STUDYING AND IMPROVING GRAPH NEURAL NETWORK-BASED MOTIF ESTIMATION

A PREPRINT

Pedro C. Vieira *
Department of Computer Science
Faculty of Science of the University of Porto
Porto, Portugal
pedrocvieira@fc.up.pt

Miguel E. P. Silva
Department of Computer Science
Faculty of Science of the University of Porto
Porto, Portugal
mepsilva@fc.up.pt

Pedro Manuel Pinto Ribeiro
Department of Computer Science
Faculty of Science of the University of Porto
CRACS/INESC-TEC
Porto, Portugal
pribeiro@fc.up.pt

ABSTRACT

Graph Neural Networks (GNNs) are a predominant method for graph representation learning. However, beyond subgraph frequency estimation, their application to network motif significance-profile (SP) prediction remains under-explored, with no established benchmarks in the literature. We propose to address this problem, framing SP estimation as a task independent of subgraph frequency estimation. Our approach shifts from frequency counting to direct SP estimation and modulates the problem as multitarget regression. The reformulation is optimised for interpretability, stability and scalability on large graphs. We validate our method using a large synthetic dataset and further test it on real-world graphs. Our experiments reveal that 1-WL limited models struggle to make precise estimations of SPs. However, they can generalise to approximate the graph generation processes of networks by comparing their predicted SP with the ones originating from synthetic generators. This first study on GNN-based motif estimation also hints at how using direct SP estimation can help go past the theoretical limitations that motif estimation faces when performed through subgraph counting.

Keywords Graph Neural Networks · Motifs · Significance-Profile

1 Introduction

In this work, "graph" and "network" are used interchangeably. A graph is deemed a network motif [Milo et al., 2004b] when it occurs more frequently than random chance would suggest, serving as a powerful tool for analysing complex networks. Recognising significant substructures helps elucidate a graph's underlying organizational principles, advancing both theoretical and practical understanding, especially in biology. For example, the feed-forward loop is a key functional pattern in gene regulation networks [Mangan and Alon, 2003]. It has also been discovered that motifs enable efficient communication and fault-tolerance across transcriptional networks [Roy et al., 2020].

Discovering a motif entails counting the number of occurrences of the desired structure, both in the network in study and in a set of control networks (networks that preserve key properties of the original) to understand its significance. This process is at least NP-complete, as it includes the subtask of determining whether a subgraph exists within a

*Correspondence to: pedrocvieira@fc.up.pt

larger network (the subgraph isomorphism problem). Although there are methods to perform an analysis based on motifs, [Ribeiro et al., 2021], they have a high temporal complexity, rendering them intractable for very large networks.

A good alternative that can help overcome these limitations is to use machine learning, specifically Graph Neural Networks (GNNs), to perform motif estimation. However, as far as we are aware, there are few methods that attempt this route. Existing methods [Zhang et al., 2020, Oliver et al., 2022, Ying et al., 2019, Chen and Ying, 2023, Chen et al., 2023, Sheng et al., 2024, Dareddy et al., 2019, Lee et al., 2018, Fu et al., 2023] typically adopt formulations that are too complex and lead to the omission of important aspects in traditional motif estimation, deviating significantly from the original concept of motif [Milo et al., 2004b, Milo et al., 2004a]. More importantly, we postulate that using GNNs just as a subgraph frequency estimation method to then use does estimations for calculating motif scores fundamentally misses the opportunity to exploit GNNs for more powerful and direct formulations. We identify that using GNNs as subgraph frequency estimators limits the task of motif estimation to the expressivity that a model can achieve at subgraph counting. In addition to this, we find that other motif estimation methods [Zhang et al., 2020, Oliver et al., 2022] only provide an indication regarding which subgraphs are motifs, whereas real-world scenarios require more detailed information on the significance of the found motifs, information regarding which subgraphs are not motifs and sometimes information regarding anti-motifs.

In contrast, **our work** is aimed at addressing these shortcomings by making small but fundamental changes. Specifically, we propose (1) estimation of motif scores that are interpretable and can be used in comparisons between networks and other downstream tasks; (2) clear statistical relevance with connections to traditional estimation based on a null model; (3) control over what graphs will be evaluated as candidate motifs; (4) complete description of the chosen graphs, allowing a detailed analysis of how a graph is placed in the space of motifs; (5) stability and time complexity improvements when estimating out-of-distribution with respect to the network size; and (6) allowing the formulation to possibly bend expressivity results for motif estimation based on subgraph counting.

Key Contributions: Our key contributions can be summarised as follows: ***(1)*** We show preliminary results that the difficulty of motif discovery with Message Passing Neural Networks (MPNNs) can be manipulated through different formulations of the target variable (e.g. different concept of what is a motif). Hence, depending on the formulation used, motif estimation does not have to follow the limitations regarding subgraph counting exposed in the literature. ***(2)*** We present a formulation – multi-target regression of normalized significance-profiles – based on simple assumptions that improves the ability of MPNNs to perform motif estimation while approximating this task to traditional motif estimation; ***(3)*** We make available a large and diverse synthetic dataset in terms of both graph topology and motif significance-profile, using twenty three random network models. We also present a collection of more than 100 real-world networks and their motif significance-profile.

2 Background and Related Work

Let a graph $\mathcal{G} = (\mathbb{V}_\mathcal{G}, \mathbb{L}_\mathcal{G}, \boldsymbol{X})$ where $\mathbb{V}_\mathcal{G}$ denotes the vertex set of $\mathcal{G}$, $\mathbb{L}_\mathcal{G} \subseteq \mathbb{V}_\mathcal{G} \times \mathbb{V}_\mathcal{G}$ the edge and $\boldsymbol{X} \in \mathbb{R}^{d_1 \times d_2}$ the vertex features such that $\forall v \in \mathbb{V}_\mathcal{G}, \boldsymbol{x} \in \mathbb{R}^{d_2}$. Let all edges be undirected such $(u, v) \in \mathbb{L}_\mathcal{G} \Leftrightarrow (v, u) \in \mathbb{L}_\mathcal{G}$. Let $\mathcal{H}$ be a subgraph of $\mathcal{G}$ if and only if $\mathbb{H}_\mathcal{H} \subseteq \mathbb{V}_\mathcal{G} \wedge \mathbb{L}_\mathcal{H} \subseteq \mathbb{L}_\mathcal{G}$, such that exists an injective homomorphism given by the injective function $f : \mathbb{V}_\mathcal{H} \mapsto \mathbb{V}_\mathcal{G}$ yielding $(v, u) \in \mathbb{L}_\mathcal{H} \Rightarrow (f(v), f(u)) \in \mathbb{L}_\mathcal{G}$. If f is bijective and f^{-1} is an homomorphism (injective by construction) the relation is an isomorphism and the subgraph induced.

In order to discover motifs, we must define three steps: (1) What is the set of graphs, S_G, that we admit as candidates for motifs; (2) What method is used to count the occurrences of graphs of S_G in the graph of interest $\mathcal{G}$; (3) How is the significance of the obtained counts calculated.

2.1 Step One - How to Define the Set of Graphs Used

In this step, S_G is typically defined *a priori*. This method is the most widely used, [Milo et al., 2004b, Milo et al., 2004a, Shen-Orr et al., 2002], and in most common cases, the selection of graphs used are ones known to be important to the area of the work in question [Shen-Orr et al., 2002, Alon, 2007].

Defining S_G *a priori* is frequent for "non-machine-learning" techniques, but it is also common in machine-learning ones [Rong et al., 2020, Ying et al., 2020]. However, when using techniques based on machine-learning, it is easier to create a task that can infer structures in $\mathcal{G}$ than when using non-machine learning approaches. Hence, motif discovery can be modulated as the task of finding the best set of graphs that can be considered motifs. To achieve this, it is typically used a graph metric (not necessarily in the formal metric sense) to measure how much a motif a graph is. For example, using maximum-likelihood estimation to attribute subgraph embeddings to certain graphs or by devising loss functions that penalise agreement with control networks [Zhang et al., 2020, Oliver et al., 2022].

2.2 Step Two - How to Count Subgraphs

Non-GNN Methods. Numerous methods exist for approximating subgraph counts, eschewing GNNs or any machine learning techniques. We refer the interested reader to [Ribeiro et al., 2021] for a survey of these methods.

GNN Methods. Counting occurrences of a graph $\mathcal{G}$ in another graph $\mathcal{H}$ using GNNs was first introduced by Chen et al. [Chen et al., 2020]. This work introduced significant limitations of what substructures MPNNs can count. Subsequent works have refined MPNN-like models to be more expressive, allowing them to have guarantees of being able to count occurrences of more graphs. One branch of such models is known as node-rooted Subgraph GNNs. These will extract a k-hop neighbourhood for each node v in the graph to be studied and calculate an added feature for v. These architectures, with models as powerful as 1-WL as the backbone, are strictly more powerful than maximum powerful MPNNs but are less powerful than the 3-WL test [Frasca et al., 2022, Yan et al., 2023, Zhang et al., 2023]. Hence, they have limitations regarding the type of structures that they can count. Huang et al. [Huang et al., 2022] gives a characterisation of what substructures node-rooted Subgraph GNNs cannot count at node-level. They show that the Subgraph GNNs cannot count cycles of more than four and paths of more than three nodes.

Recently a new theoretical view of Subgraph GNNs based on the Subgraph Weisfeiler-Lehman, a new version of the WL test, has been proposed [Zhang et al., 2023]. This analysis presents a characterisation of the expressive power of all node-rooted Subgraph GNNs. They conclude that no node-rooted Subgraph GNN can be more powerful than the 2-folklore-WL (3-WL) (also noticed in [Yan et al., 2023, Frasca et al., 2022]). In that same work, the authors demonstrate that no node-rooted Subgraph GNN can achieve the maximum expressivity of their time complexity class. This result draws a limitation in the design of node-rooted Subgraph GNNs. Regarding induced subgraph counting at graph level, the subject of our work, 3-WL models can count all patterns of 3 or less nodes [Lanzinger and Barceló, 2023] and 1-WL models cannot count any pattern with 3 or more nodes.

2.3 Step Three - How is Significance Obtained

After obtaining the frequency of the structures in S_G, the next step is to evaluate their significance. Hence, it is necessary to have an idea of what would produce, with no factor other than random chance, a (control) network similar to $\mathcal{G}$ for some characteristic of interest. Let us denote as NULL a model that can achieve that goal. One example of NULL is a model that, given a graph $\mathcal{G}$, randomly switches edges while keeping the degree distribution of $\mathcal{G}$ – degree distribution is the characteristic of interest. The rewiring process is completely random and without any bias towards any predisposition [Milo et al., 2004b, Milo et al., 2004a]. A myriad of NULL models can be conceived depending on the characteristic to studied.

2.4 Motifs and Graph Neural Networks

Motif estimation, when approached through the lens of GNNs, appears to be a challenge that, to the best of our knowledge, remains largely unexplored in the existing literature. Despite using the term "motif", works like [Besta et al., 2022] do not overlap with our work. They attend to the identification of the chance of higher-order structures to appear and they can be seen as a generalisation of link prediction. We will now outline the interactions that we have identified.

Directly counting. One of the approaches that better matches direct motif estimation with GNNs involves two main steps. First, a GNN, such as a node-rooted Subgraph GNN, is used to count the occurrences of the graphs of interest, S_G, within the input graph, $\mathcal{G}$. Second, a suitable null model is selected, and T graphs are generated according to this model. The same GNN used for $\mathcal{G}$ is then applied to count the occurrences of S_G in each of the T control graphs. Additionally to generic GNNs, SPMiner [Ying et al., 2020] is an example of a method that specialises in subgraph counting.

Motifs as tool. Other works that integrate GNNs and motifs typically deviate from estimating motifs and use pre-computed ones to enhance the power of GNNs or explain decisions. Examples of this work include Motif Convolutional Networks [Lee et al., 2018], motif2vec [Dareddy et al., 2019], Motif Graph Attention Network [Sheng et al., 2024], Motif Graph Neural Network [Chen et al., 2023], Heterogeneous Motif Graph Neural Network [Yu and Gao, 2022], GNNExplainer [Ying et al., 2019] and TempME [Chen and Ying, 2023].

Learning Motifs. The works that directly address motif estimation, such as those that later used them to refine downstream tasks, include MICRO-Graph [Zhang et al., 2020] and MotiFiesta [Oliver et al., 2022].

We will ignore the methods that use motif as a tool since they reside out of the scope of the problem. The main problems of the presented works are the following: (1) either the model does not assume a null model and returns raw counts of occurrences of a general $\mathcal{H}$ in $\mathcal{G}$ (SPMiner and other frequency estimation models), or (2) the model may use a null

model to guide motif search but only returns the subgraph(s) that are considered a motif by the model, meaning it is typically not possible to query for a specific $\mathcal{H}$ (MICRO-Graph and MotiFiesta). In fact, methods in this category commonly encounter challenges in regulating the size and shape of the graph proffered as a candidate motif.

Additionally, models that return the raw count of occurrences (case (1)) can suffer from poor generalisation since the number of graph structures grows super-exponentially [Fu et al., 2023]. Hence, as the size of a graph $\mathcal{G}$ grows, the possible counts of a substructure $\mathcal{H}$ in $\mathcal{G}$ also grow super-exponentially, causing high variation between results of small and very-large graphs when estimating out-of-distribution with respect to the network size. This fact can hinder the learning process of models that aim at being agnostic of network size and topology. Furthermore, since these methods essentially perform subgraph counting, they are bounded by the limitations of subgraph counting. This means that the task of motif estimation is limited by the expressivity that a model can achieve at subgraph counting. Finally, for raw count models, since no null model is assumed, obtaining significance implies subsequent computation.

As for models falling under category (2), they typically ignore everything not branded as a motif, sometimes not even returning a motif score for the graphs regarded as such. Additionally, since the metric employed for discerning the suitable graph for a candidate motif has to be a learnable function, it can be hard to interpret its meaning in order to evaluate the strength of the motif. This can lead to the lack of clarity in the application to real-world scenarios in downstream tasks e.g. compare strength across networks. Finally, if no null model is applied, like in the case of category (1) obtaining significance implies subsequent computation.

3 Method

Hereafter, as per the first paragraph of Section 2, referencing the number of occurrences of a graph $\mathcal{H}$ within $\mathcal{G}$, denotes the induced count of $\mathcal{H}$ in $\mathcal{G}$. Furthermore, all graphs are undirected and they do not have edge features.

According to the definition of motif adopted, to understand if a graph $\mathcal{H}$ is a motif of a graph $\mathcal{G}$, we must know the number of occurrences of $\mathcal{H}$ in $\mathcal{G}$. Let us denote such count as $C(\mathcal{H}, \mathcal{G})$. Furthermore, to grasp the importance of $\mathcal{H}$ in $\mathcal{G}$, it is needed to know the count of $\mathcal{H}$ across sufficient graphs derived from a null model denoted as NULL (control graphs). The chosen null model is the one described in Section 2.3, and uses the degree distribution as the characteristic of interest that will be used in the control networks [Milo et al., 2004b, Milo et al., 2004a]. Let us denote the average count of $\mathcal{H}$ in graphs derived from NULL as $C^{\mu}(\mathcal{H}, \mathcal{G}_{\text{NULL}})$ and the standard deviation as $C^{\sigma}(\mathcal{H}, \mathcal{G}_{\text{NULL}})$. Hence, $Z(\mathcal{H}, \mathcal{G}_{\text{NULL}}) = \frac{C(\mathcal{H},\mathcal{G}) - C^{\mu}(\mathcal{H},\mathcal{G}_{\text{NULL}})}{C^{\sigma}(\mathcal{H},\mathcal{G}_{\text{NULL}})}$ denotes the standardization (Z-score) of the occurrences of a graph $\mathcal{H}$ in $\mathcal{G}$.

3.1 Our Approach

We first impose a restriction on the number and type of graphs the model will predict. We refer to the set of graphs used as Ω. The function notation $\Omega(\mathcal{G})$ gives the set of all graphs that have the same number of connected nodes as the graph $\mathcal{G}$. The restriction of the number of graphs implies that the proposed model will not be able to search if an arbitrary graph is or is not a motif. However, by having a model that has a more restricted objective, we aim to achieve higher precision in the said objective while being able to fully understand how the graphs in Ω are positioned in the motif spectrum.

Furthermore, we do not follow the approach of predicting $C(\mathcal{H}, \mathcal{G})$. Instead, we aim at directly modelling a statistical motif score like $Z(\mathcal{H}, \mathcal{G})$. This achieves full transparency on what null model is used and how it is used. Additionally, this eliminates the need to compute multiple networks based on the null model to determine significance. In fact, this step is completely skipped, gaining a lot in performance, while the result remains statistically interpretable.

Instead of modelling the learning task as predicting a single value $Z(\mathcal{H}, \mathcal{G})$ for some $\mathcal{H}$ and some $\mathcal{G}$, we model it as a multi-target regression problem in order to predict the motif score of multiple subgraphs at once. This formulation means that a model will train to predict all the graphs of Ω at the same time. Hence, this characterisation naturally allows the construction of motif profiles [Milo et al., 2004a]. Thus, we define a vector of Z-scores, $\boldsymbol{z} = [Z(\mathcal{H}_1, \mathcal{G}) \dots Z(\mathcal{H}_n, \mathcal{G})]$. Since Ω has a restricted size, one aspect that deserves careful consideration is deciding what is the size of Ω and what graphs compose it. Should the selected graphs exhibit negligible relation, an attempt to predict the Z-score concurrently for all graphs may prove harmful to the performance of the model. In this case, such an approach forces the model to incorporate distinct patterns to predict scores for each graph, thereby resulting in a sub-optimal global predictive efficacy. However, if Ω is composed of a well-thought group of graphs, allowing them to share common patterns from a learning perspective, we hypothesise that the performance of the model can improve when compared to predicting just one Z-score, due to the possibility of what is learned about a target variable be "shared" with others through weight sharing (one other advantage is the need to only train a single model instead of multiple). A good candidate for Ω

should have patterns that are interconnected with each other, either from the point of view of the Z-score distribution or from a topological one.

Building on top of what was described in the last paragraph, we focus on small graphs, in particular all connected graphs of size three and four. This is also supported by existing relevant literature [Milo et al., 2004a, Milo et al., 2004b, Shen-Orr et al., 2002, Asikainen et al., 2020, Pržulj, 2007, Ribeiro and Silva, 2013] suggesting that to understand a complex network, it is important to understand how small graphs behave. We focus on these graphs because their proximity in size should allow them to have a topological connection that translates in a connection in their Z-scores. Restricting the size of the graphs used in Ω to small ones also has the added benefit that we can get the ground-truth of motif scores for a diverse type and size of networks, allowing for a richer train dataset. Furthermore, we expect that using a set of graphs of increasing size in the number of nodes and edges gives enough interconnectivity between their patterns from both a topological and a Z-score distribution point of view to allow the model to have a strong inductive bias towards meaningful patterns, allowing for a stronger performance. For example, a graph with many size four cliques will probably have a small amount of 4-stars. For the chosen graphs it is possible to create two groups in Ω, the graphs of size three and the graphs of size four.

Finally, we normalise the Z-score, $Z(\mathcal{H}, \mathcal{G})$, across groups of graphs, according to $s_i = z_i / (\sum_{j \in \Omega(i)} z_j^2)^{1/2}$ [Milo et al., 2004a]. After normalisation, the values of $\boldsymbol{z}$ are constrained between -1 and 1, independent of network size. This will be beneficial at maintaining the predictive stability of the model when estimation out-of-distribution with respect to the network size. Additionally, this will further enhance the usage of the score for downstream tasks as it allows comparison across networks of different sizes. In fact, this normalisation keeps a wide motif spectrum, allowing to make fine distinctions between graphs. For example, a graph with score 0.8 appears more often that expected than a graph with motif score 0.6. Moreover, this normalisation imposes a mathematical interconnectivity between the Z-scores of graphs of the same group. This relationship, where the sum of squared normalised Z-scores equals 1, supports a multi-target objective and further strengthens the problem formulation by adding an additional layer of interdependence among graphs. Let us denote the normalised Z-scores as, $\boldsymbol{s}$, also known as significance-profiles. The learning task thus consists of minimising the MSE between the true and predicted significance-profiles.

3.2 On the Relation with Expressivity Through Subgraph Counting

It is expected that the expressivity regarding substructure counting to be highly related to the expressivity of discovering the significance-profile of graphs. Concretely, the problem of counting graphs is a subset of the problem of discovering significance-profiles where reducing the null model to nothing reduces the problem to graph counting.

Since P, the problem of counting graphs, is a subset of S, the problem of significance-profile estimation, it is possible to obtain instances of S that are as hard as P, easier than P and harder than P. Under the assumption that S and P function around the same set of graphs, these differences in difficulty come from the choice of null model. In the case of $S = P$, the null model should do nothing, for example, returning always 0. In the case of $S < P$, the null model could always return the counts of each subgraph in a graph $\mathcal{G}$ without modifying $\mathcal{G}$, reducing the problem to always predicting a vector of zeros. For the case of $S > P$, employing a null model that randomly returns counts for $\mathcal{G}$ should make the problem theoretically harder since the model would have to learn the random process employed to correctly construct the significance-profiles. Thus, theoretical guarantees of expressivity might not hold depending on the selected null model. For instance, a recent demonstration solved the dimensionality of the k-WL test for induced subgraph count. It is stated that to perform induced subgraph count of any pattern with k nodes we need at least dimensionality k [Lanzinger and Barceló, 2023]. Furthermore, no induced pattern with $k+1$ nodes can be counted with dimensionality k, a result not verified to non-induced counts [Lanzinger and Barceló, 2023]. However, when working with significance-profiles over graphs of size k, the relation is more complex since it depends on the null model.

Testing with 1-WL bounded models? MPNNs cannot perform induced counts of patterns of three or more nodes [Chen et al., 2020]. Nevertheless, MPNNs are not inherently incapable of counting patterns in any graphs. Rather, for a pattern $\mathcal{H}$, there exists graphs $\mathcal{G}_1, \mathcal{G}_2 \in \mathcal{G}$ such that $C(\mathcal{H}, \mathcal{G}_1) \neq C(\mathcal{H}, \mathcal{G}_2)$ and for any MPNN M under 1-WL, $M(\mathcal{G}_1) = M(\mathcal{G}_2)$. Hence, M cannot discover $C(\mathcal{H}, \mathcal{G}_1)$ and $C(\mathcal{H}, \mathcal{G}_2)$ simultaneously. However, within the 1-WL framework, MPNNs remain valuable and find practical applications in real-world scenarios. Furthermore, we did not construct any characterisation of the problem space of S regarding P for the null model used. Hence, we might have made the problem easier (or harder) than substructure counting. Thus, we will scrutinise the capability of MPNNs to address our particular challenge. Furthermore, testing with more expressive models like Subgraph GNNs is hard due to their complexity [Frasca et al., 2022].

4 Datasets

We are not interested in limit testing the power of our formulation or comparing it to theoretical tests: rather, our goal is to test our model on a large dataset with diverse topological features and motif scores. Hence, we avoid standard GNN datasets [Wu et al., 2018, Morris et al., 2020, Hu et al., 2021b, Hu et al., 2021a]. Given the difficulty of obtaining a real-world dataset spanning multiple domains while retaining the defined properties, we rely exclusively on synthetic graphs. Concretely, we use 23 synthetic generators (11 non-deterministic, 12 deterministic). We explore their graph-generating space to extract all types of topologies while limiting the graph size to avoid excessive training times. The final dataset contains 109164 graphs in the non-deterministic segment and 38400 in the deterministic ($\approx$ 250 million nodes and $\approx$ 750 million edges). Finally, for the ground-truth, we calculate $\boldsymbol{s}$ using G-Tries [Ribeiro and Silva, 2013].

Using synthetic data to train GNNs is not a new concept, but the most of the popularly used datasets typically have very small graphs (at most few hundreds of nodes) and are generated from a small set of generators, often random regular graphs and Erdős-Renyi graphs [Chen et al., 2020]. Another popular type of synthetic graph dataset for benchmarking is small handcrafted graphs to limit test GNN models [Abboud et al., 2021, Murphy et al., 2019, Balcilar et al., 2021, Wang and Zhang, 2023]. While still very limited, the only exceptions identified use some Barabási-Albert graphs, some graphs with crude community structure, trees and grids [Veličković et al., 2020, Corso et al., 2020]. In contrast, our approach employs multiple graph generators that simulate real-world phenomena, yielding a dataset with high topological diversity and a close resemblance to real data.

Our analysis of the SPs for all graphs in the synthetic data reveals that the 3-path and triangle exhibit limited motif scores, namely $\{\pm 1/\sqrt{2}, 0, 1\}$ for the 3-path and $\{\pm 1/\sqrt{2}, 0\}$ for the triangle. Consequently, their Z-scores are interdependent. In fact, except when both are zero or when the 3-path is 1 and the triangle is 0 (an artefact of the G-Trie model), they are symmetric. Additionally, the significance profile of size-four graphs is partly encoded in the size-three profile, suggesting an advantage in using both sizes as target variables. For only size-four, we observed no further strict dependencies other than the mathematical constraint stated in Section 3.

Real-World Data. Since we are interested in assessing the performance of the models with real-world data, we compiled a dataset based on real networks of multiple categories. Besides varying the type of network, we vary in their relative size in terms of number of nodes and edges. We have categorised them into two groups based on their average size with respect to the training set: (1) small-scale networks, which range from slightly smaller to around average the size of synthetic ones, and (2) medium-large-scale, which exceed (sometimes significantly) the average synthetic network size.

The additional material (Section A) provides more details for this section.

5 Methodology

The model used in the experiments is similar to the one described in by Chen et. al [Chen et al., 2020], definition A.1. from Appendix A. We utilise the same architecture only modifying the target and loss to the ones described in Section 3. We employ GIN [Xu et al., 2019], GAT [Veličković et al., 2018], GraphSage [Hamilton et al., 2017] and GCN [Kipf and Welling, 2017] as backbones and optimise all models with Optuna so that they can represent the best possible result for the used parameter space. We train two instances, one based on non-deterministic (ND) datasets and other on deterministic (D) ones.

"Correct" Predictions. Criteria for a prediction of a significance-profile to be correct/useful depends largely on the research field. Hence, we evaluate the result on multiple simple error thresholds, namely, $\leq 5\%, \leq 10\%, \leq 25\%$ and $< 50\%$. The 50% threshold means that all predictions that match just the signal are considered as correct. We count predictions divided by synthetic generator. Furthermore, we ensure that for a prediction to count as "correct", SP values for all graphs in Ω must have the correct sign. Hence, for example, for the $\leq 5\%$ threshold, it is not enough for the error to be below $2 \times 0.05 = 0.1$, where 2 is maximum possible error, the signal of the predictions must also match.

Let T, be a naive benchmark model, predicting a random significance-profile, but taking into account the restrictions, described in Section 3, on the range of values each group of Ω can take. For the defined thresholds, according to $1e6$ simulations, T will have a rate of "correct" guesses of 0.0001% for 5%, 0.0019% for 10% 0.1108% for 25% and 0.4012% for 50%.

The additional material (Sections B and C) presents more details about the model, how it was trained and the results for more thresholds. Github link for code and replication steps.

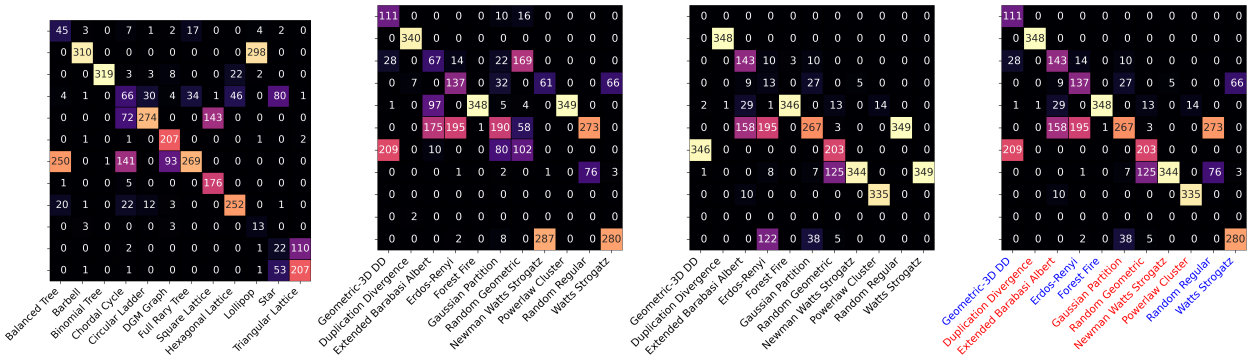
6 Results

For the ND segment, the best models were an instance of GIN and one of GraphSage (SAGE). As for the D segment, an instance of GIN stood as the undisputed best. All others were significantly worse and hence not further analysed.

6.1 Predictions in the Synthetic Dataset

Figure 1 presents confusion-matrix style heatmaps, H, with respect to the graph generator for the agreement between the predicted and true significance-profiles (SPs). Let exist a graph $\mathcal{G}$, model M, graph generators X and Y and an $\mathbb{S}$, denoting the set of all possible true SPs from the used datasets. Let $M(\mathcal{G}) = \hat{\boldsymbol{s}}$ and $\boldsymbol{s}_k = \text{argmin}_{\boldsymbol{s}_i \in \mathbb{S}} d(\hat{\boldsymbol{s}}, \boldsymbol{s}_i)$ where d is the mean absolute difference. If the X is the generator of the graph that originates $\boldsymbol{s}_k$ and Y the generator from the graph that lead to $\hat{\boldsymbol{s}}$, then $H[X, Y] \leftarrow H[X, Y] + 1$. Hence, a large number on the diagonal means that the generator did a good job at predicting SPs that can be traced back to the correct graph generator.

Following Figure 1, we conclude that the predictions made by all the models are reasonable for all generators. The results suggest that the model is sufficiently expressive to distinguish between different graph generators, as predictions often align with the correct graph generator. In fact, for Figure 1a, the errors mainly originate from mismatches between generators that are fundamentally very similar. For example, the lollipop graph can be considered a special case of the barbell graph. The same applies for the balanced and full rary tree. As for Figure 1b and 1c, the same applies. For example, the Watts-Strogatz graphs are often confused with the Newman-Watts-Strogatz graphs. Furthermore, we note that graphs that are known to completely impossible to be distinguished by 1-WL models like the Random Regular are always misrepresented. It is worth noting that SAGE and GIN extract different knowledge from the graphs. In fact, they seem complement each other rather well, as it can be seen in Figure 1d. Assuming that the choice of null model had little impact in the difficulty of the problem, the conclusion of the ability of a model with expressivity $\leq$ 1-WL to be able to distinguish graphs of different generators (inter-generator predictions) can be seen as a partial empirical confirmation of an old result by Babai and Kucera [Babai and Kucera, 1979, Babai et al., 1980], regarding the 1-WL test being able to distinguish any random graph with high probability as the size of graph approaches infinity.



(a) GIN D (b) GIN ND (c) SAGE ND (d) Best ND

Figure 1: Agreement between predictions and true significance-profiles. Last image combines best of SAGE and GIN.

Tables 1 and 2, give a detailed look at correct predictions as enunciated in Section 5. Overall, for this type of analysis, the performance of the models is reliable for some graph families but exhibits systematic errors in others. For example, the powerlaw cluster, forest-fire and duplication-divergence for the ND and star-graphs and circular-ladders for D present good performances. Some allow for $\geq 75\%$ correct predictions at 25% error, and all easily beat the baseline T. However, generally, most of the predictions do not accurately represent the true SPs. The conclusion of the inability of the models to generally distinguish graphs with high granularity among those in the same generator (intra-generator predictions) has theoretical backing for the case of the random regular generator [Babai and Kucera, 1979, Cai et al., 1989, Babai et al., 1980]. As for the other generators, following the result in [Babai and Kucera, 1979, Babai et al., 1980], theoretically, it should be highly probable that a model as powerful as the 1-WL could distinguish most of the graphs, not only at inter-generator level, but also at intra-generator level. We hypothesise that the mentioned result might not be very useful in practice. However, due to the good results for some generators, it is possible that size of graphs used is not enough for the bad performing generators.

Table 1: Number of D graphs with a predicted SP deemed "correct" (C) or "incorrect" (I). Generators with $\geq 50\%$ or $\geq 40\%$ are highlighted.

	GIN							
	5%		10%		25%		50%	
	C	I	C	I	C	I	C	I
Balanced Tree	13	307	14	306	39	281	52	268
Barbell	63	257	112	208	138	182	143	177
Binomial Tree	0	320	0	320	0	320	2	318
Chordal Cycle	0	320	55	265	104	216	114	206
Circular Ladder	0	320	145	175	152	168	152	168
DGM Graph	0	320	67	253	114	206	116	204
Full Rary Tree	4	316	21	299	34	286	36	284
Square Lattice	0	320	65	255	66	254	66	254
Hexagonal Lattice	0	320	0	320	0	320	3	317
Lollipop	0	320	0	320	100	220	100	220
Star	6	152	70	88	76	82	83	75
Triangular Lattice	35	285	117	203	143	177	144	176

Table 2: Number of ND graphs with a predicted SP deemed "correct" (C) or "incorrect" (I). Generators with $\geq 90\%$, $\geq 75\%$ or $\geq 50\%$ are highlighted.

	GIN								SAGE							
	5%		10%		25%		50%		5%		10%		25%		50%	
	C	I	C	I	C	I	C	I	C	I	C	I	C	I	C	I
Geometric-3D DD	0	349	67	282	246	103	246	103	0	349	3	346	45	304	53	296
Duplication Divergence	0	349	246	103	293	56	293	56	10	339	33	316	42	307	44	305
Extended Barabasi Albert	0	349	16	333	62	287	63	286	0	349	41	308	224	125	226	123
Erdos-Renyi	0	349	0	349	0	349	9	340	0	349	0	349	0	349	11	338
Forest Fire	1	348	50	299	261	88	335	14	4	345	100	249	323	26	334	15
Gaussian Partition	0	349	0	349	14	335	32	317	0	349	0	349	1	348	15	334
Random Geometric	0	349	27	322	86	263	86	263	0	349	1	348	34	315	113	236
Newman Watts Strogatz	0	349	126	223	134	215	145	204	0	349	0	349	161	188	222	127
Powerlaw Cluster	0	349	176	173	349	0	349	0	97	252	301	48	349	0	349	0
Random Regular	0	349	0	349	0	349	0	349	0	349	0	349	0	349	61	288
Watts Strogatz	0	349	29	320	74	275	97	252	0	349	0	349	43	306	56	293

6.2 Validations of the Assumptions Made

We begin by validating whether the assumption that predicting multiple scores simultaneously yields benefits over predicting each score individually. To do this, we trained eight models, each corresponding to a graph in Ω, using the ND segment. We employed the GIN variant of MPNNs, chosen for its theoretical and practical advantages. The models were trained without prior assumptions so that, when training for a single prediction, each model specialized in its respective graph. Table 3 presents the percentiles of the squared difference between true and predicted SPs on the validation dataset for each graph.

Based on Table 3, multi-target regression generally improves prediction accuracy across all graphs except for, arguably, the four-node clique and four-path. This supports our expectation, outlined in Section 3, that jointly predicting graphs with shared traits enhances performance. Regarding the increased error observed in the other two graphs, we hypothesise that this may be due to their limited benefit from shared information, as other graphs lack sufficient encoded data to enhance their predictions beyond a specialized model. Nevertheless, given the relative magnitude of error changes, we conclude that multi-target regression is superior for motif estimation. Additionally, it offers the advantage of requiring only a single model while maintaining competitive training times compared to single-target regression.

The second assumption we must validate is whether estimating SP directly is beneficial over calculating the SP after estimating subgraph counts. To do this, we trained a model, using the ND segment, to predict the frequency of graphs in Ω directly. Model and training procedure remain as in the former comparison.

Table 3: Error percentiles in the validation set of the squared error and their percent decrease/increase comparing the multitarget to the single target model.

Graph Type	 single multi	 single multi	 single multi	 single multi	 single multi	 single multi	 single multi	 single multi
100%	2.921 2.868 -1.814%	1.231 0.991 -19.496%	1.193 0.816 -31.601%	1.498 0.861 -42.523%	1.056 1.268 +20.076%	2.275 2.388 +4.967%	1.814 2.193 +20.893%	2.940 2.621 -10.850%
95%	0.279 0.250 -10.394%	0.316 0.396 +25.316%	0.309 0.320 +3.560%	0.350 0.347 -0.857%	0.547 0.502 -8.227%	0.517 0.489 -5.416%	1.294 0.741 -42.736%	1.463 0.808 -44.771%
75%	0.078 0.091 +16.667%	0.095 0.043 -54.737%	0.047 0.041 -12.766%	0.083 0.053 -36.145%	0.067 0.038 -43.284%	0.082 0.100 +21.951%	0.210 0.042 -80.000%	0.357 0.048 -86.555%
50%	0.004 0.009 +125.000%	0.017 0.011 -35.294%	0.004 0.006 +50.000%	0.008 0.007 -12.500%	0.005 0.003 -40.000%	0.007 0.012 +71.429%	0.051 0.007 -86.275%	0.042 0.007 -83.333%
25%	0.000 0.001 +100.000%	0.001 0.000 -100.000%	0.000 0.000 0.00%	0.001 0.001 0.00%	0.001 0.000 -100.000%	0.001 0.002 +100.000%	0.005 0.000 -100.000%	0.012 0.003 -75.000%

To avoid generating 500 random networks per test instance, we opted for approximating the Z-score that would be later obtained from subgraph estimation. To achieve this, we decomposed the subgraph frequency variable into actual frequency (y) and model error (z). Assuming that the difference between a value $z \sim \mathrm{z}$ and μ_z is proportional to σ_z, minding signal indetermination, following $(y - \mathbb{E}[\mathrm{y}]) \pm \sigma_\mathrm{z} \,/\, \left(Var(\mathrm{y})^2 + Var(\mathrm{z})^2\right)^{1/2}$, we do not have to do additional calculations since all values were either acquired during the training of the model (values regarding z) or were collected during the dataset construction (values regarding y). Table 4 presents percentiles for the absolute difference between true and predicted SPs. The values under "Count" correspond to the minimum difference (hence, worst case comparison) resulting from estimations with all valid signal combinations.

Table 4: Error percentiles in the real-world dataset of the absolute difference and their percent decrease/increase comparing direct SP estimation (SP) to estimating graph frequencies (Count).

Graph Type	 Count SP	 Count SP	 Count SP	 Count SP	 Count SP	 Count SP	 Count SP	 Count SP
75%	0.469 0.222 -52.584%	0.377 0.323 -14.378%	0.326 0.199 -38.799%	0.322 0.227 -29.618%	0.361 0.173 -52.117%	0.680 0.298 -56.225%	0.509 0.172 -66.154%	0.625 0.278 -55.554%
50%	0.339 0.116 -65.907%	0.236 0.109 -53.718%	0.175 0.056 -68.007%	0.174 0.083 -52.079%	0.214 0.083 -61.143%	0.311 0.126 -59.456%	0.368 0.072 -80.448%	0.358 0.079 -78.013%
25%	0.041 0.044 +5.426 %	0.115 0.042 -63.960%	0.106 0.036 -66.396%	0.065 0.031 -52.797%	0.063 0.020 -68.212%	0.144 0.041 -71.614%	0.328 0.027 -91.903%	0.300 0.021 -92.887%

Results show a significant improvement when using direct estimation for all graphs. Given its gain and computational efficiency (not having to directly calculate the occurrences for the control graphs), direct significance-profile prediction is preferable for motif estimation in the chosen null model.

6.3 Model Predictions in the Real-World Dataset

Similarly to the synthetic data, the predictions on the real-world dataset are, generally, not very good at precise intra-category level estimation. Interestingly, models grouped real-world graphs based on their "similarity" to synthetic datasets. For example, consider the *ia-escorts-dynamic* network (Figure 9b). In this network, nodes represent buyers and escorts, and edges indicate interactions between them. Its SP profile closely resembled that of a duplication-divergence model. Next, take the *coauthor-CS* network (Figure 9a). Here, nodes represent authors, and an edge connects two nodes if the authors have co-authored a paper. This network produced an SP profile similar to that of a forest-fire model. Finally, consider the *ia-primary-school-proximity* network (Figure 9b). In this case, nodes represent individuals, and an edge is formed when two people are in close physical proximity for a certain period. The SP profile for this network matched that of a geometric model. Among other correct matches, these three examples demonstrate a concrete case where the model found the appropriate synthetic generator for the real-world network[2] based on the significance-profile. This suggests that while models struggle with precise real-world predictions, they can help identify the closest synthetic model for real networks based on significance-profiles. (Images of predictions available in the additional material). Additionally, the model remains stable, with errors increasing by at most $\approx 20\%$ as networks scales up to 1000 times the train size.

[2]More details about these networks can be found in the supplemental material

Points of divergence. In network similarity discovery based on SP, two key concepts are crucial. First, the model's ability to distinguish networks is constrained by the expressivity of the space of the SP used. The more expressive the space, the more reliable the ability of the model to distinguish networks. Secondly, if the model predicts similar profiles for two graphs $\mathcal{G}$ and $\mathcal{H}$, indicating they resemble a graph $\mathcal{F}$, this suggests $\mathcal{G}$ and $\mathcal{H}$ may originate from a process similar to $\mathcal{F}$. However, this conclusion is valid only if the true profiles of $\mathcal{G}$ and $\mathcal{H}$ are indeed similar; otherwise, the model's lack of expressivity leads to incorrect conclusions.

6.4 Time Comparisons

We will compare the efficiency of doing predictions using a GNN model with using Gtrie. We will employ the normalized measures of Speedup, Speedup $=$ Total Time Used$^{(A)}$/Total Time Used$^{(B)}$ and Core Efficiency Gain Core Efficiency Gain $=$ Total Core Time$^{(A)}$/Total Core Time$^{(B)}$ for tasks A and B.

The GNN model presented a ~ 1.6 million speedup for the medium-large real dataset with a ~ 1625 core efficiency gain. For the small real dataset, the values were ~ 18211 speedup and a ~ 19 core efficiency gain. For the deterministic segment we got a ~ 82594 speedup and a ~ 112 times more efficiency per core. As for the non-deterministic segment, we got a ~ 548888 speedup and a ~ 745 more core efficient task.

7 Conclusions

Although no GNN-based method is specifically designed for predicting motifs, our MPNN models with synthetic data still fall short for precise real-world SP discovery. However, we empirically showed that through simple modifications, namely, multitarget regression and direct SP estimation, we achieved stability across out-of-distribution network sizes and improved results over existing traditional SP estimation via subgraph counting. Additionally, we hint that when performing direct SP estimation, the incorporation of a null model may help overcome the theoretical limitations of motif estimation based on subgraph counting. Our findings also suggest that the models are promising for network categorisation since they can distinguishing high-level differences between graphs. Future work can focus on categorising the SP estimation problem space based on different null models.

References

[Abboud et al., 2021] Abboud, R., Ceylan, I. I., Grohe, M., and Lukasiewicz, T. (2021). The Surprising Power of Graph Neural Networks with Random Node Initialization. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 2112–2118, California.

[Akiba et al., 2019] Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[Albert and Barabási, 2000] Albert, R. and Barabási, A. L. (2000). Topology of evolving networks: Local events and universality. *Physical Review Letters*, 85(24):5234–5237.

[Albert et al., 1999] Albert, R., Jeong, H., and Barabási, A.-L. (1999). Diameter of the world-wide web. *Nature*, 401(6749):130–131.

[Alon, 2007] Alon, U. (2007). Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461.

[Asikainen et al., 2020] Asikainen, A., Iñiguez, G., Ureña-Carrión, J., Kaski, K., and Kivelä, M. (2020). Cumulative effects of triadic closure and homophily in social networks. *Science Advances*, 6(19).

[Babai et al., 1980] Babai, L., Erdős, P., and Selkow, S. M. (1980). Random Graph Isomorphism. *SIAM Journal on Computing*, 9(3):628–635.

[Babai and Kucera, 1979] Babai, L. and Kucera, L. (1979). Canonical Labelling of Graphs in Linear Average Time. *Annual Symposium on Foundations of Computer Science - Proceedings*, (2):39–46.

[Balcilar et al., 2021] Balcilar, M., Heroux, P., Gauzere, B., Vasseur, P., Adam, S., and Honeine, P. (2021). Breaking the limits of message passing graph neural networks. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 599–608. PMLR.

[Barabási and Pósfai, 2017] Barabási, A.-L. and Pósfai, M. (2017). *Network science*. Cambridge University Press.

[Barabási et al., 2000] Barabási, A.-L., Albert, R., and Jeong, H. (2000). Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications*, 281(1-4):69–77.

[Besta et al., 2022] Besta, M., Grob, R., Miglioli, C., Bernold, N., Kwasniewski, G., Gjini, G., Kanakagiri, R., Ashkboos, S., Gianinazzi, L., Dryden, N., and Hoefler, T. (2022). Motif prediction with graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 35–45, New York, NY, USA. Association for Computing Machinery.

[Brandes et al., 2003] Brandes, U., Gaertler, M., and Wagner, D. (2003). *Experiments on Graph Clustering Algorithms*, pages 568–579. Springer Berlin Heidelberg.

[Cai et al., 1989] Cai, J.-Y., Furer, M., and Immerman, N. (1989). An optimal lower bound on the number of variables for graph identification. In *30th Annual Symposium on Foundations of Computer Science*, pages 612–617. IEEE.

[Chen and Ying, 2023] Chen, J. and Ying, R. (2023). TempME: Towards the Explainability of Temporal Graph Neural Networks via Motif Discovery. (NeurIPS):1–24.

[Chen et al., 2023] Chen, K., Liu, S., Zhu, T., Qiao, J., Su, Y., Tian, Y., Zheng, T., Zhang, H., Feng, Z., Ye, J., and Song, M. (2023). Improving Expressivity of GNNs with Subgraph-specific Factor Embedded Normalization. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, volume 1, pages 237–249, New York, NY, USA. ACM.

[Chen et al., 2020] Chen, Z., Chen, L., Villar, S., and Bruna, J. (2020). Can graph neural networks count substructures? *Advances in Neural Information Processing Systems*, 2020-Decem(NeurIPS).

[Corso et al., 2020] Corso, G., Cavalleri, L., Beaini, D., Liò, P., and Veličković, P. (2020). Principal Neighbourhood Aggregation for Graph Nets. *Advances in Neural Information Processing Systems*, 2020-Decem(NeurIPS).

[Dall and Christensen, 2002] Dall, J. and Christensen, M. (2002). Random geometric graphs. *Physical Review E*, 66(1):016121.

[Dareddy et al., 2019] Dareddy, M. R., Das, M., and Yang, H. (2019). motif2vec: Motif Aware Node Representation Learning for Heterogeneous Networks. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1052–1059. IEEE.

[Dorogovtsev et al., 2002] Dorogovtsev, S. N., Goltsev, A. V., and Mendes, J. F. F. (2002). Pseudofractal scale-free web. *Physical Review E*, 65(6).

[Erdős and Rényi, 1960] Erdős, P. and Rényi, A. (1960). On the Evolution of Random Graphs. *Magyar Tudományos Akadémia Értesitöje*, 5(1):17–61.

[Frasca et al., 2022] Frasca, F., Bevilacqua, B., Bronstein, M. M., and Maron, H. (2022). Understanding and Extending Subgraph GNNs by Rethinking Their Symmetries. *Advances in Neural Information Processing Systems*, 35(NeurIPS).

[Fu et al., 2023] Fu, T., Wei, C., Wang, Y., and Ying, R. (2023). DeSCo: Towards Generalizable and Scalable Deep Subgraph Counting.

[Ginoza and Mugler, 2010] Ginoza, R. and Mugler, A. (2010). Network motifs come in sets: Correlations in the randomization process. *Physical Review E*, 82(1).

[Golovin et al., 2017] Golovin, D., Solnik, B., Moitra, S., Kochanski, G., Karro, J. E., and Sculley, D., editors (2017). *Google Vizier: A Service for Black-Box Optimization*.

[Hamilton et al., 2017] Hamilton, W. L., Ying, R., and Leskovec, J. (2017). Inductive representation learning on large graphs.

[Higham et al., 2008] Higham, D. J., Rašajski, M., and Pržulj, N. (2008). Fitting a geometric graph to a protein-protein interaction network. *Bioinformatics*, 24(8):1093–1099.

[Holme and Kim, 2002] Holme, P. and Kim, B. J. (2002). Growing scale-free networks with tunable clustering. *Physical Review E*, 65(2):026107.

[Hu et al., 2021a] Hu, W., Fey, M., Ren, H., Nakata, M., Dong, Y., and Leskovec, J. (2021a). Ogb-lsc: A large-scale challenge for machine learning on graphs.

[Hu et al., 2021b] Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. (2021b). Open graph benchmark: Datasets for machine learning on graphs.

[Huang et al., 2022] Huang, Y., Peng, X., Ma, J., and Zhang, M. (2022). Boosting the Cycle Counting Power of Graph Neural Networks with I^2-GNNs. pages 1–27.

[Ispolatov et al., 2005] Ispolatov, I., Krapivsky, P. L., and Yuryev, A. (2005). Duplication-divergence model of protein interaction network. *Physical Review E*, 71(6):061911.

[Kipf and Welling, 2017] Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pages 1–14.

[Lanzinger and Barceló, 2023] Lanzinger, M. and Barceló, P. (2023). On the power of the weisfeiler-leman test for graph motif parameters.

[Lee et al., 2018] Lee, J. B., Rossi, R. A., Kong, X., Kim, S., Koh, E., and Rao, A. (2018). Higher-order graph convolutional networks.

[Leskovec et al., 2007] Leskovec, J., Kleinberg, J., and Faloutsos, C. (2007). Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data*, 1(1).

[Liaw et al., 2018] Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E., and Stoica, I. (2018). Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*.

[Lubotzky, 1994] Lubotzky, A. (1994). *Discrete Groups, Expanding Graphs and Invariant Measures*. Birkhäuser Basel.

[Mangan and Alon, 2003] Mangan, S. and Alon, U. (2003). Structure and function of the feed-forward loop network motif. *Proceedings of the National Academy of Sciences*, 100(21):11980–11985.

[Milo et al., 2004a] Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., Ayzenshtat, I., Sheffer, M., and Alon, U. (2004a). Superfamilies of Evolved and Designed Networks. *Science*, 303(March):1538–1542.

[Milo et al., 2004b] Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2004b). Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 9781400841(October):217–220.

[Moritz et al., 2018] Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M. I., and Stoica, I. (2018). Ray: A distributed framework for emerging ai applications.

[Morris et al., 2020] Morris, C., Kriege, N. M., Bause, F., Kersting, K., Mutzel, P., and Neumann, M. (2020). Tudataset: A collection of benchmark datasets for learning with graphs.

[Murphy et al., 2019] Murphy, R. L., Srinivasan, B., Rao, V., and Ribeiro, B. (2019). Relational Pooling for Graph Representations. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:8192–8202.

[Newman and Watts, 1999] Newman, M. and Watts, D. (1999). Renormalization group analysis of the small-world network model. *Physics Letters A*, 263(4-6):341–346.

[Oliver et al., 2022] Oliver, C., Chen, D., Mallet, V., Philippopoulos, P., and Borgwardt, K. (2022). Approximate Network Motif Mining Via Graph Learning.

[Penrose, 2003] Penrose, M. (2003). *Random geometric graphs*. Oxford Studies in Probability. Oxford University Press, London, England.

[Pržulj, 2007] Pržulj, N. (2007). Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183.

[Ribeiro et al., 2021] Ribeiro, P., Paredes, P., Silva, M. E. P., Aparicio, D., and Silva, F. (2021). A survey on subgraph counting: Concepts, algorithms, and applications to network motifs and graphlets. *ACM Comput. Surv.*, 54(2).

[Ribeiro and Silva, 2013] Ribeiro, P. and Silva, F. (2013). G-tries: a data structure for storing and finding subgraphs. *Data Mining and Knowledge Discovery*, 28(2):337–377.

[Rong et al., 2020] Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., and Huang, J. (2020). Self-supervised graph transformer on large-scale molecular data.

[Roy et al., 2020] Roy, S., Ghosh, P., Barua, D., and Das, S. K. (2020). Motifs enable communication efficiency and fault-tolerance in transcriptional networks. *Scientific Reports*, 10(1).

[Shen-Orr et al., 2002] Shen-Orr, S. S., Milo, R., Mangan, S., and Alon, U. (2002). Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics*, 31(1):64–68.

[Sheng et al., 2024] Sheng, J., Zhang, Y., Wang, B., and Chang, Y. (2024). MGATs: Motif-Based Graph Attention Networks. *Mathematics*, 12(2):293.

[Silva et al., 2023] Silva, M. E. P., Gaunt, R. E., Ospina-Forero, L., Jay, C., and House, T. (2023). Comparing directed networks via denoising graphlet distributions. *Journal of Complex Networks*, 11(2).

[Veličković et al., 2018] Veličković, P., Casanova, A., Liò, P., Cucurull, G., Romero, A., and Bengio, Y. (2018). Graph attention networks. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pages 1–12.

[Veličković et al., 2020] Veličković, P., Ying, R., Padovano, M., Hadsell, R., and Blundell, C. (2020). Neural Execution of Graph Algorithms. *8th International Conference on Learning Representations, ICLR 2020*.

[Wang and Zhang, 2023] Wang, Y. and Zhang, M. (2023). Towards Better Evaluation of GNN Expressiveness with BREC Dataset.

[Watanabe, 2023] Watanabe, S. (2023). Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance.

[Watts and Strogatz, 1998] Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.

[Wegner, 2014] Wegner, A. E. (2014). Motif Conservation Laws for the Configuration Model. pages 4–6.

[Wu et al., 2018] Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. (2018). Moleculenet: A benchmark for molecular machine learning.

[Xu et al., 2019] Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2019). How Powerful are Graph Neural Networks? pages 1–17.

[Yan et al., 2023] Yan, Z., Zhou, J., Gao, L., Tang, Z., and Zhang, M. (2023). Efficiently Counting Substructures by Subgraph GNNs without Running GNN on Subgraphs.

[Ying et al., 2019] Ying, R., Bourgeois, D., You, J., Zitnik, M., and Leskovec, J. (2019). GNNExplainer: Generating Explanations for Graph Neural Networks. *Advances in neural information processing systems*, 32:9240–9251.

[Ying et al., 2020] Ying, R., Wang, A. Z., and Jiaxuan You, J. L. (2020). Spminer: Frequent subgraph mining by walking in order embedding space.

[Yu and Gao, 2022] Yu, Z. and Gao, H. (2022). Molecular representation learning via heterogeneous motif graph neural networks.

[Zhang et al., 2023] Zhang, B., Feng, G., Du, Y., He, D., and Wang, L. (2023). A Complete Expressiveness Hierarchy for Subgraph GNNs via Subgraph Weisfeiler-Lehman Tests. *Proceedings of Machine Learning Research*, 202:41019–41077.

[Zhang et al., 2020] Zhang, S., Hu, Z., Subramonian, A., and Sun, Y. (2020). Motif-driven contrastive learning of graph representations.

A Data Details

In this section we discuss with more detail some aspects of the data, both synthetic and real-world, used in the experiments discussed in the main text. Section A.1 presents details of how the non-deterministic synthetic data was generated and Section A.2 details about the deterministic segment. Section A.3 details findings on how the significance-profiles behave with respect to each other. Finally, Section A.4 makes a brief analysis of the real-world data used.

A.1 Non-Deterministic Generators

For the Erdős-Renyi model [Erdős and Rényi, 1960], we mainly aim at creating graphs in the three (out of four) main topological phases a graph achieve [Barabási and Pósfai, 2017]. We exclusively uniformly control the number of nodes within each of the delineated phases, namely, the "critical", "supercritical" and "connected" states. This strategic regulation facilitates substantial variability in graph size while preventing an excessive escalation of the referred metric that could possibly impede further computational processing.

For the Watts-Strogatz [Watts and Strogatz, 1998] and Newman Watts-Strogatz [Newman and Watts, 1999], we regulated the generation based on the total number of nodes, the initial number of neighbours and the probability of rewiring in order to generate networks that represented key sections of the characterisation based on the clustering coefficient and path length, as given by Watts and Strogatz [Watts and Strogatz, 1998].

For the extended Barabási-Albert model [Albert and Barabási, 2000], we defined as hyperparameters the total number of nodes and amount of connections a new node gains. Based on the equations delineated in the original article, we modulate our parameters grounded on two variables: the probability associated with the formation of new links (p) and the probability of rewiring existing connections (q). We aim to have graphs characterised by a power-law degree distribution with an exponent ranging uniformly between 2 and 3.

For the cluster power-law [Holme and Kim, 2002], we vary uniformly the number of nodes and calculate the necessary probability according to the original study to obtain a clustering coefficient of 0.35, 0.45 or 0.55.

The duplication-divergence generator [Ispolatov et al., 2005] operates by randomly selecting a node v from an initial graph and duplicating all edges connected to v with a retention probability denoted as σ. We select three regimes. Two of the selected regimes exhibit self-averaging behaviour concerning the number of edges, specifically when $0 < \sigma < e^{-1}$ and $e^{-1} < \sigma < 1/2$. The non-self-averaging regime is characterised by $1/2 < \sigma < 1$.

In the Gaussian random partition model [Brandes et al., 2003], k groups of nodes are generated with t nodes derived from a Gaussian distribution with mean s and variance v. The connectivity between nodes in a group is given by a probability p, and the connectivity inter-groups is given by q. In this generator, we parameterise the number of nodes $|V|$, the size of the k groups and the maximum number of allowed edges $|E_{\text{max}}|$. Both the p and q probabilities are calculated to not exceed the maximum number of edges according to Equation 1.

$$q \leq min\left(1,\ \frac{2|E_{\text{max}}|}{|V|^2 + |V|\big(\kappa \cdot s^{1/2} - s(\kappa+1)\big)}\right) \tag{1}$$

$$p \leq min(1,\ \kappa \cdot q) \tag{2}$$

We defined p as always having the possibility of going above q because we would like to have networks that can have a community structure in order to have a more diverse set of graphs. Hence, we put the bound of p as being scaled over q by κ, which we called over-attractiveness. The values used for the v and κ are 10 and 5 respectively. All other parameters are uniformly sampled from a predefined range[3].

In the case of the forest-fire model [Leskovec et al., 2007], we varied the number of nodes and the backward and forward probability between 0 and 0.4 (inclusive) to try to steer away from very aggressive Densification Power Law exponents and clique-like graphs, characteristics that, if severe, can hinder the subsequent steps from a computational point of view. With the values for the probabilities defined above, we expect to observe sparse networks that slowly "densify over time", together with decreasing diameter. All the graphs are made undirected after being generated.

For the random geometric graph, since some properties of the graph related to its connectedness, such as maximum cluster size and coefficient, vary with the dimension of the unit hypercube used [Dall and Christensen, 2002, Penrose, 2003], besides the number of nodes, we we decided to vary the dimension of the hypercube between 2 and 5. However,

[3] More details for the parameters available in the code.

we did not efficiently explore all possible configurations within the referred dimensions because we limited the number of edges. Similarly to a random geometric model, we used a random geometric model in 3D with duplication divergence [Higham et al., 2008]. For this model, we followed Silva et al. [Silva et al., 2023].

The last model in the non-deterministic segment is the random regular generator. In this case, the parameters subject to uniform variation were the total number of nodes and the degree assigned to each node, which once determined, remain constant across all nodes.

A.2 Deterministic Generators

We complemented the graphs generated by the non-deterministic generators with smaller amounts of graphs from deterministic generators. These generators have their network completely and without randomness determined once their parameters are chosen.

The first group of deterministic generators consists of multiple types of trees. We use the binomial tree model parametrised on its order and the balanced tree (full rary-tree) parametrised on its height and branching factor.

The second group is based on modified cycles. We use the circular ladder generator, varying the complete size of the graph and the chordal cycle [Lubotzky, 1994], also varying its complete size.

The third group is based on complete graphs and encompasses the barbell and lollipop graphs. The barbell graph is made of two complete graphs of size k connected by a path of size m. The lollipop is a barbell graph with only one complete graph and the path. In order to not complicate subsequent steps, we carefully limited the size of the complete graphs.

The fourth group consists of the Dorogovtsev-Goltsev-Mendes model [Dorogovtsev et al., 2002]. This generator modulates a scale-free discrete degree distribution with exponent $1 + ln\,3/ln\,2$ by using a rather simple rule: "*At each time step, to every edge of the graph, a new vertex is added, which is attached to both the end vertices of the edge.*" (in [Dorogovtsev et al., 2002]). We vary the magnitude of the number of nodes and edges by changing n, resulting in $3(3^n + 1)/2$ and 3^{n+1} nodes and edges respectively.

The fifth group consists of lattices. Namely, we use 2D hexagonal, triangular lattices and 3D square lattices. The first 2 lattices have the option of allowing for boundary periodicity. All lattices vary in terms of the size of each dimension.

Finally, the last group consists of star graphs of various sizes.

Since the types of graphs that the deterministic generators generate are not subject to randomness, it is redundant to create multiple graphs for each set of parameters. However, in order to introduce a degree of randomness to the deterministic graphs, we introduced a probability of random rewiring of a percentage of edges after the graph is generated. The rewiring procedure for a single edge consists of selecting an edge (u, v) from a graph $\mathcal{G}$, deleting it and attaching one of the ends, u or v, to another node w. If u is picked and (u, w) already exists, then $\mathcal{G}$ will exit the procedure with one less edge. Since we want some variability but still want to preserve the original deterministic graphs, for each generator, two sets of graphs $\mathbb{S}_1$ and $\mathbb{S}_2$ will be constructed according to the proposed generator parameters mentioned above. After that, $\mathbb{S}_1$ is not subject to any rewiring, and for each graph in $\mathbb{S}_2$, $p\%$ of its edges are rewired according to the procedure described earlier. According to this methodology, we generated four versions. The first had 2 edges swapped, the second 25% of the edges swapped, the third 10% and the fourth 60%. We stick to version two due to being the best performing one according to preliminary tests. This fact means that 25% seems to be a good choice of random-rewiring so that the information encoded in the deterministic graphs is maximised.

A.3 Pattern interconnectivity

In the Section 4 of the main text we introduced a result regarding the symmetry of the score of size three patterns. We have a strong indication that even without the normalisation of the Z-score, the number of occurrences between connected graphs of the same size is highly related. More formally, the relation between the Z-scores of the graphs of size three can be described as follows. Let x be a random variable denoting the number of induced occurrences of triangles in any graph that follows the degree distribution D. Let y be a random variable denoting the occurrence of induced 3-paths in any graph that follows the degree distribution D. Equation 3 gives the relation between Z-scores of x and y.

$$X = \begin{cases} 0, & \text{if} \quad \mathrm{y} - \mu_{\mathrm{y}} = 0 \\ -\frac{\sigma_{\mathrm{x}}}{\sigma_{\mathrm{y}}}(Y - \mu_{\mathrm{y}}) + \mu_{\mathrm{x}}, & \text{otherwise} \end{cases} \tag{3}$$

When standardised to a mean of 0 and a standard deviation of 1, the Z-scores of both variables, exhibit symmetry. Hence, it is possible to express their non-standardised values as linear combinations of each other. Considering the mean and standard deviation of the counts of 3-paths and triangles for D, given any graph $\mathcal{G}$ that follows D, it is possible to get the concrete count of triangles from the count of 3-paths and vice-versa.

Even though from a practical point of view, the result from Equation 3 has little implications due to the dependence on the first raw moment and the second central moment of both the distributions of x and y, it presents a strong indication of what was postulated in Section 3 of the main text. That is, it further solidifies the connectivity between the graphs selected for Ω. In this case, the relation is so strong that we believe to be redundant to try to predict both scores. Moreover, following the normalisation procedure, the restrained nature of the result raises questions about the choice of modelling the problem as a regression task for the size three graphs. However, despite these observations, we stick to our initial formulation since in theory it does not significantly undermine the capacity of the model.

The result experimentally verified in the above paragraphs can be seen as a small extension of Ginoza and Mugler [Ginoza and Mugler, 2010] and Wegner [Wegner, 2014] to undirected patterns of size three. In particular, adapting from Wegner, Equation 4 displays the conservation law for the number of induced 3-paths.

$$\#\text{3-paths}_{ind} = \underbrace{\#\text{3-paths}_{\overline{ind}}}_{\text{fully defined by degree sequence}} - \underbrace{3\#\text{triangles}}_{\text{not fully defined by degree sequence}} \tag{4}$$

Since the number of non-induced 3-paths depends only on the degree sequence $\left(\sum_{i=0}^{|V|}\binom{|N(i)|}{2}\right)$, it will not change under the configuration model. Hence, the number of induced 3-paths is a variable that once the degree sequence is fixed, depends only on the number of triangles. As for the number of triangles, they depend on the order the edges are added to the graph under Equation 5.

$$\text{total triangles} = \sum_{t=0}^{|E|} \#\text{triangles}_t \tag{5a}$$

$$\text{total 3-path} = \sum_{t=0}^{|E|} (\#\text{3-path}_t - \#\text{triangles}_t) \tag{5b}$$

$$\#\text{triangles}_{t+1} = |\{w | w \in N(u^t) \wedge w \in N(v^t)\}| \tag{5c}$$

$$\#\text{3-path}_{t+1} = |N(u^t)| + |N(v^t)| - 2\#\text{triangles}_{t+1} \tag{5d}$$

where nodes u and v represent the nodes that were connected by an edge at iteration t. Hence, any realisation of a degree sequence through the configuration model will always have its number of induced 3-paths negatively correlated with the number of triangles.

Regarding the relation between graphs of size three and graphs of size four, by analysing Figure 2, it is possible to understand that there is a relation between the significance-profiles of these graphs. This relationship is particularly pronounced concerning the 4-star, tri-pan and 4-clique, as the values of the significance-profiles assumed by these graphs are distributed across the spectrum centred at 0, contingent upon the value held by the 3-path. For example, if the significance-profile of the 3-path is positive, the significance-profile of the 4-star will most likely be positive following the distribution given by the yellow histogram. As for the 4-cycle and bi-fan, this relation is not as strong. For the 4-cycle, we learn that the values are mostly zero when the significance-profile for the 3-path is negative and is quite dispersed across the space otherwise, with a small peak close to the 1 value. As for the bi-fan, besides learning that it is unlikely that it takes a large negative value, even though hard to discern from the figure, 46.2% of the values are 0 when the significance-profile for the 3-path is positive, and 69.7% are between -0.1 and 0.1 for the same conditions.

A.4 Real-World Dataset

The selected type categories are described in list A.4. The numbers between the square brackets in each bullet point correspond to the number of networks each category has in each of the scale categories (small and medium-large).

- **ANIMAL SOCIAL**: [10/8] Networks describing the social behaviour of non-human animals, spanning species such as ants, dolphins, lizards, sheep, and others.
- **BIOLOGICAL**: [10/10] Networks of protein-protein interactions, a metabolic networks of small organisms, and a networks of disease connections in humans based on shared genes.

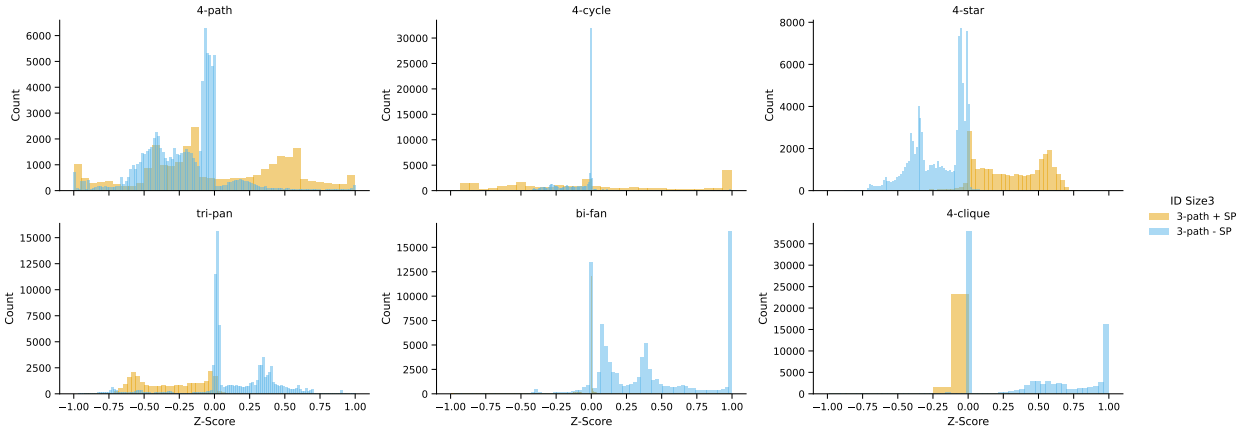


Figure 2: Distribution of the significance profiles for the graphs of size 4, given the value the 3-path took. The positive value corresponds to $1/\sqrt{2}$ and the negative to $-1/\sqrt{2}$.

- **BRAIN**: [9/10] Connectome networks of various brain regions such as the cerebral cortex, interareal cortical areas, and synaptic networks, across multiple species including cats, worms, mice, macaques, and humans.
- **CHEMOINFORMATICS**: [10/0] Networks of multiple different enzyme structures.
- **COLLABORATION CITATION** [6/8]: Networks of of paper citations and author collaborations.
- **INFRASTRUCTURE**: [5/7] Electric grids and road networks.
- **INTERACTION**: [5/6] Networks of physical contact between humans in various contexts, together with some digital contact, for example, by e-mail or a phone call.
- **SOCIAL COMMUNICATION**: [2/10] Interaction between humans in social networks such as mutually liked Facebook pages, friendship connections and retweets.

Figure 3a and Figure 3b show a summary of the number of edges and nodes for the different type and scale categories. The red dashed lines represent the average of the minimum node (or edges) quantity, the average of the mean node (or edges) quantity and the average of the maximum node (or edges) quantity respectively, calculated for all 23 graph models used in the synthetic dataset.

In Figure 3a and Figure 3b, we can observe that the distinction between scale categories is influenced by the type category of the network. For example, ANIMAL SOCIAL networks tend to be smaller than INFRASTRUCTURE networks in the medium-large category. However, this relationship varies by scale, as both network types exhibit similar sizes in the small-scale category. In any case, in the general scenario, we expect that the small-scale category encloses networks from being smaller than an average network from the training set until networks that can be slightly larger than the mean training network. As for the medium-large category, they hold networks that can be around the average size of a training network to networks that are several orders of magnitude larger than the average size of a training network. The dashed red lines in Figure 3a and Figure 3b help validate this statement.

In summary, we have 56 networks in the small-scale category and 59 in the medium-large category. If a graph in the test set was not already a simple undirected static graph when it was obtained, we transformed it in a graph following said conditions[4].

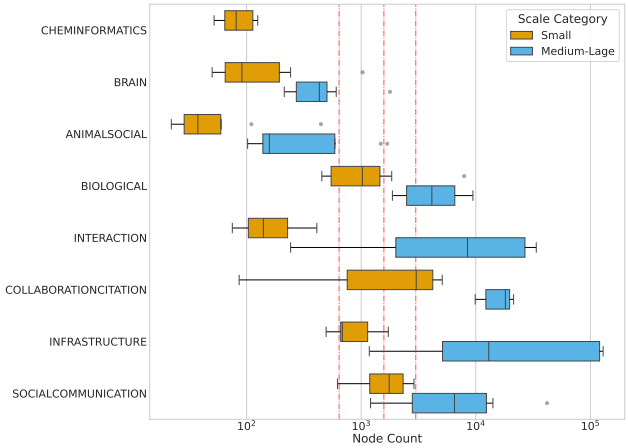
B Methodological Details

Section B.1 describes how the data was partitioned for training and Section B.2 the base model, the hyperparameter space and the method and software used for training.

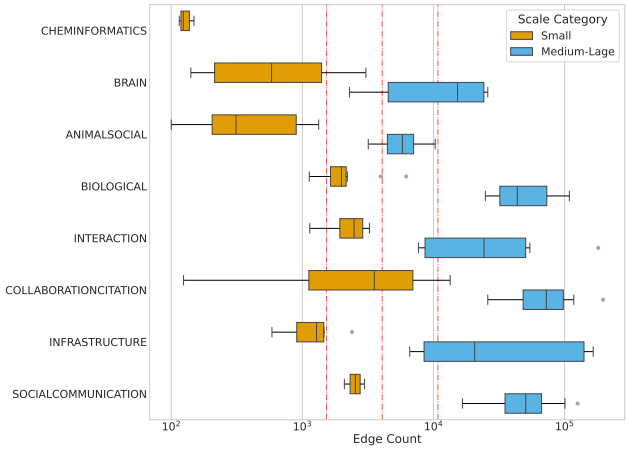
B.1 Exact dataset partition

We conduct separate experiments using the two segments of the produced data, the deterministic and the non-deterministic. Furthermore, the split in train-validation-test is stratified by random sampling a percentage p of

[4]Please see the supplemental material for the references for each of the individual 115 networks.



(a) Summary of how the number of nodes is distributed for the real networks.



(b) Summary of how the number of edges is distributed for the real networks.

Figure 3: Summary of the distribution of the node and edge count of the real networks. All data is presented in logarithmic scale.

each of the generators for each segment. In the case of the deterministic segment, we use all 3200 graphs available. With $p = 0.7$, the training set has $3200 \times 0.7 \times 12 = 26880$ graphs, and the validation set has $3200 \times 0.2 \times 12 = 7680$ graphs. The remaining 10% are used for the test set. We avoided using larger datasets due to memory restrictions. As for the non-deterministic segment, in order for it to have a comparable total size to the deterministic segment, we sampled $3490 \times 0.7 \times 11 = 26873$ graphs and the validation set $3490 \times 0.2 \times 11 = 7678$.

B.2 Initial Base Model

The model ($\mathcal{B}$) consists of three modules. The first module consists of K layers of a GNN. The job of the first module is to work on the graph data and adjust the node embeddings so that the second module, a global pooling function, can summarise them into a single graph-level embedding. The third module is an MLP that takes as input the graph embedding and will adapt it to output the final prediction for the normalised Z-scores of the graphs. Figure 4 shows a diagram of the model.

All the optimisations for the hyperparameters of $\mathcal{B}$ were performed by Optuna [Akiba et al., 2019] with 450 rounds of suggestions of hyperparameters, orchestrated through Ray [Liaw et al., 2018, Moritz et al., 2018]. Moreover, the hyperparameter sampling procedure employed the Tree-Structured Parzen Estimator [Watanabe, 2023], while the pruning strategy was executed through the application of the median rule [Golovin et al., 2017]. Table 5 presents the hyperparameter space used for model $\mathcal{B}$.

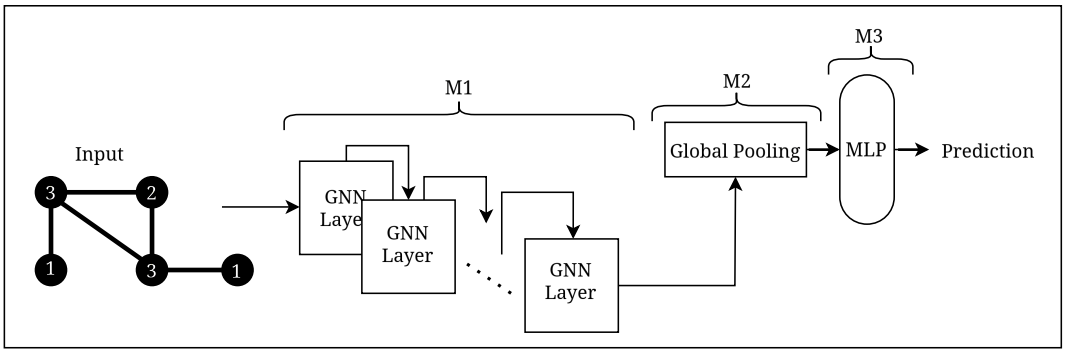


Figure 4: Illustration of the base model $\mathcal{B}$ divided in three modules, M1, M2 and M3.

Table 5: Break down of the hyperparameter space used for $\mathcal{B}$.

	Min	Max	Epochs	Batch Size	Learning Rate (log)
GNN Depth (M1)	2	3	100	{16, 32, 64, 128, 25}	[0.00001,0.001]
Hidden Dimension (M1)	6	16			
GNN Dropout (M1)	0.0	0.9			
Jumping Knowledge (M1)	max, cat, lstm				
Global Pool (M2)	add				
MLP Depth (M3)	2	6			
Hidden Dimension (M3)	6	16			
MLP Dropout (M3)	0.2	0.9			

The asymmetry in the hyperparameter space presented in Table 5 stems from our choice of preemptively test a slightly larger hyperparameter space and identify some values that resulted in very bad results. From this early testing phase, we also narrowed down M3 from a global add, mean or max function to just the global add function. This result aligns with some limitations that are known for the mean and max pooling functions [Xu et al., 2019]. AS for the number of epochs, we verified that 100 epochs are generally enough to obtain results before convergence or near convergence of the model. Furthermore, the fixed values of 100 epochs can be shortened not only by the pruner but also by an early-stopping module with a grace period of 25 epochs, synced with the median pruner, and patience of other 25 epochs of not seeing an improvement for the global minimum loss. Moreover, since we believe our problem does not need very long range dependencies since the structures in Ω can be fully defined by a hop size of 2, in order to try to limit the problem of over-smoothing we limited the maximum number of GNN layers to 3 based on the findings that most networks have a small diameter [Albert et al., 1999, Barabási et al., 2000, Watts and Strogatz, 1998]. By limiting the GNN layers, we also hope to reduce over-squashing.

Each experimental iteration with different M1 backbones comprised 450 trials, each uniquely characterised by a distinct combination of hyperparameters suggested by Optuna. Most of the training was done using a single consumer-grade NVIDIA RTX 3090 and later a NVIDIA RTX A6000. Inference was performed using a consumer-grade NVIDIA RTX 4070. Computations involving G-trie were performed on a AMD Opteron 6380.

The code is available in this Github repository and the data, figures and other intermediary results are available in this Figshare link. You can also view the results of the training procedure using the main Wandb page for this project.

C Experiment Results

Figure 5 and 6 shows the summary of the results from the 450 rounds of hyperparameter optimisation for each model used in M1. The solid line represents the mean score, and the semi-transparent bound around each line represents the standard error. The displayed metrics are the MSE for the train and validation data, the median absolute error, $med\big(\{med_i(|y_i - \hat{y}_i|, \forall i \in |\boldsymbol{y}|)\}\big)$, the maximum absolute error calculated for a full prediction of a significance profile, and the mean value for the worst-performing prediction of a graph from Ω. The maximum error is given by $max(\{\sum_{j\in|\Omega|}|\boldsymbol{Y}_{[i,j]} - \hat{\boldsymbol{Y}}_{[i,j]}|, \forall i \in |D_{\text{valid}}|\})$ where $\boldsymbol{Y}$ is a 2-d matrix with the first dimension giving the number of examples in the validation dataset and the second dimension the length of $\boldsymbol{s}$. As for the mean value of the worst-performing predictions, it is given by $mean(max\{\sum_{i\in|D_{\text{valid}}|}|(\boldsymbol{Y}_{[i,j]} - \hat{\boldsymbol{Y}}_{[i,j]})|, \forall j \in |\Omega|\})$.

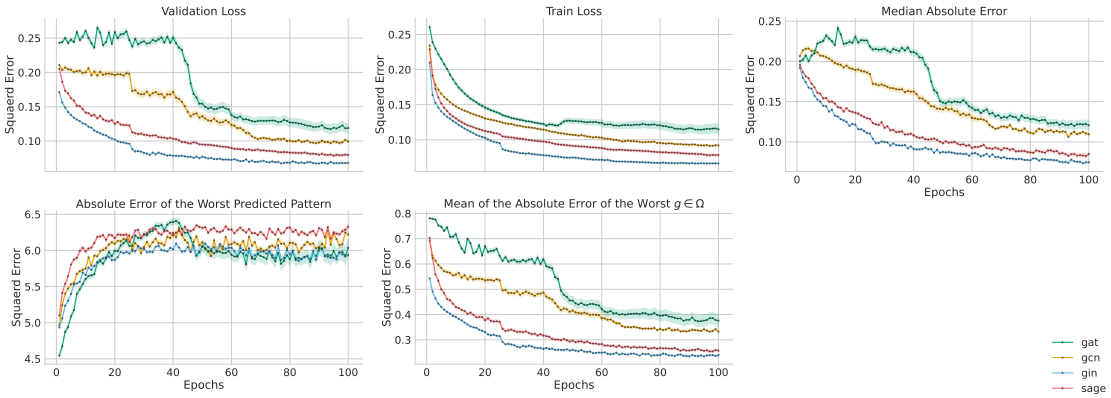


Figure 5: Learning curves for the various backends used for M1 when trained with the deterministic segment of graph generators.

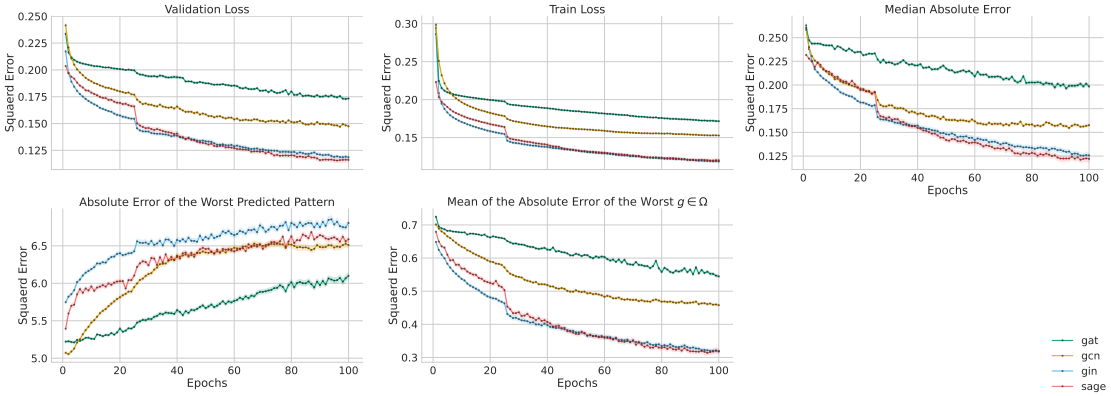


Figure 6: Learning curves for the various backends used for M1 when trained with the non-deterministic segment of graph generators.

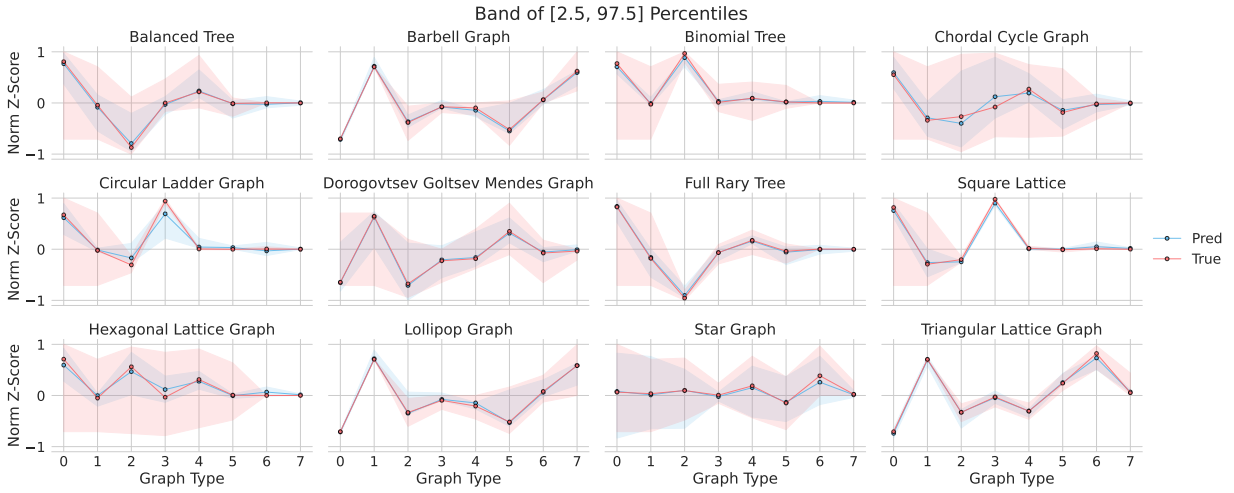
The learning curves for the deterministic segment (Figure 5) show that all models improve significantly within the first 50 epochs, especially in all metrics except for the maximum absolute error. GIN outperforms all other models by a wide margin, prompting its selection for further analysis. For the non-deterministic segment (Figure 6), the performance of GraphSAGE and GIN is very close, with GraphSAGE holding a slight numerical edge. Since both models perform comparably, both will be retained for further evaluation.

C.1 Predictions

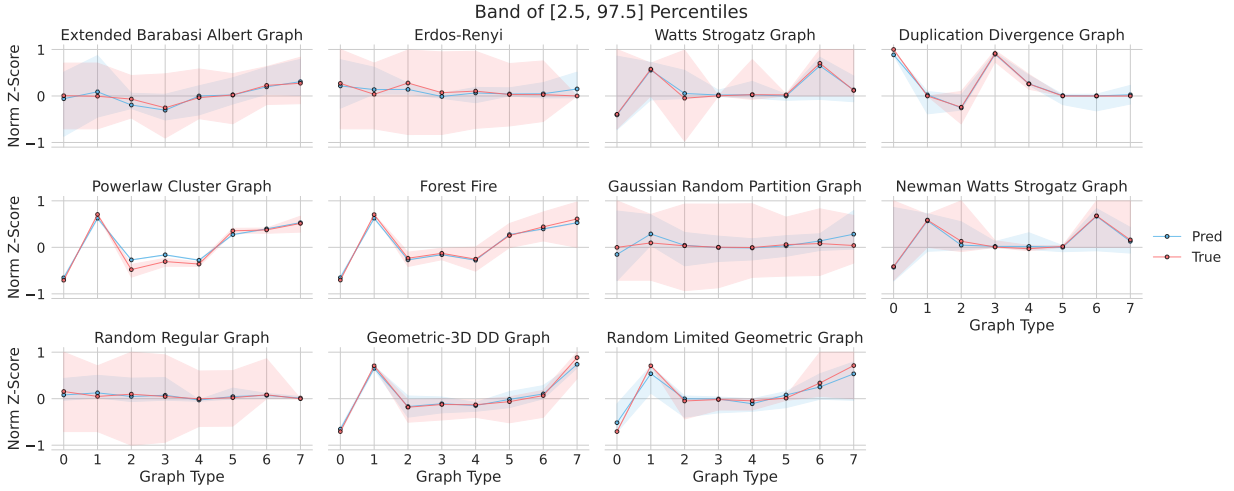
Figures 7 display a summary of the predictions for each generator made by each selected model. The agreement between the true and predicted mean significance-profile is further evidence that the models can perform inter-generator prediction. Furthermore, the much tighter percentile band for the predicted mean significance-profile can be a rough indicator that the models struggle to make accurate intra-generator predictions.

Figure 8a and 8b shows a analysis similar to the one made in the main text in tables 1 and 2. In this case, we calculate the percentage of "correct" predictions for error thresholds ranging from 5% to 50% in steps of 1%. This allows us to see the how volume of "correct" predictions evolves over increasing thresholds.

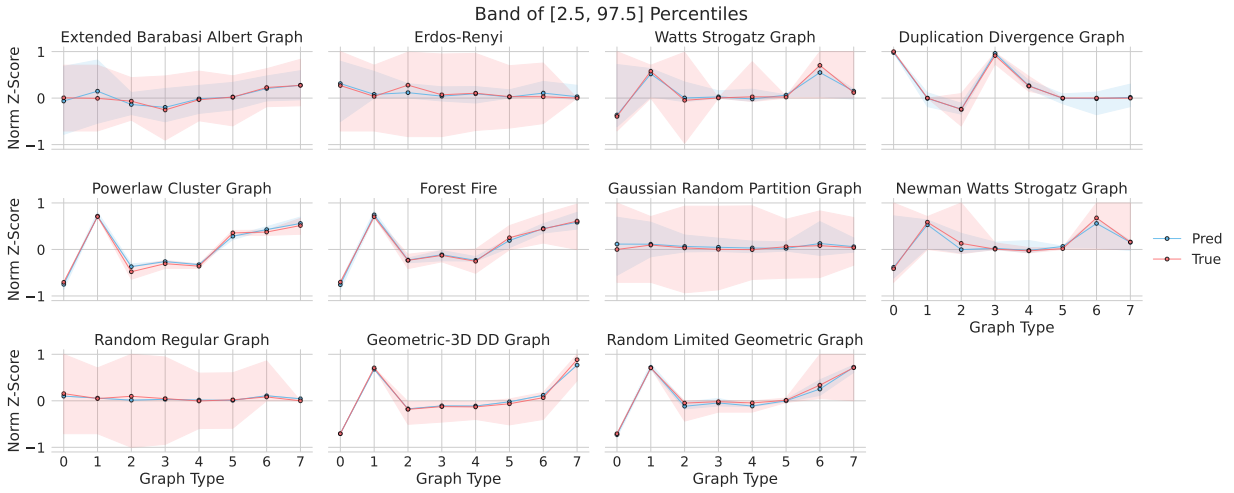
Figure 9b and 9a show some examples of predictions made in the real-world dataset. We can see that even though the model generally struggles to make accurate intra-generator predictions, it makes predictions that can be traced back to the synthetic generator. The *ia-escorts-dynamic*, *coauthor-CS*, and *ia-primary-school-proximity* are the examples highlighted in the main text. For these specific examples, the significance-profiles can be traced back to the duplication-divergence, forest-fire, and geometric generators respectively. This suggests that these generators produce the most similar graphs to real-world networks from a motif analysis perspective. This alignment is expected, as these generators are designed to replicate such patterns.



(a) GIN trained on the deterministic segment.



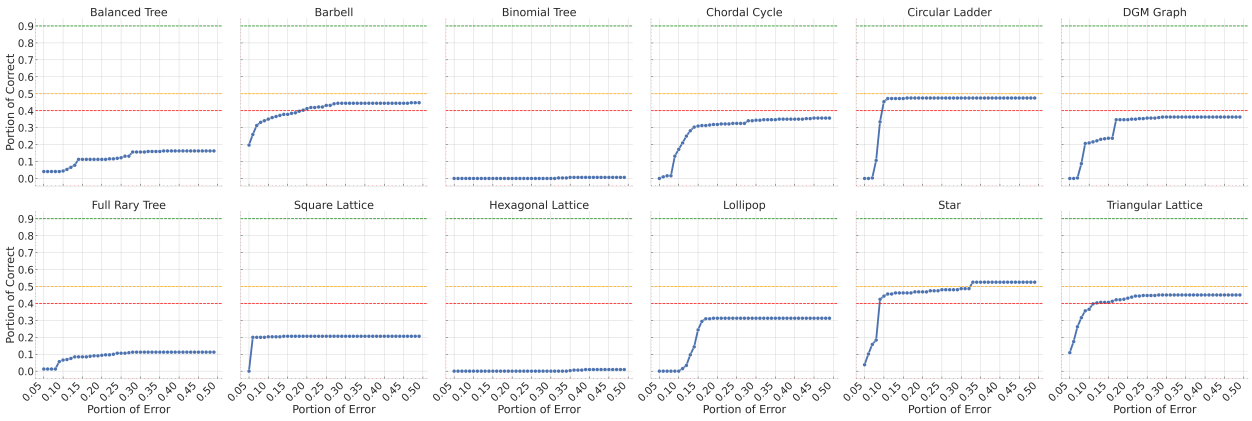
(b) GIN trained on the non-deterministic segment.



(c) SAGE trained on the non-deterministic segment.

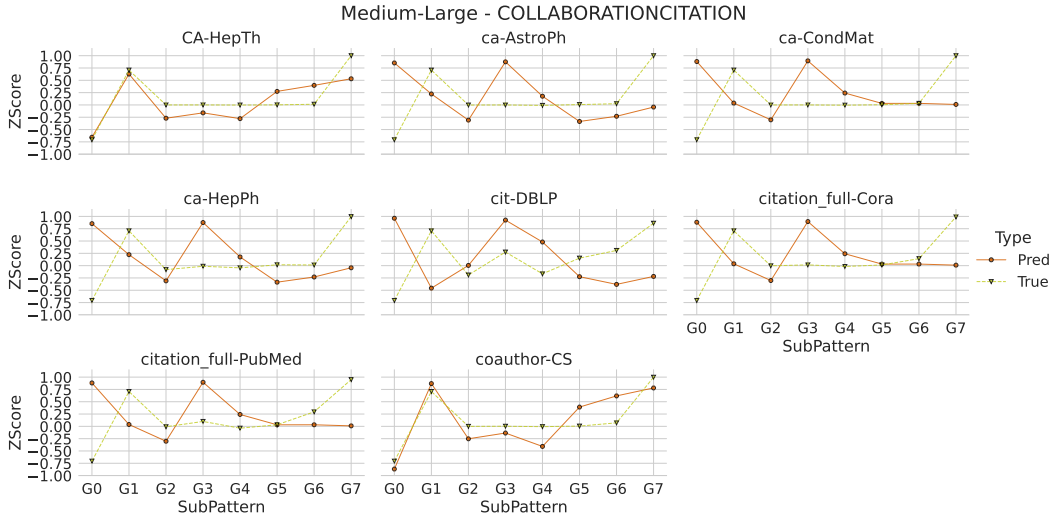
Figure 7: Predictions for each model in each of their corresponding synthetic test datasets.

(a) Non-Deterministic Segment

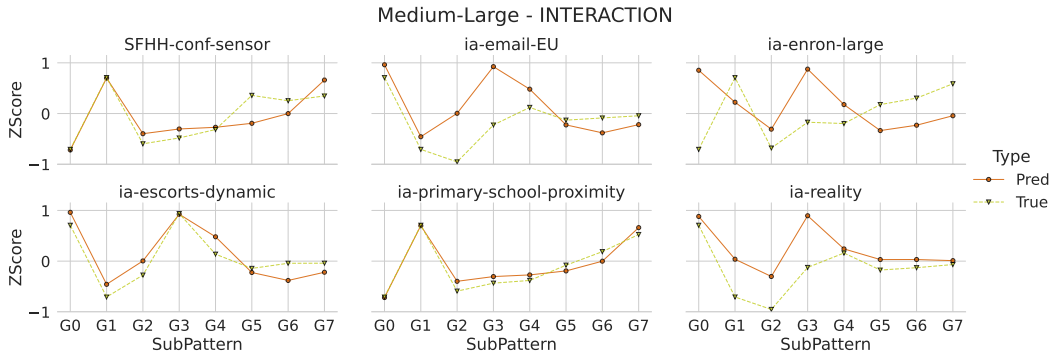


(b) Deterministic Segment

Figure 8: Evolution of the percentage of "correct" predictions, as defined in the main text, starting from 5% to 50% in 1%. The green line denotes the 90% threshold, the orange the 50% and the red 40%.



(a)



(b)

Figure 9: Predictions by GIN trained on the non-deterministic segment. Orange lines with circles are predictions and dark-yellow with triangles true values.