right) - \log_2 \left(2ve^2 \right) + o(1)
\end{aligned}$$

A bound on $c^{one}_{f_{\text{isom}}}$ can be derived with the help of Lemma D.1:

$$\begin{aligned}
c^{one}_{f_{\text{isom}}} \log_2 s &\geq c^{both}_{f_{\text{isom}}} \log_2 s - \max_{G_b, G_c} \log_s \left(|\{ f(G_a, G_b, G_c) \,:\, G_a \in \mathcal{X}_a \}| \right) \log_2 s\\
&= 2 \log_2 g(v) - 1 - \log_2 g(v)\\
&\geq \frac{v^2}{2} - v \log_2 \left(\frac{v\sqrt{2}}{e} \right) - \log_2 \left(2e\sqrt{v} \right) + o(1).
\end{aligned}$$

This proof is concluded by factoring $c^{one}_{f_{\text{isom}}}$ as a function of $c^{both}_{f_{\text{isom}}}$.

C.3 Proof of Theorem 3.2

I will begin by proving a more general result. Specifically, it will be shown that the expected communication complexity is directly bounded by the entropy of the isomorphism class of a graph sampled from $\mathbb{G}$.

Lemma C.1. *The expected number of symbols that Alice and Bob need to exchange to jointly compute the isomorphism class $f_{isom}(G)$ of a graph sampled from $G = (G_a, G_b, G_c) \sim \mathbb{G}$ is at least*

$$c^{both}_{f_{isom}}(\mathbb{G}) \geq \min_{G_c} \mathrm{H}_s(f_{isom}(G)|G_c)\,.$$

Proof. The first step is to condition the expected communication complexity on G_c:

$$\begin{aligned}
c_{f_{\mathrm{isom}}}(\mathbb{G}) &= \min_{\pi} \mathrm{E}_{\mathbb{G}}[c_{f_{\mathrm{isom}}}(\pi)] \\
&= \min_{\pi} \sum_{G_c} \mathrm{P}(G_c)\, \mathrm{E}_{\mathbb{G}}[c_{f_{\mathrm{isom}}}(\pi)|G_c] && \text{(due to the law of total expectation)} \\
&= \min_{\pi} \sum_{G_c} \mathrm{P}(G_c)\, \mathrm{E}_{\mathbb{G}}[c_{f_c}(\pi)] && \text{(by the definition } f_c(\cdot,\cdot) \coloneqq f_{\mathrm{isom}}(\cdot,\cdot,G_c)) \\
&\geq \sum_{G_c} \mathrm{P}(G_c) \min_{\pi} \mathrm{E}_{\mathbb{G}}[c_{f_c}(\pi)] \geq \min_{G_c} c_{f_c}(\mathbb{G}).
\end{aligned}$$

Denote by $\mathcal{L}_\pi$ the set of leafs of a protocol π that computes f_c and by L_π the random variable induced by the distribution $\mathbb{G}$ (for brevity, the conditioning on G_c remains implicit in the following). We have that

$$\mathrm{H}_s(L_\pi) = \sum_{\ell \in \mathcal{L}_\pi} \mathrm{P}(L_\pi = \ell) \log_s \left(\frac{1}{\mathrm{P}(L_\pi = \ell)} \right). \tag{8}$$

Upon closer consideration, there are $|\mathcal{Y}|$ types of leafs such that $\mathcal{L}_\pi = \bigcup_{y=1}^{|\mathcal{Y}|} \mathcal{L}_{\pi,y}$, with each subset $\mathcal{L}^l_\pi$ containing all leafs for which the protocol outputs the graph isomorphism class y. From Lemma D.2 and because $\mathcal{L}_{\pi,1}, \ldots, \mathcal{L}_{\pi,|\mathcal{Y}|}$ form a partitioning of $\mathcal{L}_\pi$, we may write:

$$\mathrm{H}_s(L_\pi) \geq \sum_{y=1}^{|\mathcal{Y}|} \mathrm{P}(L_\pi \in \mathcal{L}_{\pi,y}) \log_s \left(\frac{1}{\mathrm{P}(L_\pi \in \mathcal{L}_{\pi,y})} \right).$$

The term $\mathrm{P}(L_\pi \in \mathcal{L}_{\pi,y})$ seen above corresponds to the probability that class y will appear in our sample:

$$\mathrm{P}(L_\pi \in \mathcal{L}_{\pi,y}) = \mathrm{P}(f(G_a, G_b, G_c) = y)$$

therefore, $\min_\pi \mathrm{H}_s(L_\pi) \geq \mathrm{H}_s(f(G)|G_c)$ and the claim follows. □

Coming back to the setting of the main theorem, denote by $k_y = |\mathcal{E}_a| + |\mathcal{E}_b|$ the number of edges of the graphs in class y (disregarding the edges $\mathcal{E}_c$). For every G_c, we have that

$$\mathrm{P}(f_{\mathrm{isom}}(G) = y \,|\, G_c) = i_c(v)\, p^{k_y} (1-p)^{2\binom{v}{2} - k_y} = i_c(v)\, p^{k_y} (1-p)^{v(v-1) - k_y}.$$

Term $i_c(v)$ corresponds to the size of the corresponding isomorphism class. Specifically, when p is not too small and $\mathrm{cut}(\mathcal{V}_a, \mathcal{V} \setminus \mathcal{V}_a) = \mathrm{cut}(\mathcal{V}_b, \mathcal{V} \setminus \mathcal{V}_b) = 1$, it can be inferred that each isomorphism class in the universe contains at most $2(v!)^2$ labeled graphs. The remaining $n! - 2(v!)^2$ permutations yield isomorphic graphs with cut larger than one.

Claim C.1. *For any $\delta > 0$, $cut(\mathcal{V}_a, \mathcal{V} \setminus \mathcal{V}_a) = cut(\mathcal{V}_b, \mathcal{V} \setminus \mathcal{V}_b) = 1$, and $p \geq (\delta + \log_v)/v$, we have $i_c(v) \leq 2(v!)^2$ with probability at least $e^{-2e^{-\delta}} + o(1)$.*

Proof. To see this consider a labeled graph $G \in \mathcal{X}$ and let $G' = (\mathcal{V}', \mathcal{E}')$ be a second labeled graph that is isomorphic to G, induced by a the label permutation $\mathcal{V}' = (\Pi(u) : u \in \mathcal{V})$. I claim that, if there exist $v_i, v_j \in \mathcal{V}_a$ for which $\Pi(v_i) \in \mathcal{V}_a$ and $\Pi(v_j) \in \mathcal{V}_b$, then $G' \notin \mathcal{X}$ (and the same holds if there exist $v_i, v_j \in \mathcal{V}_b$ for which $\Pi(v_i) \in \mathcal{V}_b$ and $\Pi(v_j) \in \mathcal{V}_b$).

The claim is proven by contradiction: suppose (for now) that G_a and G_b are connected. Then, for every set $\mathcal{S}$ of cardinality v that is a strict subset of *both* $\mathcal{V}_a$ and $\mathcal{V}_b$ ($\mathcal{S}$ corresponds to the nodes with labels $(1, \cdots, v)$ in G') the cut between $\mathcal{S}$ and its complement must be $\text{cut}(\mathcal{S}, \mathcal{V} \setminus \mathcal{S}) = \sum_{v_i, v_j} \{v_i \in \mathcal{S} \text{ and } v_j \notin \mathcal{S}\} = \sum_{v_i, v_j} \{v_i \in \mathcal{S} \text{ and } v_j \in (\mathcal{V}_a \setminus \mathcal{S})\} + \sum_{v_i, v_j} \{v_i \in \mathcal{S} \text{ and } v_j \in (\mathcal{V}_b \setminus \mathcal{S})\} \geq 1+1$. The latter, however, is impossible as we have assumed that $\forall G' \in \mathcal{X}$, we must have $\text{cut}(\mathcal{V}'_a, \mathcal{V}' \setminus \mathcal{V}'_a) = \text{cut}(\mathcal{V}'_b, \mathcal{V}' \setminus \mathcal{V}'_b) = 1$. Therefore, the only valid permutations Π are those that abide to either (i) if $v_i \in \mathcal{V}_a \to \Pi(v_i) \in \mathcal{V}_a$ and if $v_i \in \mathcal{V}_b \to \Pi(v_i) \in \mathcal{V}_b$ (there are $(v!)^2$ such permutations), or (ii) if $v_i \in \mathcal{V}_a \to \Pi(v_i) \in \mathcal{V}_b$ and if $v_i \in \mathcal{V}_a \to \Pi(v_i) \in \mathcal{V}_a$ (there are $(v!)^2$ such permutations).

In the studied distribution, there is a non-zero probability that a disconnected graph appears. However, the probability is exponentially small when $p > \log v / v$. It is well known (see e.g., Theorem 4.1 by Frieze and Karoński [2016]) that, for any $\delta > 0$ and $p = \frac{\delta + \log v}{v}$, a random graph on v nodes is connected with probability

$$\text{P}(G_a \text{ is connected}) = \text{P}(G_b \text{ is connected}) = e^{-e^{-\delta}} + o(1)$$

and, by independence, $\text{P}(G \text{ is connected}) = e^{-2e^{-\delta}} + o(1)$. □

Based on the above observation, the conditional entropy of $f(G)$ can be rewritten as

$$\begin{aligned}
\text{H}_2(f_{\text{isom}}(G)|G_c) &= \sum_{y \in \mathcal{Y}} \text{P}(f_{\text{isom}}(G) = y|G_c) \log_2 \left(\frac{1}{\text{P}(f_{\text{isom}}(G) = y|G_c)} \right) \\
&\geq \sum_{k=0}^{v(v-1)} \frac{\binom{v(v-1)}{k}}{i_c(v)} i_c(v)\, p^k (1-p)^{v(v-1)-k_y} \log_2 \left(\frac{1}{i_c(v)\, p^k (1-p)^{v(v-1)-k}} \right) \\
&= \sum_{k=0}^{v(v-1)} \binom{v(v-1)}{k} p^k (1-p)^{v(v-1)-k} \left(-\log_2 i_c(v) + v(v-1) \log_2 \left(\frac{1}{1-p} \right) + k \log_2 \left(\frac{1-p}{p} \right) \right) \\
&= \log_2 \left(\frac{1-p}{p} \right) \left(\sum_{k=0}^{v(v-1)} \binom{v(v-1)}{k} p^k (1-p)^{v(v-1)-k} k \right) + v(v-1) \log_2 \left(\frac{1}{1-p} \right) - \log_2 i_c(v) \\
&\geq \log_2 \left(\frac{1-p}{p} \right) \left(\sum_{k=0}^{v(v-1)} \binom{v(v-1)}{k} p^k (1-p)^{v(v-1)-k} k \right) + v(v-1) \log_2 \left(\frac{1}{1-p} \right) - \log_2 i_c(v)
\end{aligned}$$

Let B be a binomial random variable with parameters $v(v-1)$ and p. The summation term is equivalent to the expectation of B:

$$\sum_{m=0}^{v(v-1)} \binom{v(v-1)}{k} p^k (1-p)^{v(v-1)-k} k = \text{E}[B] = v(v-1)p$$

and, therefore,

$$\begin{aligned}
\mathrm{H}_2(L_\pi) &\geq \log_2\left(\frac{1-p}{p}\right)v(v-1)p + v(v-1)\log_2\left(\frac{1}{1-p}\right) - \log_2 i_c(v) \\
&= v(v-1)\mathrm{H}_2(p) - \log_2 i_c(v) \qquad \left(\text{by definition } \mathrm{H}_2(p) = \log_2\left(\tfrac{1-p}{p}\right)p + \log_2\left(\tfrac{1}{1-p}\right)\right) \\
&= v(v-1)\mathrm{H}_2(p) - 2\log_2 v! - 1 \qquad (\text{see Claim C.1 } i_c(v) \leq 2(v!)^2) \\
&\geq v(v-1)\mathrm{H}_2(p) - 2\left(v\log_2\left(\frac{v}{e}\right) + \frac{1}{2}\log_2\left(ve^2\right)\right) - 1 \qquad (\text{since } x! \leq \sqrt{xe^2}\,(x/e)^x) \\
&= v^2\,\mathrm{H}_2(p) - v\left(2\log_2\left(\frac{v}{e}\right) + \mathrm{H}_2(p)\right) - \log_2\left(2ve^2\right)
\end{aligned}$$

Invoking Lemma C.1, one obtains:

$$\begin{aligned}
c^{both}_{f_{\mathrm{isom}}}(\mathbb{B}_{v,p}) &\geq \min_{G_c} \frac{\mathrm{H}_2(f_{\mathrm{isom}}(G)|G_c)}{\log_2 s} \\
&\geq v^2\,\mathrm{H}_s(p) - v\left(2\log_s\left(\frac{v}{e}\right) + \mathrm{H}_s(p)\right) - \log_s\left(2ve^2\right) = \beta
\end{aligned} \tag{9}$$

Then Lemma D.1 gives:

$$\begin{aligned}
c^{one}_{f_{\mathrm{isom}}}(\mathbb{B}_{v,p})\log_2 s &\geq c^{both}_{f_{\mathrm{isom}}}(\mathbb{B}_{v,p})\log_2 s - \max_{G_b,G_c}\log_s\left(|\{f_{\mathrm{isom}}(G_a,G_b,G_c)\,:\,G_a\in\mathcal{X}_a\}|\right)\log_2 s \\
&= c^{both}_{f_{\mathrm{isom}}}(\mathbb{B}_{v,p})\log_2 s - \log_2\left(\frac{|\mathcal{X}_a|}{v!}\right) \\
&= c^{both}_{f_{\mathrm{isom}}}(\mathbb{B}_{v,p})\log_2 s - \frac{v(v-1)}{2} + \log_2(v!) \\
&\geq v(v-1)\left(\mathrm{H}_2(p) - \frac{1}{2}\right) - \left(v\log_2\left(\frac{v}{e}\right) + \frac{1}{2}\log_2\left(ve^2\right)\right) - 1 \\
&= v^2\mathrm{H}_2(p) - \frac{v}{2}\left(2\log_2\left(\frac{v}{e}\right) + \mathrm{H}_2(p)\right) - \frac{1}{2}\log_2\left(2ve^2\right) - \frac{v^2 - v + v\mathrm{H}_2(p) + 1}{2} \\
&= \frac{\beta\log_2 s - v^2 + v(1-\mathrm{H}_2(p)) - 1}{2}
\end{aligned}$$

implying $c^{one}_{f_{\mathrm{isom}}}(\mathbb{B}_{v,p}) \geq \frac{\beta}{2} - \frac{v^2 - v(1-\mathrm{H}_2(p))+1}{2\log_2 s}$.

C.4 Proof of Theorem 3.3

According to Otter [1948], the number of unlabeled trees on v nodes grows like

$$t(v) \sim c\,\alpha^v\,v^{-5/2},$$

where the values c and α known to be approximately 0.5349496 and 2.9557652 (sequence A051491 in the OEIS). Moreover, it was shown in the proof of Theorem 3.1, the number of monochromatic rectangles is at least $(t(v)+1)\,t(v)/2$.

Corollary B.1 then implies

$$\begin{aligned}
c^{both}_{f_{\mathrm{isom}}} &\geq \log_s\left(\frac{(t(v)+1)\,t(v)}{2}\right) \\
&\geq \log_s\left(\frac{t(v)^2}{2}\right) \\
&\sim 2\log_s\left(\alpha^v\,v^{-5/2}\right) - \log_s\left(c^2/2\right) \sim 2v\log_s\alpha - 5\log_s v + \log_s 7 = \beta
\end{aligned}$$

Further, from Lemma D.1 one can derive:

$$\begin{aligned} c^{one}_{f_{\text{isom}}} &\geq c^{both}_{f_{\text{isom}}} - \max_{G_b,G_c} \log_s \left(|\{f(G_a,G_b,G_c) \,:\, G_a \in \mathcal{X}_a\}|\right) \\ &= c^{both}_{f_{\text{isom}}} - \log_s t(v) \\ &\sim \log_s \left(\alpha^v \, v^{-5/2}\right) - \log_s (c/2) \sim v \log_s \alpha - \frac{5}{2}\log_s v + \frac{1}{2}\log_s 14 \end{aligned}$$

implying $c^{one}_{f_{\text{isom}}} \geq \frac{\beta + \log_s 2}{2}$.

Let me now consider the case that G is sampled uniformly at random from the set of all trees in $\mathcal{X}$. It is a consequence of Lemma C.1 that when the graph $(G_a, G_b, G_c) \sim \mathbb{G}$ (conditioned on G_c) is sampled uniformly at random from a collection of isomorphism classes, the expected communication complexity is at least

$$c^{both}_{f_{\text{isom}}}(\mathbb{T}_v) \geq \min_{G_c} \log_s |\{f(G_a,G_b,G_c) \;:\; G \in \mathcal{X} \text{ s.t. } G_c\}|.$$

This can be seed to be identical to the worst-case bound encountered above. The derivation thus can be carried out analogously (and the same holds for $c^{one}_{f_{\text{isom}}}(\mathbb{T}_v)$ by Lemma D.1).

C.5 Proof of Proposition 3.1

In general terms, the impossibility statement comes as a consequence of the definition of communication complexity: if the number of required exchanged symbols exceeds the symbols the learner can exchange (i.e., its communication capacity) then the latter will not be able to identify exactly f_{isom}. The factor of 2 comes in because N can exchange $2c_N$ symbols overall (c_N from Alice to Bob and c_N from Bob to Alice).

The specifics depend on the appropriate definition:

Majority-voting necessitates $|\mathcal{M}_G| \geq \mu$, meaning that when $|\mathcal{M}_G| \geq \mu > n - 2v$ at least one of the two parties should have gathered sufficient information to determine $f_{\text{isom}}(G)$ at the final layer. Therefore, m should be "one". With consensus on the other hand, we have that $|\mathcal{M}_G| \geq n - \mu > n - v$ which implies that both parties need to know the class.

The worst-case communication complexity definition guarantees that there exists at least one input for which the required number of symbols is $c^{(m)}_{f_{\text{isom}}}$. Thus, since $\mathbb{D}$ is densely supported on $\mathcal{X}$, the impossibility must occur with strictly positive probability. The impossibility also applies to any universe $\mathcal{X}'$ that is a strict superset of $\mathcal{X}$. This can be easily derived by conditioning on $\mathcal{X} \subset \mathcal{X}'$ (which can only decrease the communication complexity) and repeating the analysis identically.

Finally, to comprehend the implications of the expected complexity bound, fix π^* to be the protocol that achieves minimal expected length and let β_m be an upper bound of π^* length over all inputs. By Lemma D.3, for any $\delta \in [0,1]$ one has

$$\mathrm{P}\big(\|\pi^*(G)\|_m > 2\delta \, c^m_{f_{\text{isom}}}(\mathbb{G})\big) \geq \frac{1-\delta}{(c^m_{f_{\text{isom}}}(\mathbb{G})/\beta_m) - \delta}.$$

and the protocol length is at most

$$\beta_{\text{one}} \leq \log_s \left(\min\{|\mathcal{X}_a|, |\mathcal{X}_b|\}\right) \quad \text{and} \quad \beta_{\text{both}} \leq \log_s \left(|\mathcal{X}_a| + |\mathcal{X}_b|\right),$$

which corresponds to Alice sending the index of G_a in $\mathcal{X}_a$ (resp. for Bob) [Rao and Yehudayoff, 2020].

D Helpful lemmata

Lemma D.1. *In the universe considered in Section 3, the following hold for any $\mathbb{D}$:*

$$c_f^{one} \geq c_f^{both} - \max_{G_b, G_c} \log_s \left(|\{f(G_a, G_b, G_c) \,:\, G_a \in \mathcal{X}_a\}|\right)$$
$$c_f^{one}(\mathbb{D}) \geq c_f^{both}(\mathbb{D}) - \max_{G_b, G_c} \log_s \left(|\{f(G_a, G_b, G_c) \,:\, G_a \in \mathcal{X}_a\}|\right).$$

Proof. Consider the setting of c_f^{one}, where for a successful termination it suffices for one party to computing the output of f. Suppose w.l.o.g., that this party is Alice. In particular, Alice determines class $y = f(G_a, G_b, G_c)$ based on a protocol π of minimal length. In this setting, Bob does not know y but he is aware of $\mathcal{X}_b^\ell$ (and G_c), where ℓ is the leaf of the protocol tree at input (G_a, G_b, G_c). Therefore, both parties know that the class must belong to the set $\{f(G_a, G_b, G_c) \,:\, G_b \in \mathcal{X}_b^\ell \text{ and } G_a \in \mathcal{X}_a\}$. It is a consequence that there exists a protocol π' of length

$$\|\pi'(G_a, G_b, G_c)\|_{both} \leq \|\pi(G_a, G_b, G_c)\|_{one} + \log_s |\{f(G_a, G_b, G_c) \,:\, G_b \in \mathcal{X}_b^\ell \text{ and } G_a \in \mathcal{X}_a\}|$$

that results in both parties knowing y. The protocol π' entails first simulating π and then Alice sending to Bob the index of y in the set of feasible classes. Moreover, since f corresponds to the graph isomorphism problem, for Alice to know y, she must also know the isomorphism class of Bob. Therefore, the feasible set of classes contains only the feasible subgraph isomorphism classes of G_a, which are at most

$$|\{f(G_a, G_b, G_c) \,:\, G_b \in \mathcal{X}_b^\ell \text{ and } G_a \in \mathcal{X}_a\}| \leq \max_{G_b, G_c} |\{f(G_a, G_b, G_c) \,:\, G_a \in \mathcal{X}_a\}|$$

The claimed inequalities then follow by the optimality of the protocol π and since the same construction can be repeated for every input. □

Lemma D.2. *Let X be a categorical random variable with sample space $\mathcal{X}$. For any partitioning $\mathcal{X} = \mathcal{A}_1, \cdots, \mathcal{A}_k$ we have that*

$$\mathrm{H}_s(X) \geq \sum_{i=1}^{k} \mathrm{P}(X \in \mathcal{A}_i) \log_s \left(\frac{1}{\mathrm{P}(X \in \mathcal{A}_i)}\right)$$

Proof. The proof is elementary. It relies on the inequality $\mathrm{P}(X = x) \leq \mathrm{P}(X \in \mathcal{A}_i)$ that holds for all $x \in \mathcal{A}_i$:

$$\begin{aligned}
\mathrm{H}_2(X) &= \sum_{i=1}^{k} \mathrm{P}(X \in \mathcal{A}_i) \sum_{x \in \mathcal{A}_i} \frac{\mathrm{P}(X = x)}{\mathrm{P}(X \in \mathcal{A}_i)} \log_s \left(\frac{1}{\mathrm{P}(X = x)}\right) \\
&\geq \sum_{i=1}^{k} \mathrm{P}(X \in \mathcal{A}_i) \min_{x \in \mathcal{A}_i} \log_s \left(\frac{1}{\mathrm{P}(X = x)}\right) \\
&= \sum_{i=1}^{k} \mathrm{P}(X \in \mathcal{A}_i) \log_s \left(\frac{1}{\max_{x \in \mathcal{A}_i} \mathrm{P}(X = x)}\right) \\
&\geq \sum_{i=1}^{k} \mathrm{P}(X \in \mathcal{A}_i) \log_s \left(\frac{1}{\sum_{x \in \mathcal{A}_i} \mathrm{P}(X = x)}\right) = \sum_{i=1}^{k} \mathrm{P}(X \in \mathcal{A}_i) \log_s \left(\frac{1}{\mathrm{P}(X \in \mathcal{A}_i)}\right),
\end{aligned}$$

as claimed. □

Lemma D.3. *For any random variable $X \leq \beta$ and $\delta \in [0,1]$ we have $\mathrm{P}(X > \delta\,\mathrm{E}[X]) \geq \frac{1-\delta}{r-\delta}$, where $r = \beta/\mathrm{E}[X]$.*

Proof. For any $t \leq \beta$,

$$\mathrm{E}[X] = \sum_{x \leq t} \mathrm{P}(X)\, x + \sum_{x > t} \mathrm{P}(X)\, x \leq \mathrm{P}(X \leq t)\, t + \mathrm{P}(X > t)\, \beta = (1 - \mathrm{P}(X > t))t + \mathrm{P}(X > t)\, \beta$$

or, equivalently, $\mathrm{P}(X > t) \geq (\mathrm{E}[X] - t)/(\beta - t)$. The final inequality is obtained by setting $t = \delta \mathrm{E}[X]$. $\square$