Master Thesis

Homomorphism distortion: A metric to distinguish them all and in the latent space bind them

by

Martin Carrasco
(mca305)

First Supervisor: Erik Bekkers
Daily Supervisor: Bastian Rieck
Second Reader: Peter Bloem

Aug 22, 2025

Submitted in partial fulfillment of the requirements for
the VU degree of Master of Science in Artificial Intelligence

Homomorphism distortion: A metric to distinguish them all and in the latent space bind them

Martin Carrasco[1,3] [0000-0003-3420-8048], Olga Zaghen[2], Erik Bekkers[2], and Bastian Rieck[3] [0000-0003-4335-0302]

[1] Vrije Universiteit Amsterdam
[2] University of Amsterdam
[3] Fribourg University

Abstract. For far too long, expressivity of graph neural networks has been measured *only* in terms of combinatorial properties. In this work we stray away from this tradition and provide a principled way to measure similarity between vertex attributed graphs. We denote this measure as the *graph homomorphism distortion*. We show it can *completely characterize* graphs and thus is also a *complete graph embedding*. However, somewhere along the road, we run into the graph canonization problem. To circumvent this obstacle, we devise to efficiently compute this measure via sampling, which in expectation ensures *completeness*. Additionally, we also discovered that we can obtain a metric from this measure. We validate our claims empirically and find that the *graph homomorphism distortion*: **a.)** fully distinguishes the `BREC` dataset with up to 4-WL non-distinguishable graphs, and **b.)** *outperforms* previous methods inspired in homomorphisms under the `ZINC-12k` dataset. These theoretical results, (and their empirical validation), pave the way for future characterization of graphs, extending the graph theoretic tradition to new frontiers.

Keywords: Representation Learning · Graph Neural Networks · Graph Theory.

1 Introduction

General statement of interest. The success of graph neural networks (GNNs) can be credited to the wide array of relational phenomena that can be modeled as graphs, such as in physics [54], computational chemistry [52], and neuroscience [5], among others. The dominating paradigm in this area is *message passing*, giving rise to message passing neural networks (MPNNs) [25]. Briefly stated, MPNNs update a vertex representation iteratively by gathering the neighbors' representations and aggregating them. After enough iterations, each vertex will have collected information from the entire graph (at least in theory). In this framework, the structure of the graph serves as an *inductive bias*, providing a notion of closeness and connectedness to information. As such, a *good* GNN should be able to leverage the topology of the graph to extract relational information. In what follows, we dig deeper into what *good* exactly means.

Problem statement and motivation One of the ways to quantify the expressivity of a GNN is by its ability to distinguish graphs [44, 45, 66]. The most common framework for *expressivity* is the Weisfeiler and Leman [43] (1-WL) test. It consists of an iterative refinement of vertex attributes based on neighborhood structures until convergence. If at the terminal iteration L, two graphs, G and G', have different attributes, the test deems them non-isomorphic. Conceptually, WL induces a separation between the classes of graphs it can distinguish and the ones it can't. A key observation is that to do this, it needs to *characterize* these graphs. In the case where WL cannot distinguish non-isomorphic G, G', there exists a *characteristic* that distinguishes them which WL ***cannot*** capture. Thus, a better characterization implies accounting for more fine-grained properties, which is a requirement to distinguish increasingly similar graphs. For example, vertex degrees distinguish certain graphs; however, strongly regular graphs (with the same parameters) cannot be distinguished by vertex degree alone [10]. From this point of view, the power of a GNN can be understood as it's capacity to extract relevant *characteristics* from a vertex attributed graph. Characterization can be exploited from different angles, such as using kernels [47, 13, 15, 35], tools from topological data analysis[2, 11, 57], and algebraic topology [40, 49], among others. One rich vein of characterization is *homomorphism counts*. Previously, it has been demonstrated that the set of homomorphism counts to all graphs can distinguish any pair of non-isomorphic graphs [36]. Even more, many properties of graphs can be directly linked to homomorphism counts to particular graphs [37]. Given that the homomorphisms between graphs are tightly linked to their properties, different works in machine learning look at homomorphism counts [30, 4, 34, 16, 63, 56, 15, 3, 62, 38, 68] and subgraph counts [55, 67, 53, 23, 9, 39, 16, 50, 1, 24]. A caveat of this and other powerful characterization methods is that they are inherently binary. They cannot convey *slight difference*. If two graphs differ in their homomorphism counts by even one value, they are automatically deemed non-isomorphic. In graph theory, this makes complete sense. However, in machine learning, we are interested in cases where we want to assign labels to *approximately isomorphic* graphs or graphs that follow a more general notion of equivalence. Then, it becomes relevant to discuss (approximate) isomorphism of vertex attributed graphs, and not purely combinatorial objects. In practice, past works rely on empirical choices of *proxy graphs* to which to measure homomorphism counts, which are highly data-dependent. Consequently, it makes sense that they struggle to distinguish hard classes of graphs, such as distance-regular graphs [2, 30].

Our approach and contribution. In this work, we explore what is a *good* characterization of a vertex attributed graph. Particularly, we seek to challenge the overarching narrative of *binary distinguishability*. In the combinatorial domain, a binary distinction criterion is valid, but in the machine learning setting, a notion that allows for approximate equivalence is needed. To address this gap, we propose a first step in this direction. We introduce the *graph homomorphism distortion*, a powerful (pseudo)-metric inspired by the idea of characterization through homomorphisms, and we show that it can completely characterize graphs, both as a (pseudo)-metric and as a graph embedding. Additionally, we demonstrate that we can achieve high empirical expressivity by building on the results from Welke et al. [62] and the expectation-completeness framework. Finally, we demonstrate that our method can fully distinguish all graphs in the recent expressivity benchmark `BREC` [61] and outperforms previous homomorphism and subgraph count-based approaches in learning settings.

Outline. First, Section 2 introduces key theoretical concepts from graph theory and machine learning needed for positioning this work in the intersection of the two. It aims to provide all the necessary background knowledge, enabling readers to comprehend the methodology without prior expertise in graph theory or GNNs. Section 3 contains an in-depth exposition on expressivity measurements on GNNs, current approaches based on the characterization capabilities of graphs by use of homomorphism counts, and the computational costs involved. Section 4 introduces the *graph homomorphism distortion* along with its properties. Section 5 contains empirical validation of our claims, both in non-learning and learning scenarios. Finally, Section 6 contains final remarks.

2 Background

This section is divided into two distinct, ordered subsections **(I.)** Graph theory, **(II.)** Graph Neural Networks . Each builds on the previous one using the defined theoretical concepts to elucidate a congruent background. For more context and terminology refer to Section A.

2.1 Graph theory

This subsection closely follows Fomin, Heggernes, and Kratsch [22], introducing the concepts that are fundamental to understanding graphs as objects and allowing the reader to familiarize with them. Some of the notation differs from that in the book to be more accessible to the machine learning community.

Graphs. A graph is a pair of sets denoted $G = (V, E)$, where $v \in V$ are the vertices of G and $\{u, v\} \in E$ are the edges of G for $u, v \in V$[4]. When we want to refer to multiple graphs at the same time, we denote the vertex set of a graph G with $V(G)$ and the edge set with $E(G)$. The number of vertices of a graph is denoted $|G|$, also called *order*, and the number of edges is denoted $\|G\|$. Two vertices u, v are adjacent if $\{u, v\} \in E$. Two edges $e \neq f$ are adjacent if there is a v such that $v \in e$ and $v \in f$. If all the vertices in G are adjacent to all other vertices, then G is called complete and denoted as K^n where $n = |G|$.

Morphisms on graphs. Morphisms between graphs are a central element of this work. As such, their treatment takes a special focus on the background. Many intuitive, or relatively simple, notions in this section will come back to haunt us as the computational aspect behind them becomes a requirement. However, that is left for Section 3.3. For now, the focus is on the morphisms.

Definition 1 (Graph homomorphism). *Given two graphs $G = (V, E)$ and $G' = (V', E')$ a function $\varphi : V \to V'$ is a graph homomorphism if it preserves edges (adjacencies), i.e. $\{u, v\} \in E$ implies that $\{\varphi(u), \varphi(v)\} \in E'$.*

[4] Note that according to this definition, we are taking into account only *undirected* graphs.

If there exists such a φ between two graphs G and G', the graphs are *homomorphic*. Also, the set of all homomorphisms from G to G' is denoted $\mathrm{Hom}(G, G')$. The cardinality of this set is denoted $\mathrm{hom}(G, G')$.

Definition 2 (Graph Isomorphism). *Given $G = (V, E)$ and $G' = (V', E')$. If there is a homomorphism $\varphi : V \to V'$ which fulfills the following conditions, then φ is an* isomorphism*:*

1. *Its inverse $\varphi^{-1} : V' \to V$ is also a homomorphism, namely if $\{u', v'\} \in E'$ then $\{\varphi^{-1}(u'), \varphi^{-1}(v')\} \in E$*
2. *φ is a bijection.*

Similarly to the previous morphism, given two graphs G and G', if there exists an isomorphism φ between them, they are said to be *isomorphic*, which is written as $G \simeq G'$. Also, if φ is an isomorphism from G to itself, it is called an *automorphism*.

Definition 3 (Graph property). *A graph property is a class of graphs $\mathcal{G}$ that is closed under isomorphism. Namely, if a graph $G \in \mathcal{G}$ has a property $\mathcal{P}$, then all possible isomorphic graphs that fulfill $G' \simeq G$ also have property $\mathcal{P}$ and belong in $\mathcal{G}$.*

Definition 4 (Graph invariant). *A graph invariant is a function ξ on a graph that assigns the same value to isomorphic graphs. For a fixed G and any other G' such that $G \simeq G'$ we have $\xi(G) = \xi(G')$. A graph invariant* distinguishes *non-isomorphic graphs G and G', denoted $G \not\equiv_\xi G'$ if $\xi(G) \neq \xi(G')$ and conversely if G and G' are isomorphic then $G \equiv_\xi G'$ if $\xi(G) = \xi(G')$.*

Subgraphs. We say that $G' = (V', E')$ is a *subgraph* of $G = (V, E)$, denoted $G' \subseteq G'$ if $V' \subseteq V$ and $E' \subseteq E$. Also, in this scenario, G is a *supergraph* of G'. If $G' \subseteq G$ and $G \neq G'$ then G' is a *proper subgraph*. If G' contains all edges $\{u', v'\} \in E$ for $u', v' \in V'$ then G' is an induced subgraph of G. That is, V' induces G' on G denoted $G' =: G[V']$. The difference between a subgraph and an induced subgraph is subtle. Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. If $V' \subseteq V$ and $E' \subseteq E$, then G' is a subgraph. However, for it to be an *induced subgraph*, we required a stricter condition. If $u', v' \in V'$, which implies $u', v' \in V$, and $\{u', v'\} \in E$, we call G' an induced subgraph only if $\{u', v'\} \in E'$ for any pair u', v'. A *spanning subgraph* is a subgraph $G' \subseteq G$ where $V' = V$.

Vertex degrees. Let $G = (V, E)$ be a non-empty graph. The degree of a vertex denoted $d(v)$ is the number $|E(v)|$ of edges v is a part of. If all vertices of G have the same degree k, then G is called k-regular.

Graph classes. Different classes of graphs will have different properties. In the following sections, we will denote graph classes with $\mathcal{F}$ and a graph belonging to this class as F. If $\mathcal{F}$ is the class of all graphs, it is denoted $\mathcal{G}$ and it's restricted to graphs of a specific order as $\mathcal{G}_n = \{|G| \leq n \mid G \in \mathcal{G}\}$.

Definition 5 (Tree decomposition). *A* tree decomposition *of a graph G is a pair $(\mathcal{T}, \{X_a\}_{a \in V(\mathcal{T})})$ where $\mathcal{T}$ is a tree and $\{X_a\}_{a \in V(\mathcal{T})}$ is the family of subsets — referred to as bags — of $V(G)$ where*

1. *each* $v \in V(G)$ *is in at least one bag* X_a
2. *for each pair* $\{v, u\} \in E(G)$ *there is at least one bag* X_a *such that* $v, u \in X_a$
3. *for each* $v \in V(G)$ *the set* $\mathcal{T}_v := \{a \in V(\mathcal{T}) | v \in X_a\}$ *induces a connected subgraph of* $\mathcal{T}$.

The width of a tree decomposition is $\max_{a \in V(\mathcal{T})} |X_a| - 1$, the maximum size of a *bag* minus 1. The minimum width among all possible decompositions is called the *treewidth* of G, denoted $\mathrm{tw}(G)$.

Vertex attributed graphs. A *vertex attributed graph* is a triplet $G = (V, E, f)$ of vertices V, edges E and a function $f : V \to \mathcal{X}$ which assigns to each vertex $v \in V$ a function value $f(v)$ in some normed space $\mathcal{X}$. We use $||\cdot||$ to denote the norm of $\mathcal{X}$.

2.2 Graph Neural Networks

Graph neural networks are neural networks whose architecture operates over graph data. The following subsection serves as an introduction to popular network architectures that operate over graphs. In general, GNNs have the following general formula:

$$\mathbf{h}_v^{\ell+1} = \phi(\mathbf{h}_v^{\ell}, \psi(\{\!\{\mathbf{h}_u^{\ell} \mid u \in \mathcal{N}(v)\}\!\}) \tag{1}$$

where $\mathbf{h}_v^{\ell}$ denotes the hidden representation of vertex v at layer ℓ, $\{\!\{\cdot\}\!\}$ denotes a *multiset* and $\mathcal{N}(\cdot) : V \to 2^V$ is the *neighborhood function* defined as

$$\mathcal{N}(v) = \{u \mid \{u, v\} \in E\}, \tag{2}$$

and $\phi(\cdot)$ and $\psi(\cdot)$ are usually called $\mathrm{UPDATE}(\cdot)$ and $\mathrm{AGGREGATE}(\cdot)$. Generally, they are a composition of a parameterized learnable function modeled by a *multi-layer perceptron* (MLP) and an *activation function* modeled by a non-linearity. Denote the final layer with L. When $\ell = L$, the hidden representation $\mathbf{h}_v^{L}$ contains the final representation of vertex v. This representation can be used to perform tasks on nodes directly. However, to perform tasks on graphs, we need to have a representation for the whole graph. We define it as follows

$$\mathbf{h}_G = \mathrm{READOUT}\big(\{\!\{\mathbf{h}_v^{\ell} \mid v \in V)\}\!\}\big) \tag{3}$$

where $\mathrm{READOUT}(\cdot)$ is usually modeled with an MLP. With this foundation in mind, we move on to particular methods.

3 Related works

We introduce three subsections with relevant and intersecting directions related to graph representation learning. **(i)** Expressivity in GNNs, **(ii)** Characterization using counts, and **(iii)** Computability of homomorphisms

3.1 Expressivity of GNNs

The expressivity of GNNs can be viewed from different lenses. The key difference between GNNs and regular NNs is the ability to combine the topology of the graphs and the information contained in the features. In a recent survey, Zhang et al. [66] divides the expressivity landscape in three categories: **(1.)** Approximation ability, **(2.)** Separation ability, and **(3.)** Subgraph counting ability. . Item 2 and Item 3 are mainly focused on the graph theoretic notion of distinguishability. Suppose we denote a GNN as a parametrized function f_θ with parameters θ. In that case, we say its *separation ability* [41, 14] is how well it can distinguish two non-isomorphic graphs. Similarly, the *subgraph counting ability* is how well it can count subgraphs. [5]. Thus, the ability or *distinguishing power* is tightly linked to the *goodness* of the characterization that f_θ can learn. If we rely, and thus analyze, the characterization power only in binary measurements, we are prohibitively limited the capacity of f_θ. In either case, it seems to be the case that current expressivity measurements are far from perfect. Models that are theoretically very expressive tend to under perform or rely on heavy hyperparameter tunning, such as TDL models [21]. In this work, we aim to challenge this overarching view of expressiveness and advocate for a way to measure expressiveness in terms of *characterization* ability of vertex attributed graphs. Namely, one in which we can account for the case where two objects are ***similar enough***. Similar views have been recently expressed in Morris et al. [42]. Nevertheless, we review the previously mentioned approach formally to strengthen our claim of the need for better measurements.

Separation ability. The separation ability of a network f_θ is the class of non-isomorphic graphs $\mathcal{G}$, that f_θ can distinguish. Formally, let $\mathcal{G}$ be a class of graphs, letting f_θ be able to *separate* all graphs in $\mathcal{G}$, we have that for all non-isomorphic $G, G' \in \mathcal{G}$ then $f_\theta(G) \neq f_\theta(G')$. The most widely adopted measurement of separation ability is based on the Weisfeiler and Lehman (1-WL) test [32]. The test can be algorithmically defined with the following update rule

Definition 6. *Let $G = (V, E, f)$ be a vertex attributed graph. Let $\mathcal{C}$ be a set of colors. Let $\mathbf{c}_v^{(\ell)} \in \mathcal{C}$ be the color of the node v at iteration ℓ and let $\mathbf{c}_v^{(0)} = f(v)$. The WL test updates the color of node v at each iteration $\ell > 0$ as follows:*

$$\mathbf{c}_v^{(\ell)} = \mathtt{HASH}\Big(\mathbf{c}_v^{(\ell-1)}, \{\!\!\{\mathbf{c}_u^{(\ell-1)} \mid u \in \mathcal{N}(v)\}\!\!\}\Big)$$

where $\mathtt{HASH}$ bijectively maps its input to a color from $\mathcal{C}$.

The algorithm terminates when no color is updated when going from iteration ℓ to $\ell + 1$. It has been shown that GNNs are upper bounded in their *separation ability* by the 1-WL test [44]. In other words, traditional GNNs cannot distinguish two non-isomorphic graphs if 1-WL can't. However, variants of the test can be constructed where each k-tuple of nodes is also assigned a representation and thus a color [26]. This variation increases the *separation ability*, meaning that a broader class of graphs can be distinguished.

[5] Note that the subgraph counting ability can also be interpreted as a way to separate graphs.

Subgraph counting ability. The subgraph counting ability of a network is the class of graphs for which f_θ can distinguish all isomorphic subgraphs of each graph. Formally, let $\mathcal{G}$ be a class of graphs, let $G = (V, E) \in \mathcal{G}$ and $G' = (V', E'), G'' = (V'', E'')$ be isomorphic and subgraphs of G. Then, $V', V'' \subset V$ and $E', E'' \subset E$. Also, assume G' and G'' are different, denoted by $V' \neq V''$. If $f_\theta(G') \neq f_\theta(G'')$ then it can *count subgraphs*. This measurement cant be understood as the ability of a method to distinguish underlying substructures in a graph. There is a close relationship between subgraph counts and homomorphism counts that will be further explored in the next subsection.

New expressivity measurement. We want to point to the latest measurements in expressivity, which extend the previous concepts to different ways to quantify characterization ability. In Wang et al. [60], a new hierarchy of expressiveness termed $\mathcal{N}$-WL is introduced, which relies on notions of neighborhoods and subgraphs. Also, Puny et al. [51] proposes that expressive power can be looked through the lens of graph polynomials. More recently, [68] proposes an approach to expressivity that incorporates both subgraph counts and homomorphism counts. In terms of graph characterization, Dell, Grohe, and Rattan [18] finds a direct connection between k-WL distinguishability and graphs with fixed treewidth k, denoted $\mathcal{T}_k$.

Many expressivity measurements are anchored in graph theoretic ways to distinguish non-isomorphic graphs and thus assume features to be constant or disregard them as a way to distinguish graphs. In theory, that makes sense. For a big enough sample of graphs and with a big enough feature space, having the same features on two graphs is unlikely. Then, only the topology matters. However, could the features of the graph also be as informative as the topology ? Trivially, if two graphs $G = (V, E, f), G' = (V', E', f')$ are not isomorphic (structurally speaking), they are different. However, could a different form of equivalence, such as in distinguishability under a class $\mathcal{F}$ for the structure and isometry for the attributes better characterize the differences in G and G' ? We leave the reader with this question in mind and move on to the current methods using homomorphism and subgraph counts.

3.2 Characterization using counts

There has been a rise in the use of graph-theoretic methods for characterization in machine learning. Given that the main expressivity criterion, WL, is purely graph-theoretic, this is a logical direction. The use of counts for characterizations can be divided into intersecting subareas.

Characterization using subgraph counts. In [9], they stray away from homomorphism counts and then rely on subgraph counts. To augment node and edge features, the counts of isomorphisms to subgraphs of the original graphs are encoded in the node and edge, respectively, along with the initial features. Additionally, the architecture uses two different updates to take into account local substructure and global patterns. In a different approach by Yan et al. [65], authors decompose the graph into its subgraphs and apply a GNN to each subgraph. They show that the theoretical expressivity of this method surpasses regular GNNs. Additionally, the ability of a GNN to count substructures hast been related tor spectral invariance [24].

Characterization using homomorphism counts. The use of homomorphism counts as an inductive bias can be traced back to Nguyen and Maehara [47]. Here, the authors show the characterization power of homomorphism counts and correctly identify classes for graphs (graphs of a fixed treewidth k) for which these counts can be efficiently computed . Then, they use these counts in an SVM. The natural extension is presented by Barceló et al. [4], where the counts to graphs $F \in \mathcal{F}$ are used as inductive biases in a GNN, termed $\mathcal{F}$-MPNNs. Note that between these two approaches, there is no real improvement in expressivity, just another architecture in which to apply the counts. Later, Wolf et al. [63] correctly identifies that counting these homomorphisms to particular classes of graphs has a relationship to properties of the graphs, such as cycles. Still, there are no real improvements over the methods. More recently, two different paths to extend characterization through homomorphisms have been explored. The first starts with Jin et al. [30] and the insight that *graph motif parameters* are essentially linear combinations of homomorphism counts. As such, subgraph counts can be expressed as weighted homomorphism counts. The authors then construct a class of graphs, denoted $\text{Spasm}(\cdot)$, which can be built from a graph of fixed treewidth k and contains only graphs of treewidth k. This line is further extended in Bao et al. [3], where weighted homomorphism counts are compared in expressivity with the WL test as well as related to popular structural encodings such as RWPEs. Nevertheless, both of these last approaches lack a *complete* characterization of graphs and, in practice, end up relying on specific choices of graph patterns based on the dataset. Another line of research is shown in Maehara and NT [38], where authors propose a network architecture that progressively learns homomorphism across the layers. Interestingly, they theoretically related it to previous architecture and WL. However, their empirical evaluation is limited and uses outdated measures of expressivity, suggesting that empirically, the method might not be so robust.

Next steps. It is essential to notice that because of the central role of homomorphisms in graph characterization, there is an ongoing trend to make GNNs increasingly *homomorphism aware*. We can group under the same umbrella the capability to count subgraphs, as we know it can be retrieved by a linear combination of homomorphism counts [30]. However, many of the techniques are restricted to non-complete characterizations. Meaning that while the characterization function based on counts evaluates the same for isomorphic graphs, it's not always the case that it is different for non-isomorphic graphs. Usually, this is a direct consequence of the graph canonization problem, and the only method that we are aware of that tackles this is Welke et al. [62]. As such, the need for further study of the characterization of graphs in machine learning is prevalent. More importantly, these methods have not been used to explore attributed graphs or establish a measure beyond binary difference. We turn our attention to the computational aspect, as we have already introduced the intuition we need to show our method.

3.3 Computability of homomorphisms

In practice, computing the number of homomorphisms between two graphs G, G' is an NP-Complete problem [12]. While this holds in general, there are specific instances of G and G' where this problem is easier. If we fix either G or G' to be a particular graph, then the *hardness* changes. When a graph is fixed on the right, we denote it with H

and we write $\mathrm{Hom}(\cdot, H)$ to refer to calculating the homomorphisms from *any* graph (without restrictions) to a fixed (or a fixed type of) graph H. These are called *right-side* homomorphisms. When a graph is fixed on the left, we denote it with G and similarly write $\mathrm{Hom}(G, \cdot)$ for left-side homomorphisms. Surprisingly, both of these scenarios are very different.

$\mathrm{Hom}(\cdot, H)$ *homomorphisms.* These are homomorphisms from any arbitrary graph to $H \in \mathcal{H}$. In Hell and Nešetřil [28], authors show that this problem is polynomial if $\mathcal{H}$ is the class of bipartite graphs and NP-Complete if it is not. Additionally, if the chromatic number of H is equal to its maximum clique size, then solving the homomorphism problem is equivalent to the k-coloring problem with k being the chromatic number of H. Further refinements can be made for certain classes of H, such as it being a cycle [22] or not being a *projective core*, [48], among others.

$\mathrm{Hom}(G, \cdot)$ *homomorphisms.* These are homomorphisms from a graph $G \in \mathcal{G}$ to any arbitrary graph. If the class $\mathcal{G}$ contains only graphs with bounded treewidth k, then the homomorphisms can be counted in polynomial time as a function of $\mathrm{tw}(G)$ [17]. Also, if and only if the cores of the graphs in $\mathcal{G}$ have bounded treewidth k, then deciding if G and any other graph are homomorphically equivalent is polynomial in time [27]. That is, finding if there is at least one homomorphism.

Properties as functions of homomorphism counts. In [16], the authors related the subgraph counts to homomorphism counts. They show how the former can be expressed as a linear combination of the latter. In [58], the authors relate the homomorphism counts targeting cycle graphs to the eigenvalues and multiplicities of the adjacency matrix of a graph. Analogously, there is a relationship between homomorphism and constraint satisfaction explored in Grohe [27]. More recently, the role of graph *minors* has been studied concerning how two graphs are distinguished by their homomorphism counts to minors [46]

Graphs with bounded treewidth. Otherwise called *partial k-trees*. In Bodlaender [7], a new way to look at graphs is introduced. Namely, *treewidth* and *pathwidth*. Recall the definition of *treewidth* from Definition 5. The authors show that different properties and characteristics of graphs can be related to either their treewidth or by the minimal tree resulting from the decomposition. A seminal result in graph theory by Daaz, Serna, and Thilikos [17] shows that counting homomorphisms becomes computable when the *right-side* homomorphism restricts G to be of bounded treewidth k and expresses the complexity in terms of k.

Optimal computations. In Bodlaender and Koster [6], the *treewidth* of a graph and its respective tree decomposition are introduced as a way to algorithmically and efficiently calculate different graph properties. Bodlaender, Bonsma, and Lokshtanov [8] connects tree decompositions to dynamic programming and makes it computationally feasible to construct these solutions. Also, the following works have focused on lowering the algorithmic complexity of testing equality using homomorphisms under a fixed class [18]. The takeaway is that homomorphism counts, homomorphism enumeration, and specific graph properties can be computed in *feasible* time if we restrict ourselves to certain classes of graphs. This restriction is not too bad, as the properties we can still extract from counts under these restrictions are still desirable.

4 A new (pseudo)-metric on vertex attributed graphs

In this section we formally introduce the *graph homomorphism distortion* as a (pseudo)-metric over the space of vertex attributed graphs, show some interesting properties of this (pseudo)-metric and, under mild assumptions, show that it is *complete* and that it can be turned into a *complete graph embedding*. Finally, we turn our attention to the problem of computability. Along the way we realize that such a *good* characterization necessarily implies we have run into the *graph canonization problem*. However, we show that we can use approximation procedures to still provide a *good* characterization. Note that some proofs are referred to Section B due to a lack of space. However, the main results of the theorems are kept due to their relevance.

4.1 Introduction to the *graph homomorphism distortion*

The *graph homomorphism distortion* aims to encode the distance between two vertex attributed graphs through the individual distances of attributes to their counterpart through a homomorphism. Additionally, we posit that these transformations are not only necessary by themselves but also by how they *distort* the attributes the graph carries. This intuition is based on the aforementioned results in graph theory and homomorphism counts. We define the measurement as

Definition 7 (Homomorphism distortion). *Given two attributed graphs $G = (V, E, f)$ and $G' = (V', E', f')$ whose attribute functions f, f' map into the same co-domain, we define their homomorphism distortion $d_{HD}(G, G')$ as*

$$\begin{aligned} d_{\mathrm{HD}}(G, G') := \max\Big(&\inf_{\varphi \in \mathrm{Hom}(G,G')} \max_{v \in V} \|f(v) - f'(\varphi(v))\|, \\ &\inf_{\varphi' \in \mathrm{Hom}(G',G)} \max_{v' \in V'} \|f'(v') - f(\varphi'(v'))\|\Big) \end{aligned} \tag{4}$$

where we set $\inf \emptyset := \inf$.

An important aspect of this formulation is the maximum distance between vertex features for a fixed homomorphism φ or φ'. We zoom in on those and define them as the *attribute distance* of a homomorphism

Definition 8 (Attribute distance). *Given two attributed graphs $G = (V, E, f)$, $G' = (V', E', f')$. If there exist a homomorphism φ from G to G' and a homomorphism φ' from G' to G, we define the attribute distance both ways as*

$$\Delta_{G,G'}(\varphi) = \max_{v \in V} \|f(v) - f'(\varphi(v))\| \quad \textit{and} \quad \Delta_{G',G}(\varphi') = \max_{v' \in V'} \|f'(v') - f(\varphi'(v'))\| \tag{5}$$

This definition plays a central role in illustrating some of the properties of homomorphism distortion. Note that sometimes we omit the subindex G, G' when it's clear which graphs we are comparing. In what remains of the work, we refer the the (pseudo)-metric only as *homorphism distortion*.

4.2 Properties of the *homomorphism distortion*

Before we make any assumptions, we show that the homomorphism distortion is a pseudo-metric on vertex-attributed graphs. As a reminder, the properties of a pseudo-metric d are

Definition 9 (Pseudo-metric). *A pseudo-metric is a tuple $(\mathcal{X}, d)$ for a set $\mathcal{X}$ and a function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ where for all $x, y, z \in \mathcal{X}$ the following hold:*

1. $d(x, x) = 0$
2. $d(x, y) = d(y, x)$
3. $d(x, z) \leq d(x, y) + d(y, z)$

Applying this definition, we show these properties hold for the *homomorphism distortion*

Proposition 1. *The graph homomorphism distortion $d_{\mathrm{HD}}(G, G')$ is a pseudo-metric, i.e. the following properties hold for arbitrary vertex-attributed graphs G, G', G'' that share the same attribute function co-domain*

1. $d_{\mathrm{HD}}(G, G') \geq 0$ *and* $d_{\mathrm{HD}}(G, G) = 0$
2. $d_{\mathrm{HD}}(G, G') = d_{\mathrm{HD}}(G', G)$
3. $d_{\mathrm{HD}}(G, G') \leq d_{\mathrm{HD}}(G, G'') + d_{\mathrm{HD}}(G'', G')$

We defer this proof to Section B.1 and continue with a bound on a graph with two different attribute functions where

Proposition 2. *Given two attribute functions $f, g : V \to \mathcal{X}$ on the same graph G, we have*

$$d_{HD}((V, E, f), (V, E, g)) \leq \|f - g\|_\infty \tag{6}$$

where $\|\cdot\|_\infty$ denotes the supremum norm with respect to the norm of $\mathcal{X}$.

We refer the reader to Section B.2 for the proof. These general properties of the *homomorphism distortion* serve as a basis to find under what cases particular functions on graphs can be of use to characterize graphs. In what follows on the work, we center on specific choices of functions on graphs and particular co-domains of these functions that are already well studied [6].

4.3 A metric to distinguish them all...

A *good* characterization needs to be able to distinguish objects in two ways. First, it needs to be the *same* for equal objects and *different* for different objects. In our case, the equivalence relationship is graph isomorphism. As such, letting G and G' be two graphs, it is of interest to ask if the homomorphism distortion is 0 when $G \simeq G'$ and if it is different than 0 if $G \not\simeq G'$. We define this notion of good characterization as follows:

[6] Note that it is not always the cases that the *homomorphism distortion* between two graph G, G' where $G \simeq G'$ with attribute functions f, f' respectively is 0.

Definition 10. *Let G, G' be two graphs. The homomorphism distortion $d_{HD}(G, G')$ is* complete *if it is different than* 0 *for non-isomorphic graphs and is* 0 *for isomorphic graphs, namely*

$$G \simeq G' \quad \textit{if and only if} \quad d_{HD} = 0 \tag{7}$$

Since f and f' have the same co-domain, we will focus on some properties of f that are required for the completeness of the homomorphism distortion.

Lemma 1. *Let $G = (V, E, f)$ and $G' = (V', E', f')$ be two isomorphic vertex attributed graphs with. The* homomorphism distortion *$d_{HD}(G, G') = 0$ if f is permutation invariant and $f = f'$.*

We refer the reader to Section B.3 for the full proof. We proceed by noting that the ability to distinguish non-isomorphic graphs is also an important property. Even more, it is a *requirement* for a correct characterization function over graphs that there is no pair of non-isomorphic graphs G, G' that are treated as equivalent. If this is the case, that non-isomorphic graphs are treated differently, we call such a characterization *complete*. To achieve this property, we need to narrow down our definition of a function over the graph. For a function over the vertices of a graph $G = (V, E)$, $f : V \times \{E\} \to \mathcal{X}$ represents that the functions takes in a node $v \in V$ and the set of all edges $\{E\}$ of the graph as argument and outputs a value in $\mathcal{X}$.

Lemma 2. *Let $G = (V, E, f)$ and $G' = (V, E, f')$ be two non-isomorphic graphs vertex attributed graphs with functions $f : V \times \{E\} \to \mathcal{X}$ and $f' : V' \times \{E'\} \to \mathcal{X}$ respectively. The* homomorphism distortion *between G and G' is never 0 , i.e. $d_{HD} \neq 0$, if f is permutation invariant, injective, and $f = f'$.*

Again, we refer the readers to Section B.4 for the full proof. Now we join these results together to show the completeness of the homomorphism distortion.

Theorem 1. *Let $G = (V, E, f)$ and $G' = (V', E', f')$ be two vertex attributed graphs with attribute functions $f : V \times \{E\} \to \mathcal{X}$ and $f' : V' \times \{E'\} \to \mathcal{X}$. If the following conditions are fulfilled*

1. *$f = f'$*
2. *f is permutation invariant*
3. *f is injective*

then d_{HD} is complete.

Proof. We require that: **(I.)** If $G \simeq G'$ then $d_{\mathrm{HD}}(G, G') = 0$, which is shown in Lemma 1, and **(II.)** If $G \not\simeq G'$ then $d_{\mathrm{HD}}(G, G') > 0$, which is shown in Lemma 2 , the claim follows from both, as we have made the required assumptions. □

We also pose a relaxation of this theorem where we do not require that the functions f, f' take into account the whole set of edges $\{E\}$, but only the 1-hop neighborhood (immediate neighbors) of a node.

Proposition 3. *Let $G = (V, E, f)$ and $G' = (V', E', f')$ be two vertex attributed graphs. Denote set of neighborhoods of all nodes in a graph G as*

$$\mathcal{N}_G^* = \{\mathcal{N}_G(v) \mid v \in V\}$$

where $\mathcal{N}_G(v) = \{u \mid \{u, v\} \in E(G)\}$. The graphs are equipped with attribute functions $f : V \times \mathcal{N}_G \to \mathcal{X}$ and $f' : V' \times \mathcal{N}_{G'} \to \mathcal{X}$. If the previous conditions on f, f' are fulfilled, then the homomorphism distortion between G, G' and their attributes given by $d_{HD}(G, G')$ is complete.

This smaller side result is proven in Section B.5. Finally, recall that in Section 2 we use $\mathcal{F}$ to denote particular *classes* of graphs, and we establish the notion of $\mathcal{F}$ distinguishability. In Section 3, we go over how it can be used as a *proxy* to calculate properties over graphs. We can establish a bound on the *homomorphism distortion* in terms of an arbitrary *class* $\mathcal{F}$ as

Lemma 3. *Let $G = (V, E, f)$ and $G' = (V', E', f')ve$ be two vertex attributed graphs and let $\mathcal{F}$ be any arbitrary class of graphs. Then, $d_{\mathrm{HD}}(G, G')$ is an upper bound*

$$d_{\mathrm{HD}}(G, G') \geq |d_{\mathrm{HD}}(F, G) - d_{\mathrm{HD}}(F, G')| \tag{8}$$

for any $F \in \mathcal{F}$. This can be transformed into a tight bound in the case where $\mathcal{F} = \mathcal{G}_n$, the class of all graphs with no more than n vertices for $n = \max(|G|, |G'|)$, resulting in

$$d_{\mathrm{HD}}(G, G') = \max_{F \in \mathcal{F}} |d_{\mathrm{HD}}(F, G) - d_{\mathrm{HD}}(F, G')|. \tag{9}$$

We defer the proof to Section B.8. Conceptually, this result means that *nicer* or bigger class $\mathcal{F}$ helps approximate the true distortion. We conclude with

Proposition 4. *The* homomorphism distortion *is permutation invariant. Formally, for two vertex attributed graphs $G = (V, E, f)$ and $G' = (V', E', f')$, let ρ be a permutation $\rho(\cdot)$. Then,*

$$d_{\mathrm{HD}}(\rho(G), \rho(G')) = d_{\mathrm{HD}}(G, G') \tag{10}$$

The proof can be found in Section B.6. On a side note, notice that the "(pseudo)" prefix in every mention of the metric has not been in vain. We take a detour to show another additional, but not minor, result to finally ***... in the latent space bind them***. We show that we can tighten the distinguishability criteria of the homomorphism distortion from a pseudo-metric to a metric. Namely, we can define an equivalence relationship over isomorphisms. If two graphs are structurally isomorphic, we take them to be equivalent. We define a metric $d^*_{HD} : (\mathcal{G}/\simeq) \times (\mathcal{G}/\simeq) \to \mathcal{X}$ where $\mathcal{G}$ is the set of all graphs and $\mathcal{G}/\simeq$ is the quotient space with graph isomorphism as the equivalence relation.

Proposition 5. *The homomorphism distortion $d^*_{HD}(G, G')$ is a metric, i.e. the following properties hold for arbitrary vertex-attributed graphs G, G', G'' if we restrict the choices of f, f':*

1. $d^*_{HD}(G, G) = 0$
2. $d^*_{HD}(G, G') = d^*_{HD}(G', G)$

3. $d^*_{HD}(G, G') > 0$
4. $d^*_{HD}(G, G') \leq d^*_{HD}(G, G'') + d^*_{HD}(G'', G')$

We defer the proof to Section B.7. With that, we conclude our analysis of the properties of the *homomorphism distortion*. Now we turn our attention to how this metric can help us to characterize individual graphs.

4.4 Distortion as a characteristic of graphs

Our main contribution is to zoom in on the potential of the homomorphism distortion as a way to *characterize* graphs. We will define what it means to have a characteristic of a graph, or analogously obtain an embedding, and what makes a particular characterization *good*. It is relevant to recall that structural encodings can be understood from the angle of characterization. For example, consider the Random Walk Positional Encodings (RWPEs) [20], which assigns a vector to each node in a graph. This vector encodes specific structural properties of the graph and can help distinguish it from other graphs. However, it is limited in some instances. In the case of the RWPEs, it is known that it cannot distinguish strongly regular graphs with the same parameters [30].

Definition 11 (Graph embedding). *A* graph embedding *is a function that acts on a set of graphs $\mathcal{G}$ and embeds them into some normed space $\mathcal{H}$ defined as $\varsigma : \mathcal{G} \to \mathcal{H}$.*

From Definition 4, we say that a graph embedding ς is an *invariant graph embedding* if for all $G \simeq G' \in \mathcal{G}$ then $\varsigma(G) = \varsigma(G')$. Importantly, a *good* graph embedding should also be able to distinguish non-isomorphic graphs. We formalize this concept as

Definition 12 (Complete graph embedding). *A graph embedding $\varsigma : \mathcal{G} \to \mathcal{H}$ is complete on $\mathcal{G}$ if it is invariant and for all non-isomorphic $G, G' \in \mathcal{G}$ then $\varsigma(G) \neq \varsigma(G')$.*

Now, we show that the *homomorphism distortion* is also a complete graph embedding [7]. From now on, we rely on the previous assumptions over the functions over graphs f, f' [8]. The *homomorphism distortion* becomes an embedding as

Definition 13 (Homomorphism distortion embedding). *Let $G = (V, E, f)$ be a vertex attributed graph and let $\mathcal{F}$ be a class of graphs. The homomorphism distortion embedding is denoted as*

$$d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}} = [d_{\mathrm{HD}}(G, F)]_{F \in \mathcal{F}}$$

where $d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}}$ is a vector of *distortions* from G to each graph F in the class $\mathcal{F}$. This is an embedding since $d_{\mathrm{HD}}(G, \cdot)$ is some normed space, as it is a vector of elements of $\mathcal{X}$, which is itself a normed space. Now we proceed to show some properties of this graph embedding.

Lemma 4. *Let $G \in \mathcal{G}_n$ be a graph with attribute function f. Letting, $\mathcal{F} = \mathcal{G}_n$ the homomorphism distortion embedding $d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}}$ is invariant.*

[7] The homomorphism counts are indeed a case of a complete graph embedding
[8] We will refer to the more general case where $f : V \times \{E\} \to \mathcal{X}$ and the same for f'.

The full proof can be found at Section B.9 [9]. With this, we can extend the previous result to show that the *homomorphism distortion embedding* is also a complete graph embedding.

Theorem 2. *Let $G \in \mathcal{G}_n$ be a graphs with attribute functions f and let $\mathcal{F} = \mathcal{G}_n$. The homomorphism distortion embedding $d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}}$ is complete.*

Proof. Let $G' \in \mathcal{G}_n$ be a graph with attribute function f'. We want to show that $G \simeq G'$ if and only if $d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}} = d_{\mathrm{HD}}(G', \cdot)_{\mathcal{F}}$. We aim to extend Lemma 4 for the converse. This is easier if we approach the problem by contradiction. Assume that $d_{HD}(G, .) = d_{HD}(G', \cdot)$ that means that for all $F \in \mathcal{F}$ the following equality holds $d_{HD}(G, F) = d_{HD}(G, F)$. However, note that since $\mathcal{F} = \mathcal{G}_n$, then $G, G' \in \mathcal{G}$. Since that is the case, there is one $F = G$ resulting in the comparison between $d_{HD}(G, G)$ and $d_{HD}(G, G')$. By Proposition 1, $d_{HD}(G, G) = 0$ and since $G \not\simeq G'$ by Proposition 1 and Lemma 2 respectively then $d_{HD}(G', G) = d_{HD}(G, G') > 0$, thus we have reached a contradiction. □

We have shown that the homomorphism distortion serves as a powerful embedding method. However, due to the *graph canonization problem*, any complete graph embedding must be as hard (NP-Complete) to compute as solving the isomorphism problem. If not, then we could solve this problem and be able to distinguish isomorphic graphs directly. To make the homomorphism distortion embedding usable, we rely on approximating the distance while preserving its expressive properties.

4.5 Approximating the uncomputable

Naturally, we cannot compute the space of all graphs $\mathcal{F}$ or the set $\mathrm{Hom}(G, F)$ for any one of $F \in \mathcal{F}$. Thus, we resort to the techniques from Welke et al. [62] to approximate these functions. We propose parameterizing our embedding and we show we are *complete* in expectation. To do this, we define a *random graph embedding* as a graph embedding parametrized by a random variable X from a distribution $\mathcal{D}$ denoted $\varsigma_X : \mathcal{G} \to \mathcal{H}$. The resulting expectation $\mathbb{E}_{X \sim \mathcal{D}}[\varsigma_X] : \mathcal{G} \to \mathcal{H}$ gives us, ideally, our true embedding. We start with the following result

Theorem 3 (Welke et al. [62]). *Let $\varsigma_X : \mathcal{G} \to \mathcal{H}$ be an expectation-complete graph embedding and $G, G' \in \mathcal{G}_n$ which are not isomorphic. For any $\delta > 0$, there exists $L \in \mathbb{N}$ such that for all $\ell \geq L$*

$$(\varsigma_{X_1}(G), \ldots, \varsigma_{X_\ell}(G)) \neq (\varsigma_{X_1}(G'), \ldots, \varsigma_{X_\ell}(G'))$$

with probability $1 - \delta$, where $X_1, \ldots, X_\ell \sim D$ $i.i.d.$

With this, we have the required tools to show expectation-completeness of the homomorphism distortion embedding if we parametrize the class of graphs $\mathcal{F}$.

[9] Note that Lemma 4 also easily follows from Proposition 4.

Proposition 6. *Let G be a graph with attribute function f. Also, let $e_F \in \mathbb{R}^{\mathcal{F}}$ be the F-th standard basis unit-vector of $\mathbb{R}^{\mathcal{F}}$. Define the graph embedding resulting from the homomorphism distortion as*

$$\varsigma_F(G) = d_{\mathrm{HD}}(G, F)e_F.$$

If $\mathcal{D}$ is a distribution with with full support on $\mathcal{G}_n$ and $F \sim \mathcal{D}$, then the random graph embedding $\varsigma_F(\cdot)$ is expectation-complete on $\mathcal{G}_n$.

We show this proof in Section B.10. The results of Welke et al. [62] show that it is sufficient to sample $\ell = \mathcal{O}(\frac{n^2+\log(1/\delta)}{\epsilon^2})$ where $1-\delta$ is the probability that the bound

$$\big|\|\varsigma_{\mathcal{F}}(G) - \varsigma_{\mathcal{F}}(G')\|^2 - \|\sqrt{Q}(\varsigma_{\mathcal{G}_n}(G) - \varsigma_{\mathcal{G}_n}(G'))\|^2\big| < \epsilon$$

holds. Analogously, we can extend the previous parametrization as

Definition 14. *A random graph embedding $\varsigma_{X,Y,Z} : \mathcal{G} \to \mathcal{H}$ parametrized by $X \sim \mathcal{D}$, $Y \sim \mathcal{Q}$ and $Z \sim \mathcal{U}$ is expectation-complete if the graph embedding $\mathbb{E}_{X,Y,Z}[\varsigma_{X,Y,Z}]$ is complete.*

We aim to use the additional random variables to parametrize the sampling of homomorphisms from a graph G to a sampled graph F. For this case, we aim to sample a subset $S \subset \mathrm{Hom}(G, F)$ such that $|S| \ll |\mathrm{Hom}(G, F)|$ and $S' \subset \mathrm{Hom}(F, G)$ such that $|S'| \ll |\mathrm{Hom}(F, G)|$. As such, we define this parametrization as

$$\varsigma_{F,H_{G,F},H_{F,G}}(G) = \max\Big(\inf_{\varphi \in H_{G,F}} \Delta(\varphi), \inf_{\varphi' \in H_{F,G}} \Delta(\varphi')\Big) \tag{11}$$

where the $H_{G,F}$ and $H_{F,G}$ are sets of ℓ samples from a distribution over the corresponding homomorphisms and $\Delta(\varphi), \Delta(\varphi')$ are as defined in Definition 8. Formally, for a distribution $\mathcal{Q}$ with full support on $\mathrm{Hom}(G, F)$ and a distribution $\mathcal{U}$ with full support on $\mathrm{Hom}(F, G)$ we have that

$$H_{G,F} = \{\varphi_1, \dots, \varphi_\ell\} \sim \mathcal{Q}, \quad H_{F,G} = \{\varphi'_1, \dots, \varphi'_\ell\} \sim \mathcal{U} \tag{12}$$

which allows for the following proposition to be constructed

Theorem 4. *Let $G = (V, E, f)$ be an attributed graph, $F \sim \mathcal{D}$ be a graph sampled from $\mathcal{D}$ with an attribute function f'. Also, let $H_{G,F} \sim \mathcal{Q}$ and $H_{F,G} \sim \mathcal{U}$ be two sets of ℓ samples from the distributions $\mathcal{Q}$,$\mathcal{U}$ conditioned on the sample F. Then, the random graph embedding $\varsigma_{F,H_{G,F},H_{F,G}}(\cdot)$ resulting from this parametrization of the homomorphism distortion embedding is expectation-complete on $\mathcal{G}_n$.*

This is the only main result for which the proof is deferred to Section B.11 due to its extension. However, the intuition follows from the previous result. It relies on the fact that if F' is one of G or G', then, with enough samples, the identity is contained in the homomorphisms which results in different vectors for each graph.

5 Experimental results

This section serves to empirically validate some of the properties of the *homomorphism distortion*. Both as a measurement of difference and as a graph embedding. We aim to answer the following questions

Q1. To what extend can the *homomorphism distortion* distinguish graphs in hard k-WL classes ?
Q2. Does the theoretical graph characterization power extend to learning scenarios ?
Q3. How good does *expectation-completeness* work practically ?

Functions over the graphs. To fulfill our assumptions, we enforce the following conditions over the function over graphs: **(a)** The function f is the same attribute function on all graphs, **(b)** f is permutation invariant, and **(c)** f is injective. Constructing such a function f is an interesting endeavor on it's own. To simplify things, we rely on popular characterization methods for graphs, denoted *positional and structural encodings* (PSEs) [29, 31]. Concretly we choose *Random Walk Positional Encodings* (RWPE) [20] and *Shortest Path Encodings* (SPE). The latter is just a vector with the size of the shortest path to each node.

Classes of graphs $\mathcal{F}$. The fundamental criteria for choosing $\mathcal{F}$ is computability. We decide to use graphs of treewidth k, for $k \in \{1, 2\}$. In practice, this results in $\mathcal{F}^{(1)} = \{\text{the set of cycles up to order } n\}$ and $\mathcal{F}^{(2)} = \{\text{the set of non-isomorphic trees of order } n\}$. For both cases n is treated as a hyperparameter and we perform experiments for different values of n. We acknowledge that in Jin et al. [30] and Bao et al. [3] authors use the more expressive class $\mathrm{Spasm}(\cdot)$ of cycle graphs. Since in this setting we are concerned with homomorphism enumeration sampling and not counts, generating these enumerations is algorithmically complex. As such, we defer this to future works.

Comparison method. In both of the following cases each graph G has a corresponding *homomorphism distortion embedding* for a given graph class $\mathcal{F}$ defined as

$$d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}} = [d_{\mathrm{HD}}(G, F)]_{F \in \mathcal{F}} \in \mathbb{R}^M \tag{13}$$

where $M = |\mathcal{F}|$ and we assume that the co-domain of f is $\mathbb{R}$. Please refer to Section C for hardware details and the full list of hyperparameters.

5.1 Non-learning scenarios

To measure the distinguishing power of the *homomorphism distortion* we employ the expressivity benchmark dataset BREC [61] [10]. Recall from Proposition 2 that we can approximate the *homomorphism distortion* between two graphs G, G' through the distances to a class $\mathcal{F}$ as

$$d_{\mathrm{HD}}(G, G') \approx \max_{F \in \mathcal{F}} |d_{\mathrm{HD}}(G, F) - d_{\mathrm{HD}}(G', F)| \tag{14}$$

[10] The difficulty in terms of k-WL distinguishability ranges from $k = 1$ to $k > 4$.

Following the methodology of Ballester and Rieck [2], we calculate our measure and establish a threshold at $\epsilon < 1 \times 10^{-3}$ for which graphs are deemed to be isomorphic[11]. To ensure fairness across different choices of f, we normalize the final distortion to the interval $[0, 1]$. Table 1 shows the results in distinguishing each class of graphs. First, *homomorphism distortion* on $\mathcal{F}_1$ achieves a high distinction rate. Interestingly, even in the harder instances of C, 4 and D. On the best configuration the method distinguishes all graphs. Something noteworthy is that if with SPE increasing the number the order of cycles considered (from $\mathcal{F}_1$ to $\mathcal{F}_2$, the accuracy increases significantly, suggesting that the shortest path distances induced by these cycles are vital for the task).

Table 1: The percentage of distinguished non-isomorphic graphs in the BREC dataset. The variables are: $\mathcal{F}_1 = \{C_3, C_4, C_5, C_8\}$, $\mathcal{F}_2 = \{C_3, C_4, C_5, C_8, C_9, C_{10}\}$, $\mathcal{F}_3 = \{T \mid \mathrm{ord}(T) = 8\}$. The $d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}}$ achieves a very high distinction rate compared to non-GNN methods 3-WL and the best result on *persistent homology* methods (PH).

Method	B	R	E	C	4	D
3-WL [61]	**100.0**	37.0	**100.0**	60.0	0.0	0.0
PH [2]	**100.0**	97.0	**100.0**	6.0	**100.0**	5.0
$f = \mathrm{RWSE}(\cdot)$						
$d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}_1}$	99.0	36.9	59.1	50.8	55.8	98.9
$d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}_2}$	99.0	36.9	59.1	50.8	55.8	98.9
$d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}_3}$	**100.0**	99.95	96.2	82.4	**100.0**	**100.0**
$f = \mathrm{SP}(\cdot)$						
$d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}_1}$	99.0	36.9	59.1	50.8	55.8	98.9
$d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}_2}$	**100.0**	**100.0**	99.9	99.8	**100.0**	**100.0**
$d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}_3}$	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**

Figure 1 shows a comparison on the particularly difficult class of 4-vertex distance-regular graphs. Note that previous approaches have a very low distinction rate. We include the full comparison of the distance matrices in Section D.

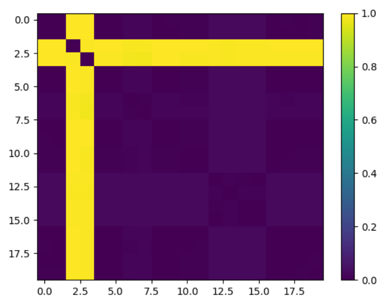
5.2 Learning scenarios

We use the three models in our experiments: namely GCN, GAT and GIN. We follow the following procedure. In the case of the inputs given by ZINC-12k we denote this transformation as $\Psi : \mathbb{N} \to \mathbb{R}^{28 \times d}$ (a common place technique) [19]. For the *homomorphism distortion embedding* we define a function $\omega(\cdot) : \mathbb{R}^M \to \mathbb{R}^d$ which we parametrize and learn with a 2-layer MLP. These changes result in the following change to the architecture

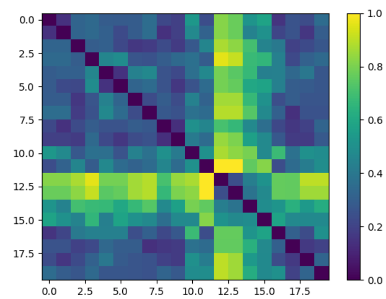
$$\mathbf{h}_v^0 = \mathrm{CONCAT}\Big(\Psi(f(v)),\ \omega(d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}})\Big) \tag{15}$$

where the initial feature of the vertex v is denoted with $f(v)$. Notice that we assign the same *homomorphism distortion embedding* to all vertices. In Section 6 we reflect on this

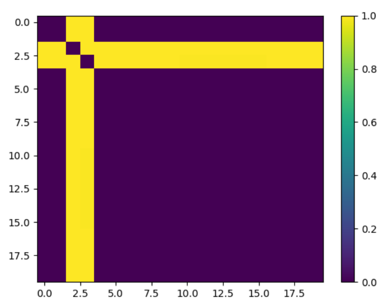
[11] Unless we raise the value to something absurd such as 2×10^{-1}, the results stay the same.



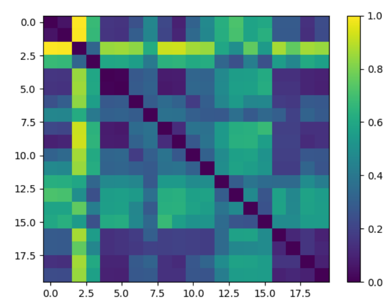
(a) $\mathcal{F} = \{C_3, \cdots, C_8,\}$ and $f = \mathrm{SPE}(\cdot)$



(b) $\mathcal{F} = \{T \mid \mathrm{ord}(T) = 8\}$ and $f = \mathrm{SPE}(\cdot)$



(c) $\mathcal{F} = \{C_3, \cdots, C_8,\}$ and $f = \mathrm{RWPE}(\cdot)$



(d) $\mathcal{F} = \{T \mid \mathrm{ord}(T) = 8\}$ and $f = \mathrm{RWPE}(\cdot)$

Fig. 1: Resulting distance matrices from $d_{\mathrm{DH}}(\cdot)$ using $\sigma = 10$ and $\gamma = 10$ on class 4 of `BREC` containing 4-vertex distance-regular graphs. The top row shows the evaluation on shortest paths Figure 1a and Figure 1b and the bottom on random walks Figure 1c and Figure 1d. Notice that cycles are not enough to distinguish this class even when using a "better" f.

decision. We also test a variation where we *additionally add* the homomorphism counts and the subgraph counts. In summary, these are the three tested conditions:

- $d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}}$: This is the *homomorphism distortion embedding*.
- $d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}} + \mathrm{Hom}$: This is the *homomorphism distortion embedding* concatenated with the homomorphism counts.
- $d_{\mathrm{HD}}(G, \cdot)_{\mathcal{F}} + \mathrm{Sub}$: This is the *homomorphism distortion embedding* concatenated with the subgraph counts.

Results report and discussion. We train the models over 5 random weight initializations and report the mean and standard deviation as $\mu \pm \sigma$. We choose the popular dataset `ZINC-12K` to validate our finds. Table 2 shows comparison in the same setup to Jin et al. [30]. Table 5 shows a comparison with the state-of-the-art methods using graph theoretic inductive biases [12]. Overall, we perform slightly better than homomorphism

[12] We only take into account reported scores *without using edge features*.

counts. However, if we augment the method with the counts of either subgraphs or homomorphisms we achieve the highest performance.

Table 2: The MAE for graph regression on `ZINC-12k` (without edge features) for $\mathcal{F} = \{C_3, \cdots, C_8\}$ on 500K parameters. Our method *outperforms* homomorphism counts. If we combine it with homomorphism counts (on GCN) or subgraph counts (on GIN) it *outperforms* subgraph counts. The ranking is denoted in best, second, third.

	GAT	GCN	GIN
Base	0.380 ± 0.009	0.282 ± 0.007	0.246 ± 0.019
Hom [30]	0.229 ± 0.008	0.235 ± 0.005	0.197 ± 0.015
Sub [30]	0.201 ± 0.004	0.198 ± 0.003	0.143 ± 0.002
$d_{HD}(G, \cdot)_{\mathcal{F}}$ (Ours)	0.217 ± 0.004	0.217 ± 0.004	0.184 ± 0.005
$d_{HD}(G, \cdot)_{\mathcal{F}}$ + Hom (Ours)	**0.171 ± 0.002**	0.181 ± 0.004	0.159 ± 0.004
$d_{HD}(G, \cdot)_{\mathcal{F}}$ + Sub (Ours)	0.187 ± 0.022	**0.176 ± 0.007**	**0.138 ± 0.004**

6 Conclusions

In this work, we aim to question the current expressivity paradigm in graph representation learning, highlighting its flaws, and propose a first step to address them. The current trend of expressivity measurement relies almost solely on combinatorial notions from graph theory. These notions, while relevant and useful, generally do not live up to expectations in practice. As such, we come up with a way to measure distances between *vertex attributed graphs*. This (pseudo)-metric is reminescent of the Haussdorf distance and aims to capture a continuous notion of dissimilarity. Under certain assumptions, we show the following properties. **a)** this (pseudo)-metric fully characterizes graphs (*completeness*) **b)** we can re-frame it as an embedding which results in a fully characterizing embedding (*complete graph embedding*) **c)** we show we can construct a metric from it by taking the natural equivalence relationship of isomorphic **d)** we show that the resulting embedding is *expectation-complete* when we lack the computational power to directly calculate it . Finally, we empirically validate the effectiveness of this framework on an expressivity dataset and in a learning scenario.

Limitation and future works. In this work, we have only scratched the surface of this new paradigm in graph representation learning. First, we make certain assumptions on the types of attributed graphs we can work with to achieve completeness. Obtaining more general results is always a valuable avenue for the future. Additionally, and for practical reasons, we use positional and structural encoding as functions over the graph; however, better functions surely exist. Also, we do not provide a theoretical characterization of which graphs can be distinguished by this method or relate it to underlying graph properties. Empirically, we also test a limited variation of classes $\mathcal{F}$ and hyperparameters. However, this is instead promising given the results. To conclude, a significant shift in graph representation learning is needed to strip the field of antiquated measurements that have little practical equivalence. There is already awareness of these flaws; however, steps need to be taken to fix them. In this work, we propose a promising direction to start searching for these answers.

Acknowledgments. First of all, I want to thank Olga for all the incredible help and support the last year. It has been a pleasure to be mentored by such a kind and smart person. Then I want to thank Erik because if not for having had him as a professor and mentor, I would not have had the opportunity to cultivate my academic passion. Last but not least, I want to thank Bastian for the continuous mentoring and encouragement during this process. On a less academic note, I want to thank all the friends I made in Amsterdam. I took great pleasure in the academic talks and found solace in sharing a part of my life with you.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Vikraman Arvind et al. “On Weisfeiler-Leman Invariance: Subgraph Counts and Related Graph Properties”. en. In: *Fundamentals of Computation Theory*. Ed. by Leszek Antoni Gasieniec, Jesper Jansson, and Christos Levcopoulos. Vol. 11651. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 111–125. ISBN: 978-3-030-25026-3 978-3-030-25027-0. DOI: 10.1007/978-3-030-25027-0_8. URL: https://link.springer.com/10.1007/978-3-030-25027-0_8 (visited on 02/28/2025).

[2] Rubén Ballester and Bastian Rieck. *On the Expressivity of Persistent Homology in Graph Learning*. en. arXiv:2302.09826 [cs]. June 2024. DOI: 10.48550/arXiv.2302.09826. URL: http://arxiv.org/abs/2302.09826 (visited on 12/01/2024).

[3] Linus Bao et al. *Homomorphism Counts as Structural Encodings for Graph Learning*. en. arXiv:2410.18676 [cs]. Feb. 2025. DOI: 10.48550/arXiv.2410.18676. URL: http://arxiv.org/abs/2410.18676 (visited on 04/22/2025).

[4] Pablo Barceló et al. “Graph neural networks with local graph parameters”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 25280–25293.

[5] Alaa Bessadok, Mohamed Ali Mahjoub, and Islem Rekik. “Graph neural networks in network neuroscience”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.5 (2022), pp. 5833–5848.

[6] H. L. Bodlaender and A. M. C. A. Koster. “Combinatorial Optimization on Graphs of Bounded Treewidth”. en. In: *The Computer Journal* 51.3 (Nov. 2007), pp. 255–269. ISSN: 0010-4620, 1460-2067. DOI: 10.1093/comjnl/bxm037. URL: https://academic.oup.com/comjnl/article-lookup/doi/10.1093/comjnl/bxm037 (visited on 02/17/2025).

[7] Hans L. Bodlaender. “A partial k-arboretum of graphs with bounded treewidth”. en. In: *Theoretical Computer Science* 209.1-2 (Dec. 1998), pp. 1–45. ISSN: 03043975. DOI: 10.1016/S0304-3975(97)00228-4. URL: https://linkinghub.elsevier.com/retrieve/pii/S0304397597002284 (visited on 02/13/2025).

[8] Hans L. Bodlaender, Paul Bonsma, and Daniel Lokshtanov. “The Fine Details of Fast Dynamic Programming over Tree Decompositions”. en. In: *Parameterized and Exact Computation*. Ed. by David Hutchison et al. Vol. 8246. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2013, pp. 41–53. ISBN: 978-3-319-03897-1 978-3-319-03898-8. DOI: 10.1007/978-3-319-03898-8_5. URL: http://link.springer.com/10.1007/978-3-319-03898-8_5 (visited on 02/27/2025).

[9] Giorgos Bouritsas et al. “Improving Graph Neural Network Expressivity via Subgraph Isomorphism Counting”. en. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.1 (Jan. 2023), pp. 657–668. ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2022.3154319. URL: https://ieeexplore.ieee.org/document/9721082/ (visited on 05/26/2025).

[10] Andries E Brouwer and Willem H Haemers. “Strongly regular graphs”. In: *Spectra of graphs*. Springer, 2011, pp. 115–149.

[11] Davide Buffelli, Farzin Soleymani, and Bastian Rieck. *CliquePH: Higher-Order Information for Graph Neural Networks through Persistent Homology on Clique Graphs*. en. arXiv:2409.08217 [cs]. Nov. 2024. DOI: 10.48550/arXiv.2409.08217. URL: http://arxiv.org/abs/2409.08217 (visited on 02/02/2025).

[12] Jan Böker et al. “The Complexity of Homomorphism Indistinguishability”. en. In: *LIPIcs, Volume 138, MFCS 2019* 138 (2019). Artwork Size: 13 pages, 540587 bytes ISBN: 9783959771177 Medium: application/pdf Publisher: Schloss Dagstuhl – Leibniz-Zentrum für Informatik Version Number: 1.0, 54:1–54:13. ISSN: 1868-8969. DOI: 10.4230/LIPICS.MFCS.2019.54. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.MFCS.2019.54 (visited on 02/13/2025).

[13] Zhengdao Chen et al. “Can graph neural networks count substructures?” In: *Advances in neural information processing systems* 33 (2020), pp. 10383–10395.

[14] Zhengdao Chen et al. “On the equivalence between graph isomorphism testing and function approximation with gnns”. In: *Advances in neural information processing systems* 32 (2019).

[15] Luca Cosmo et al. “Graph Kernel Neural Networks”. en. In: *IEEE Transactions on Neural Networks and Learning Systems* (2024). arXiv:2112.07436 [cs], pp. 1–14. ISSN: 2162-237X, 2162-2388. DOI: 10.1109/TNNLS.2024.3400850. URL: http://arxiv.org/abs/2112.07436 (visited on 02/28/2025).

[16] Radu Curticapean, Holger Dell, and Dániel Marx. “Homomorphisms Are a Good Basis for Counting Small Subgraphs”. en. In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. arXiv:1705.01595 [cs]. June 2017, pp. 210–223. DOI: 10.1145/3055399.3055502. URL: http://arxiv.org/abs/1705.01595 (visited on 02/13/2025).

[17] Josep Daaz, Maria Serna, and Dimitrios M Thilikos. “Counting H -colorings of partial k-trees”. en. In: *Theoretical Computer Science* (2002).

[18] Holger Dell, Martin Grohe, and Gaurav Rattan. *Lovász Meets Weisfeiler and Leman*. en. arXiv:1802.08876 [cs]. May 2018. DOI: 10.48550/arXiv.

1802.08876. URL: http://arxiv.org/abs/1802.08876 (visited on 02/17/2025).

[19] Vijay Prakash Dwivedi et al. “Benchmarking graph neural networks”. In: *Journal of Machine Learning Research* 24.43 (2023), pp. 1–48.

[20] Vijay Prakash Dwivedi et al. “Graph neural networks with learnable structural and positional representations”. In: *arXiv preprint arXiv:2110.07875* (2021).

[21] Yam Eitan et al. “Topological blindspots: Understanding and extending topological deep learning through the lens of expressivity”. In: *arXiv preprint arXiv:2408.05486* (2024).

[22] Fedor V Fomin, Pinar Heggernes, and Dieter Kratsch. “Exact algorithms for graph homomorphisms”. In: *Theory of Computing Systems* 41.2 (2007), pp. 381–393.

[23] Fabrizio Frasca et al. “Understanding and extending subgraph gnns by rethinking their symmetries”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 31376–31390.

[24] Jingchu Gai et al. “Homomorphism expressivity of spectral invariant graph neural networks”. In: *arXiv preprint arXiv:2503.00485* (2025).

[25] Justin Gilmer et al. “Neural message passing for quantum chemistry”. In: *International conference on machine learning*. Pmlr. 2017, pp. 1263–1272.

[26] Martin Grohe. *Descriptive complexity, canonisation, and definable graph structure theory*. Vol. 47. Cambridge University Press, 2017.

[27] Martin Grohe. “The complexity of homomorphism and constraint satisfaction problems seen from the other side”. en. In: *Journal of the ACM* 54.1 (Mar. 2007), pp. 1–24. ISSN: 0004-5411, 1557-735X. DOI: 10.1145/1206035.1206036. URL: https://dl.acm.org/doi/10.1145/1206035.1206036 (visited on 02/17/2025).

[28] Pavol Hell and Jaroslav Nešetřil. “On the complexity of H-coloring”. In: *Journal of Combinatorial Theory, Series B* 48.1 (1990), pp. 92–110.

[29] Yinan Huang et al. “On the stability of expressive positional encodings for graphs”. In: *arXiv preprint arXiv:2310.02579* (2023).

[30] Emily Jin et al. *Homomorphism Counts for Graph Neural Networks: All About That Basis*. en. arXiv:2402.08595 [cs]. June 2024. DOI: 10.48550/arXiv.2402.08595. URL: http://arxiv.org/abs/2402.08595 (visited on 04/28/2025).

[31] Charilaos I Kanatsoulis et al. “Learning efficient positional encodings with graph neural networks”. In: *arXiv preprint arXiv:2502.01122* (2025).

[32] Sandra Kiefer. “The Weisfeiler-Leman algorithm: an exploration of its power”. In: *ACM SIGLOG News* 7.3 (2020), pp. 5–27.

[33] TN Kipf. “Semi-Supervised Classification with Graph Convolutional Networks”. In: *arXiv preprint arXiv:1609.02907* (2016).

[34] Matthias Lanzinger and Pablo Barceló. *On the Power of the Weisfeiler-Leman Test for Graph Motif Parameters*. en. arXiv:2309.17053 [cs]. Mar. 2024. DOI: 10.48550/arXiv.2309.17053. URL: http://arxiv.org/abs/2309.17053 (visited on 04/28/2025).

[35] Xuyuan Liu et al. "Exploring consistency in graph representations: from graph kernels to graph neural networks". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 136393–136423.

[36] L. Lovász. "Operations with structures". In: *Acta Mathematica Academiae Scientiarum Hungarica* 18.3 (1967), pp. 321–328. ISSN: 1588-2632. DOI: 10.1007/BF02280291. URL: https://doi.org/10.1007/BF02280291.

[37] László Lovász. *Large Networks and Graph Limits*. en. Vol. 60. Colloquium Publications. Providence, Rhode Island: American Mathematical Society, Dec. 2012. ISBN: 978-0-8218-9085-1 978-1-4704-1583-9. DOI: 10.1090/coll/060. URL: http://www.ams.org/coll/060 (visited on 04/28/2025).

[38] Takanori Maehara and Hoang NT. "Deep homomorphism networks". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 56076–56107.

[39] Takanori Maehara and Hoang NT. *Learning on Random Balls is Sufficient for Estimating (Some) Graph Parameters*. en. arXiv:2111.03317 [cs]. Nov. 2021. DOI: 10.48550/arXiv.2111.03317. URL: http://arxiv.org/abs/2111.03317 (visited on 02/13/2025).

[40] Kelly Maggs, Celia Hacker, and Bastian Rieck. "Simplicial representation learning with neural k-forms". In: *arXiv preprint arXiv:2312.08515* (2023).

[41] Haggai Maron et al. "On the universality of invariant networks". In: *International conference on machine learning*. PMLR. 2019, pp. 4363–4371.

[42] Christopher Morris et al. "Position: Future directions in the theory of graph machine learning". In: *Forty-first International Conference on Machine Learning*. 2024.

[43] Christopher Morris et al. "Weisfeiler and leman go machine learning: The story so far". In: *Journal of Machine Learning Research* 24.333 (2023), pp. 1–59.

[44] Christopher Morris et al. "Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks". en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (July 2019). Number: 01, pp. 4602–4609. ISSN: 2374-3468. DOI: 10.1609/aaai.v33i01.33014602. URL: https://ojs.aaai.org/index.php/AAAI/article/view/4384 (visited on 06/02/2025).

[45] Christopher Morris et al. *WL meet VC*. en. arXiv:2301.11039 [cs]. May 2023. DOI: 10.48550/arXiv.2301.11039. URL: http://arxiv.org/abs/2301.11039 (visited on 02/28/2025).

[46] Daniel Neuen. *Homomorphism-Distinguishing Closedness for Graphs of Bounded Tree-Width*. en. arXiv:2304.07011 [math]. July 2023. DOI: 10.48550/arXiv.2304.07011. URL: http://arxiv.org/abs/2304.07011 (visited on 02/17/2025).

[47] Hoang Nguyen and Takanori Maehara. "Graph homomorphism convolution". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 7306–7316.

[48] Karolina Okrasa and Paweł Rzążewski. *Fine-grained complexity of the graph homomorphism problem for bounded-treewidth graphs*. en. arXiv:1906.08371 [cs]. Feb. 2020. DOI: 10.48550/arXiv.1906.08371. URL: http://arxiv.org/abs/1906.08371 (visited on 02/13/2025).

[49] Theodore Papamarkou et al. "Position: Topological deep learning is the new frontier for relational learning". In: *Proceedings of machine learning research* 235 (2024), p. 39529.

[50] Paolo Pellizzoni, Till Hendrik Schulz, and Karsten Borgwardt. "Graph Neural Networks Can (Often) Count Substructures". In: *The Thirteenth International Conference on Learning Representations*. 2025.

[51] Omri Puny et al. "Equivariant polynomials for graph neural networks". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 28191–28222.

[52] Patrick Reiser et al. "Graph neural networks for materials science and chemistry". In: *Communications Materials* 3.1 (2022), p. 93.

[53] Pedro Ribeiro et al. "A survey on subgraph counting: concepts, algorithms, and applications to network motifs and graphlets". In: *ACM computing surveys (csur)* 54.2 (2021), pp. 1–36.

[54] Martin JA Schuetz, J Kyle Brubaker, and Helmut G Katzgraber. "Combinatorial optimization with physics-inspired graph neural networks". In: *Nature Machine Intelligence* 4.4 (2022), pp. 367–377.

[55] Joshua Southern et al. "Balancing Efficiency and Expressiveness: Subgraph GNNs with Walk-Based Centrality". In: *arXiv preprint arXiv:2501.03113* (2025).

[56] Matteo Togninalli et al. "Wasserstein weisfeiler-lehman graph kernels". In: *Advances in neural information processing systems* 32 (2019).

[57] Victor Toscano-Duran, Florian Rottach, and Bastian Rieck. "Molecular Machine Learning Using Euler Characteristic Transforms". In: *arXiv preprint arXiv:2507.03474* (2025).

[58] Edwin R. Van Dam and Willem H. Haemers. "Which graphs are determined by their spectrum?" en. In: *Linear Algebra and its Applications* 373 (Nov. 2003), pp. 241–272. ISSN: 00243795. DOI: `10.1016/S0024-3795(03)00483-X`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S002437950300483X` (visited on 02/27/2025).

[59] Petar Veličković et al. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017).

[60] Qing Wang et al. "N-WL: A New Hierarchy of Expressivity for Graph Neural Networks". In: *The Eleventh International Conference on Learning Representations*. 2023.

[61] Yanbo Wang and Muhan Zhang. *An Empirical Study of Realized GNN Expressiveness*. en. arXiv:2304.07702 [cs]. June 2024. DOI: `10.48550/arXiv.2304.07702`. URL: `http://arxiv.org/abs/2304.07702` (visited on 04/09/2025).

[62] Pascal Welke et al. "Expectation-complete graph representations with homomorphisms". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 36910–36925.

[63] Hinrikus Wolf et al. *Structural Node Embeddings with Homomorphism Counts*. en. arXiv:2308.15283 [cs]. Nov. 2023. DOI: `10.48550/arXiv.2308.15283`. URL: `http://arxiv.org/abs/2308.15283` (visited on 05/26/2025).

[64] Keyulu Xu et al. "How powerful are graph neural networks?" In: *arXiv preprint arXiv:1810.00826* (2018).

[65] Zuoyu Yan et al. “An efficient subgraph gnn with provable substructure counting power”. In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2024, pp. 3702–3713.

[66] Bingxu Zhang et al. “The Expressive Power of Graph Neural Networks: A Survey”. en. In: *IEEE Transactions on Knowledge and Data Engineering* 37.3 (Mar. 2025). arXiv:2308.08235 [cs], pp. 1455–1474. ISSN: 1041-4347, 1558-2191, 2326-3865. DOI: 10.1109/TKDE.2024.3523700. URL: http://arxiv.org/abs/2308.08235 (visited on 04/28/2025).

[67] Bohang Zhang et al. “A complete expressiveness hierarchy for subgraph gnns via subgraph weisfeiler-lehman tests”. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 41019–41077.

[68] Bohang Zhang et al. “Beyond weisfeiler-lehman: A quantitative framework for gnn expressiveness”. In: *arXiv preprint arXiv:2401.08514* (2024).

A Extended background

A.1 Graph theory

We introduce the notion of *indistiguishability* to denote that a homomorphism that counts over a particular class $\mathcal{F}$ cannot distinguish two graphs as

Graph Coloring. A k-coloring of a graph is an assignment of a color to each vertex of a graph from a set of k possible colors such that no two adjacent nodes have the same color. The k coloring of a graph G can be found in $\mathcal{O}^*(k^{\mathrm{tw}(G)})$ if the tree decomposition of a graph G of width $\mathrm{tw}(G)$ is given Bodlaender and Koster [6].

Graph cores. A graph G is called a *core* if for every proper subgraph H there is no homomorphism from G to H. If two graphs G, G' are homomorphically equivalent, then finding whether there exists a homomorphism between an arbitrary graph H to G is equivalent to finding one from H to G'. Since every graph is homomorphically equivalent to its core, the problem of finding homomorphisms can be restricted to cores

Definition 15 **($\mathcal{F}$-Indistinguishability).** *Let G, G' be two graphs. Let $F \in \mathcal{F}$ be a class of graphs. If for all $F \in \mathcal{F}$ it holds that $\hom(F, G) = \hom(F, G)$ then G and H are called $\mathcal{F}$-indistinguishable.*

Homomorphic image. For a graph G, an homomorphic image is a graph G' obtained by merging two non-adjacent vertex. $\mathrm{Spasm}(G)$ is the set of all such homomorphic images of G.

A.2 Graph Neural Networks

Graph Convolutional Networks [33]. Let $G = (V, E, f)$ be a vertex attributed graph. We can represent the relationships between edges in G using a matrix A, termed an *adjacency matrix*, using the following definition

$$A_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \in E \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

where $A \in \{0,1\}^{|V|\times|V|}$ and $v_i, v_j \in V$ are the i-th and j-th vertices of V. Additionally, let I_N be the $N \times N$ identity matrix where $N = |V|$. Now we define the *Graph Convolutional Network* hidden representation update as

$$\mathbf{H}^{\ell+1} = \sigma(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}\mathbf{H}^{\ell}\mathbf{W}^{\ell}) \tag{17}$$

where $\tilde{A} = A + I_N$ is the *adjacency matrix* with self-loops, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ is the *diagonal degree matrix*, $\mathbf{H}^{\ell} \in \mathbb{R}^{N\times C}$ is the hidden representation matrix of N vectors of size C at layer ℓ and $\mathbf{W}^{\ell} \in \mathbb{R}^{C\times H}$ is the learnable weight matrix and H is the hidden dimensions. To simplify the definitions, we assume that $f : V \to \mathbb{R}^C$, where C is the feature dimensions. Now, we can denote the features matrix of G as $X \in \mathbb{R}^{N\times C}$, where the i-th vertex has feature $X_i = f(v_i)$. Note that if we define $\hat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ then Equation (17) can be restated in a simplified version as

$$\mathbf{H}^{\ell+1} = \sigma(\hat{A}\mathbf{H}^{\ell}\mathbf{W}^{\ell}) \tag{18}$$

Graph Isomorphism Network [64] . The Graph Isomorphism Network (GIN) is a network that aims to realize the maximum expressivity potential using the GNN framework. The *node update function* is defined as

$$\mathbf{h}_v^{\ell+1} = \mathrm{MLP}^{\ell+1}\left(\left(1+\epsilon^{\ell+1}\right)\cdot \mathbf{h}_v^{\ell} + \sum_{u\in\mathcal{N}(v)} \mathbf{h}_u^{\ell}\right) \tag{19}$$

where ϵ can be a scalar or a learnable parameter.

Graph Attention Network [59]. The Graph Attention Network (GAT) is, in essence, the addition of an attention coefficient to the general message passing framework in Equation (1). The node updated is denoted by

$$\mathbf{h}_v^{\ell+1} = \sigma\left(\sum_{u\in\mathcal{N}(v)} \alpha_{vu}\mathbf{W}^{\ell}\mathbf{h}_u^{\ell}\right) \tag{20}$$

where α_{vu} is a normalization of the *attention coefficient* between v, u. The attention coefficient is defined as

$$e_{vu} = a(\mathbf{W}^{\ell}\mathbf{h}_v^{\ell}, \mathbf{W}^{\ell}\mathbf{h}_u^{\ell}) \tag{21}$$

where $a : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the shared self-attention function and the normalization is performed as

$$\alpha_{vu} = \mathrm{softmax}_u(e_{vu}) = \frac{\exp(e_{vu})}{\sum_{w\in\mathcal{N}(v)}\exp(e_{vw})} \tag{22}$$

for all pairs of edges $\{v, u\} \in E$. Note that the actual implementation differs slightly as multiple heads are used, which results in a repetition of k times Equation (20), which are concatenated together to achieve the final results.

B Proofs

B.1 Proof of Proposition 1

Proposition B.1. *The graph homomorphism distortion $d_{\mathrm{HD}}(G, G')$ is a pseudo-metric, i.e. the following properties hold for arbitrary vertex-attributed graphs G, G', G'' that share the same attribute function co-domain*

1. $d_{HD}(G, G') \geq 0$ *and* $d_{HD}(G, G) = 0$
2. $d_{HD}(G, G') = d_{HD}(G', G)$
3. $d_{HD}(G, G') \leq d_{HD}(G, G'') + d_{HD}(G'', G')$

Proof. Since norms are non-negative, we have that $d_{\mathrm{HD}}(G, G') \geq 0$. If $G = G'$, the identity mapping $f(v) = v$ is in $\mathrm{Hom}(G, G)$, minimizing the right-hand side of Definition 7, which evaluates to 0. Thus, Item 1 holds. As for the symmetry, Definition 7 evaluates terms involving $\mathrm{Hom}(G, G')$ and $\mathrm{Hom}(G', G)$ respectively, and taking their maximum. This operation is inherently symmetric, so switching G and G' has no effect. Thus, d_{HD} is symmetric and Item 2 holds. Finally, to prove the triangle inequality, i.e., Item 3, we make use of the fact that we can compose homomorphisms. Without loss of generality, we assume that G, G', and G'' are homomorphically equivalent, i.e., there is at least one homomorphism between arbitrary pairs of these graphs. Taking any two such homomorphisms $\varphi : G \to G''$ and $\varphi' : G'' \to G'$ their composition $\varphi' \circ \varphi$ is a homomorphism from G to G'. As such, it occurs in the calculation of $d_{\mathrm{HD}}(G, G')$. Due to the properties of the norm used in Definition 7, splitting the composition into φ and φ' is always lower bounded by taking the infimum over all homomorphisms in $d_{\mathrm{HD}}(G, G')$. Hence, the triangle inequality described in Item 3 holds. □

B.2 Proof of Proposition 2

Proposition B.2. *Given two attribute functions $f, g : V \to \mathcal{X}$ on the same graph G, we have*

$$d_{HD}((V, E, f), (V, E, g)) \leq \|f - g\|_\infty \tag{23}$$

where $\|\cdot\|_\infty$ denotes the supremum norm with respect to the norm of $\mathcal{X}$.

.

Proof. Since we are dealing with the same graph, the identity is in $\mathrm{Hom}(G, G)$. If we evaluate Definition 7 with the identity, we have that $\max_{v \in V} \|f(v) - g(v)\|$ appears in the right-hand side minimization. Since the infimum is taking for calculating Definition 7 the claim follows. □

B.3 Proof of Lemma 1

Lemma B.1. *Let $G = (V, E)$ and $G' = (V', E')$ be two isomorphic graphs with attribute functions $f : V \to \mathcal{X}$ and $f' : V' \to \mathcal{X}$ respectively. The* homomorphism distortion $d_{HD}(G, G') = 0$ *if f is permutation invariant and $f = f'$.*

Proof. Since $G \simeq G'$, there is one $\varphi \in \mathrm{Hom}(G, G')$ that is an isomorphism or permutation. Then, every term in the calculation will take the form of $f(v) - f(\varphi(v)$, taking into account that f is invariant to permutations and φ is a permutation, we have that $f(v) = f(\varphi(v))$. Then, the difference is 0 for all nodes under the isomorphism φ. This follows for the opposite direction, taking $\varphi^{-1} \in \mathrm{Hom}(G', G)$, which exists since $G \simeq G'$ and since we are taking the infimum over all homomorphisms, then the claim follows. $\square$

B.4 Proof of Lemma 2

Lemma B.2. *Let $G = (V, E)$ and $G' = (V, E)$ be two non-ismorphic graphs with attribute functions $f : V \times \{E\} \to \mathcal{X}$ and $f' : V' \times \{E'\} \to \mathcal{X}$ respectively. The* homomorphism distortion *between G and G′ is never* 0 *, i.e. $d_{HD} \neq 0$, if f is permutation invariant, injective, and $f = f'$.*

Proof. As a short-hand, we still refer to the mapping f and f' on an arbitrary vertex as $f(v)$ instead of $f(v, E)$, but it takes both a vertex and the entire edge set as attributes. Since G and G' are not isomorphic, for any pair of homomorphisms $\varphi \in \mathrm{Hom}(G, G')$ and $\varphi' \in \mathrm{Hom}(G', G)$ we aim to show that either $\Delta(\varphi) > 0$ or $\Delta(\varphi') > 0$. Depending on the relationship of the order of G and G', we have three cases:

1. Case 1: If $|G'| < |G|$ then we know that φ is not injective since it cannot map all of V to V'. Then, we know that there exist at least $v_1, v_2 \in V$ such that $\varphi(v_1) = \varphi(v_2) = v'$ for an arbitrary $v' \in V'$. Since f is injective then $f(v_1) \neq f(v_2)$ and thus either $\|f(v_1) - f'(\varphi(v_1))\| > 0$ or $\|f(v_2) - f'(\varphi(v_2))\| > 0$. Because we take the maximum, it follows that $\Delta(\varphi) > 0$.
2. Case 2: If $|G| < |G'|$ is similar to Case 1, where φ' is the non-injective homomorphism and f' is the injective function over it. It follows that $\Delta(\varphi') > 0$.
3. Case 3: If $|G| = |G'|$ we need to maneuver slightly more. Take any arbitrary $\varphi \in \mathrm{Hom}(G, G')$. Since φ is not an isomorphism, either: (a) φ is not a bijection or (b) φ^{-1} exists since there is a bijection φ between G, G' but since $G \not\simeq G'$ then φ' is not a homomorphism. If φ is not a bijection, and both the domain and co-domain are finite, then trivially φ is injective if and only if it's surjective. If it is neither, the claim is resolved through Case 1.

   On the other hand, if φ is a bijection but φ^{-1} is not a homomorphism, there exists two vertices $v_1', v_2' \in V'$ and an edge between them $\{v_1', v_2'\} \in E'$ such that $\{\varphi^{-1}(v_1), \varphi^{-1}(v_2')\} \notin E$. In other words, there is an edge $\{v_1', v_2'\} \in E'$ that does not have an equivalence back in E. The important detail is that φ^{-1} does not preserve the neighborhood, and the neighborhood structure of vertices is not preserved under φ. Since f (and f') are functions of a vertex and the set of edges, injective and $E \neq E'$, then two different edge sets do not map to the same value in the co-domain. Thus, for any vertex $v \in V$ it follows that $\|f(v) - f(\varphi(v))\| > 0$. $\square$

B.5 Proof of Proposition 3

Proposition B.3. *Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. The set of neighborhoods of all nodes in a graph G is denoted $\mathcal{N}_G^* = \{\mathcal{N}_G(v) \mid v \in V\}$ where $\mathcal{N}_G(v) = \{u \mid \{u, v\} \in E(G)\}$. The graphs are equipped with attribute functions $f : V \times \mathcal{N}_G \to \mathcal{X}$ and $f' : V' \times \mathcal{N}_{G'} \to \mathcal{X}$. If the following conditions on f, f' are fulfilled*

1. *$f = f'$*
2. *f is permutation invariant*
3. *f is injective*

then the homomorphism distortion between G, G' and their attributes given by $d_{HD}(G, G')$ is complete.

Proof. Recall that the set of neighbors of a node v_1' is $\mathcal{N}(v_1') = \{w \mid (w, v_1') \in E'\}$. Now, if f and f' are functions of the neighborhood and the node and are injective then since φ^{-1} does not preserve the neighborhood. □

B.6 Proof of Proposition 4

Proposition B.4. *The* homomorphism distortion *is permutation invariant. Formally, let ρ be a permutation $\rho(\cdot)$. Then,*

$$d_{\mathrm{HD}}(\rho(G), \rho(G')) = d_{\mathrm{HD}}(G, G') \tag{24}$$

Proof. First, by definition, we have that the application of a permutation on a graph results in an isomorphic graph. Then, let $\rho(G) = H$ and $\rho(G') = H'$. If $G \simeq G'$ then trivially $d_{\mathrm{HD}}(G, G') = d_{\mathrm{HD}}(H, H') = 0$. Now, let's look at the case where $G \not\simeq G'$. Let φ be a homomorphism between G and G'. Note that the inverse ρ^{-1} maps H back into G also H' back into G'.Now, define $\psi = \rho \circ \varphi \circ \rho^{-1}$ which is a homomorphism between H, H' by definition. Note that for any node $v \in V(G)$ there is a node $h \in H$ such that $f(h) = f(\rho(v)) = f(v)$ since f is permutation invariant. Also note that, $f'(\psi(h)) = f'(\rho(\varphi(v)) = f'(\varphi(v))$ which results in the following

$$\begin{aligned}
\Delta_{H,H'}(\psi) &= \max_{h \in V(H)} \|f(h) - f'(\psi(h))\| \\
&= \max_{v \in V(G)} \|f(\rho(v)) - f(\rho \circ \varphi(v)\| \quad \text{(by definition of } \rho) \\
&= \max_{v \in V(G)} \|f(v) - f(\varphi(v))\| \quad \text{(by injectivity of } f) \\
&= \Delta_{G,G'}(\varphi)
\end{aligned}$$

And we can make the same procedure the other way around to obtain

$$\max(\Delta_{H,H'}(\psi), \Delta_{H',H}(\psi')) = \max(\Delta_{G,G'}(\varphi), \Delta_{G',G}(\varphi')) \tag{25}$$

where we know that for

$$\begin{aligned}
d_{\mathrm{HD}}(H, H') &\leq \max(\Delta_{H,H'}(\psi), \Delta_{H',H}(\psi')) = \max(\Delta_{G,G'}(\varphi), \Delta_{G',G}(\varphi')) \\
&\leq d_{\mathrm{HD}}(G, G').
\end{aligned}$$

In the last step, we can equivalently do the inverse process. Now take ψ to be a homomorphism from H to H'. Define $\varphi = \rho^{-1} \circ \psi \circ \rho$ and follow the same process such that

$$\begin{aligned}
\Delta_{G,G'}(\varphi) &= \max_{v \in V(G)} \|f(v) - f'(\varphi(v))\| \\
&= \max_{h \in V(H)} \|f(\rho^{-1}(h)) - f'(\rho^{-1} \circ \psi(h))\| \\
&= \max_{h \in V(H)} \|f(h) - f(\psi(h))\| \\
&= \Delta_{H,H'}(\psi)
\end{aligned}$$

which leads to

$$\begin{aligned}
d_{\mathrm{HD}}(G, G') &\leq \max(\Delta_{G,G''}(\psi), \Delta_{G',G}(\psi')) = \max(\Delta_{H,H'}(\varphi), \Delta_{H',H}(\varphi')) \\
&\leq d_{\mathrm{HD}}(H, H')
\end{aligned}$$

And the claim follows. □

B.7 Proof of Proposition 5

Proposition B.5. *The homomorphism distortion $d^*_{HD}(G, G')$ is a metric, i.e. the following properties hold for arbitrary vertex-attributed graphs G, G', G'' if we retrict the choices of f, f':*

1. $d^*_{HD}(G, G) = 0$
2. $d^*_{HD}(G, G') = d^*_{HD}(G', G)$
3. $d^*_{HD}(G, G') > 0$
4. $d^*_{HD}(G, G') \leq d^*_{HD}(G, G'') + d^*_{HD}(G'', G')$

Proof. Items 1, 2, and 4 are derived in the same way as in Proposition 1. Item 3 can be inferred by using Theorem 1. Namely, we can safely assume $G \not\simeq G'$ since G and G' are in different equivalence classes. By Theorem 1 it follows that for non-isomorphic graphs the $d^*_{HD}(G, G') > 0$. □

B.8 Proof of Lemma 3

Lemma B.3. *The* homomorphism distortion *of two graphs G, G' given by $d_{HD}(G, G')$ can be approximated with a set of surrogate graphs $F \in \mathcal{F}$ where $\mathcal{F}$ is a set class of graphs by*

$$d_{HD}(G, G') = \max_{F \in \mathcal{F}} |d_{HD}(F, G) - d_{HD}(F, G')| \tag{26}$$

Proof. Since d_{HD} satisfies the triangle inequality we have

$$|d_{HD}(F, G) - d_{HD}(F, G')| \leq d_{HD}(G, G') \tag{27}$$

If this holds for all F, we may obtain a tighter bound via

$$d_{HD}(G, G') = \max_{F \in \mathcal{F}} |d_{HD}(F, G) - d_{HD}(F, G')| \tag{28}$$

□

B.9 Proof of Lemma 4

Lemma B.4. *Let G be a graph with attribute function f and let $\mathcal{F}$ be a class of graphs. The homomorphism distortion embedding $d_{\mathrm{HD}}(G,\cdot)_{\mathcal{F}}$ is invariant if $\mathcal{F}=\mathcal{G}_n$.*

Proof. Let G' be a graph with attribute function f'. We need to show that if $G \simeq G'$ then the following holds $d_{\mathrm{HD}}(G,\cdot)_{\mathcal{F}} = d_{\mathrm{HD}}(G',\cdot)_{\mathcal{F}}$. By Lemma 1 we know $d_{\mathrm{HD}}(G,G') = 0$. Now, we have two cases that need to be the same:

1. $\Delta_{G,F}(\psi) = \Delta_{G',F}(\varphi)$ where ψ is the minimum homomorphism from G to F and φ is the minimum from G' to F.
2. $\Delta_{F,G}(\psi') = \Delta_{F,G'}(\varphi')$ where ψ' is the minimum homomorphism from F to G and φ' is the minimum from F to G'.

For the first, we define $\psi = \varphi \circ \rho$ and denote the isomorphism $\rho : G \to G'$, and the attribute function over F as g, then we get that

$$\begin{aligned}
\Delta_{G,F}(\psi) &= \max_{v \in V(G)} \|f(v) - g(\psi(v))\| \\
&= \max_{v \in V(G)} \|f(v) - g(\varphi(\rho(v)))\| \quad \text{(by definition of } \psi) \\
&= \max_{v \in V(G)} \|f'(\rho(v)) - g(\varphi(\rho(v)))\| \quad \text{(using } f(v) = f'(\rho(v)))
\end{aligned}$$

Since ρ is a bijection from G to G', if we let $v' = \rho(v)$, the maximization is equivalent to

$$\Delta_{G,F}(\psi) = \max_{v' \in V(G')} \|f'(v') - g(\varphi(v'))\| = \Delta_{G',F}(\phi)$$

The reverse is also true by defining $\phi = \psi \circ \rho^{-1}$. Now, in the second case, let $\psi' : F \to G$ be a homomorphism. We define a corresponding map $\phi' : F \to G'$ by the composition $\phi' = \rho \circ \psi'$. Notice that

$$\begin{aligned}
\Delta_{F,G'}(\varphi') &= \max_{u \in V(F)} \|g(u) - f'(\varphi'(u))\| \\
&= \max_{u \in V(F)} \|g(u) - f'(\rho(\psi'(u)))\| \quad \text{(by definition of } \varphi') \\
&= \max_{u \in V(F)} \|g(u) - f(\psi'(u))\| \quad \text{(using the identity } f'(\rho(v)) = f(v)) \\
&= \Delta_{F,G}(\psi')
\end{aligned}$$

then the claim follows □

B.10 Proof of Proposition 6

Proposition B.6. *Let G be a graph with attribute function f. Also, let $e_F \in \mathbb{R}^{\mathcal{F}}$ be the F-th standard basis unit-vector of $\mathbb{R}^{\mathcal{F}}$. Define the graph embedding resulting from the homomorphism distortion as*

$$\varsigma_F(G) = d_{\mathrm{HD}}(G,F)e_F.$$

If $\mathcal{D}$ is a distribution with with full support on $\mathcal{G}_n$ and $F \sim \mathcal{D}$, then the random graph embedding $\varsigma_F(\cdot)$ is expectation-complete on $\mathcal{G}_n$.

Proof. Denote the expectation over a discrete distribution $\mathcal{D}$ with full support on $\mathcal{F}$ as

$$g = \mathbb{E}_{F\sim D}[\varsigma_F(G)] = \sum_{F' \in \mathcal{G}_n} P(F = F') \cdot d_{\mathrm{HD}}(G, F')e_{F'}$$

where g is a vector with each entry $(g)_{F'}$ corresponding to one $F' \in \mathcal{F}$. Let G' be a graph with attribute function f', where $G' \not\simeq G$ and $g' = \mathbb{E}_F[\varsigma_F(G)]$. By Theorem 1 and since $G \not\simeq G'$ then $d_{\mathrm{HD}}(G, \cdot) \neq d_{\mathrm{HD}}(G', \cdot)$. Thus, there exists an F' such that $d_{\mathrm{HD}}(G, F') \neq d_{\mathrm{HD}}(G', F')$. By definition of D we have that $P(F = F') > 0$, since it has full support on $\mathcal{G}_n$ and thus $P(F = F')d_{\mathrm{HD}}(G, F') \neq P(F = F')d_{\mathrm{HD}}(G', F')$ which implies $g \neq g'$. It follows that $\mathbb{E}_{F\sim D}[\varsigma_F(\cdot)]$ is complete. □

B.11 Proof of Theorem 4

Theorem B.1. *Let $G = (V, E)$ be a graph with a function f, and $F \sim \mathcal{D}$ be a graph sampled from D with an attribute function f'. Also, let $H_{G,F} \sim Q$ and $H_{F,G} \sim \mathcal{U}$ be two sets of ℓ samples from the distributions $\mathcal{Q}$,$\mathcal{U}$ conditional on the sample F. Then, the random graph embedding resulting from this parametrization of the homomorphism distortion embedding is expectation-complete on $\mathcal{G}_n$*

Proof. We aim to show that the *random graph embedding*

$$\mathbb{E}_F[\mathbb{E}_{H_{G,F},H_{F,G}}[\varsigma_{F,H_{G,F},H_{F,G}}(G)]](\cdot)$$

is complete. First, recall that using Definition 8 and Definition 7 we can rewrite the expectation over a particular F generated by the homomorphism distortion as

$$\sigma_F(X) := \mathbb{E}_{H_{G,F},H_{F,G},}[\max(\inf_{\varphi\in H_{G,F}} \Delta(\varphi), \inf_{\varphi'\in H_{F,G}} \Delta(\varphi')]. \tag{29}$$

This allows us to rewrite the full expectation more compactly by unrolling the outer expectation

$$\begin{aligned}\mathbb{E}_F[\mathbb{E}_{H_{G,F},H_{F,G}}[\varsigma_{F,H_{G,F},H_{F,G}}(G)]] &= \sum_{F'\in\mathcal{G}_n} P(F = F')\mathbb{E}_{H_{G,F},H_{F,G}}[\varsigma_{F,H_{G,F},H_{F,G}}(G)]e_{F'} \\ &= \sum_{F'\in\mathcal{G}_n} P(F = F')\sigma_{F'}(G)e_{F'}\end{aligned}$$

where $e_{F'} \in \mathbb{R}^{\mathcal{G}_n}$ as defined previously. Now, let $G' = (V', E')$ be another graph with attribute function f' such that $G \not\simeq G'$. Also, we denote the expectation over the parametrized *homomorphism distortion* over a particular F for G and G' as

$$g = \sigma_F(G), \quad g' = \sigma_F(G'). \tag{30}$$

To show that the embedding of G, G' is different, we need to show that $|g - g'| > 0$. Recall that we already know that $d_{\mathrm{HD}}(G, G') > 0$ by Theorem 1. Set $p_{G'} := P(F = G')$. Since $\mathcal{D}$ has full support on $\mathcal{G}_n$ then $p_{G'} > 0$. For some $\delta > 0$ there exists a $L \in \mathbb{N}$

such that for all $\ell \geq L$ $\mathrm{id}_{G'} \in H_{G',G'} = \{\varphi_1, \ldots, \varphi_\ell\}$ with probability $1-\delta$. Since $\mathcal{Q}, \mathcal{U}$ have full support we know that $H_{G',G'} \subseteq \mathrm{Hom}(G', G')$. Note that for a sufficiently large L, the identity is always the minimum where

$$\inf_{\varphi \in H_{G',G'}} \Delta(\varphi) = \Delta(\mathrm{id}_{G'}) = 0$$

which results in $\sigma_{G'}(G') = 0$. Additionally, notice that for the case where we take the expectation over a particular infimum, with high probability and by monotonicity of expectation

$$\mathbb{E}_{H_{G,F}}[\inf_{\varphi \in H(G,G')} \Delta(\varphi)] \geq \inf_{\varphi \in \mathrm{Hom}(G,G')} \Delta(\varphi)$$

which implies that

$$\sigma_{G'}(G) \geq d_{\mathrm{HD}}(G, G') \tag{31}$$

allowing us to decompose the difference $|g - g'|$ for a fixed $F' = G'$ into

$$\begin{aligned}
|g - g'| &= \left|\sigma_{F'}(G) - \sigma_{F'}(G')\right| \\
&\geq \left|\sigma_{G'}(G) - \sigma_{G'}(G')\right| \quad \text{(by assumption)} \\
&\geq \left|\sigma_{G'}(G) - 0\right| \quad \text{(since distance to itself is 0)} \\
&\geq \sigma_{G'}(G) \\
&\geq \cdot d_{\mathrm{HD}}(G, G') \quad \text{(by Equation (31))}
\end{aligned}$$

since $d_{\mathrm{HD}}(G, G') > 0$,then the claim follows. □

C Extended experiments

Hardware setup. We perform all experiments on the `Snellius` cluster on a node with Intel Xeon Platinum 8360Y with 36 @ 2.4 GHz, 512 GiB DRAM, and GPU acceleration using an NVIDIA A100 with 40GiB of HBM2 memory.

Hyperparameters. Table 3 shows the hyperparameters of the *homomorphism distortion embedding* under which we optimize on and Table 4 shows the hyperparameters we keep fixed.

D Extended results

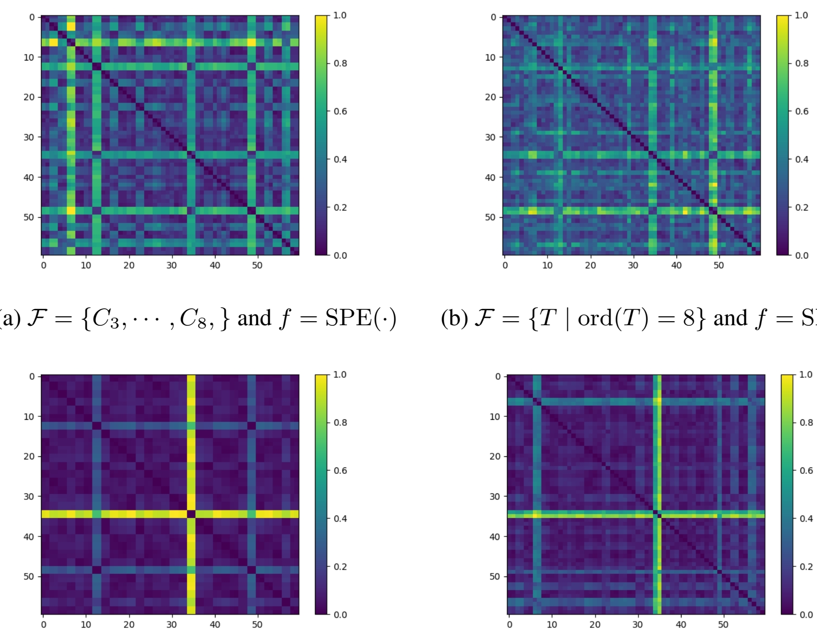
To dissipate the curiosity of what the difference is between the choices of f for other graph classes we briefly go over one set of results of the $d_{\mathrm{HD}}(\cdot)$ on them. Figures 2 to 6 show the distances for all the subsets of `BREC`. There are two main behaviors to watch out for. First, SPEs perform better than RWPEs, a surprising result since RWPEs are usually used as a *cheap but good* characterization function. Second, using trees rs

Table 3: Hyperparameters under which optimization is performed.

Hyperparameter	Values
$\mathcal{F}_1$	$\{C_8, C_9, C_{10}, C_{11}\}$
$\mathcal{F}_2$	$\{T_8, T_9\}$
σ	$\{10, 20, 30, 40\}$
γ	$\{10, 20, 42\}$
Batch size	$\{64, 128\}$

Table 4: Fixed hyperparameters.

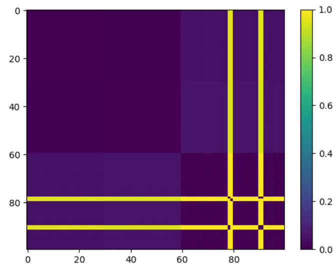
Hyperparameter	Value
LR	0.001
LR patience	20
Min. LR	0.00001
LR factor	0.5
Warmup epochs	10
Weight decay	0
Max epoch	1000
Layers	172
Hidden dimension	16



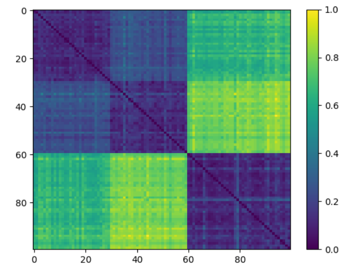
(a) $\mathcal{F} = \{C_3, \cdots, C_8,\}$ and $f = \mathrm{SPE}(\cdot)$ (b) $\mathcal{F} = \{T \mid \mathrm{ord}(T) = 8\}$ and $f = \mathrm{SPE}(\cdot)$

(c) $\mathcal{F} = \{C_3, \cdots, C_8,\}$ and $f = \mathrm{RWPE}(\cdot)$ (d) $\mathcal{F} = \{T \mid \mathrm{ord}(T) = 8\}$ and $f = \mathrm{RWPE}(\cdot)$

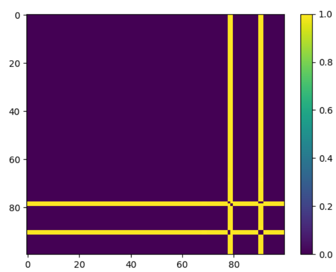
Fig. 2: Class B containing basic (up to non 1-WL distinguishable) graphs. The distance matrices $d_{\mathrm{DH}}(\cdot)$ with $\sigma = 10$ and $\gamma = 10$ are shown. The top row shows the evaluation on SPEs Figure 2a and Figure 2b and the bottom on RWPEs Figure 2c and Figure 2d.



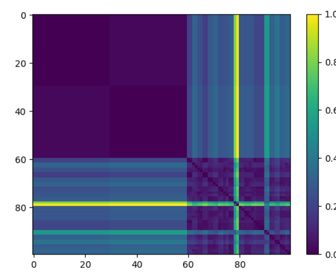
(a) $\mathcal{F} = \{C_3, \cdots, C_8,\}$ and $f = \mathrm{SPE}(\cdot)$



(b) $\mathcal{F} = \{T \mid \mathrm{ord}(T) = 8\}$ and $f = \mathrm{SPE}(\cdot)$



(c) $\mathcal{F} = \{C_3, \cdots, C_8,\}$ and $f = \mathrm{RWPE}(\cdot)$

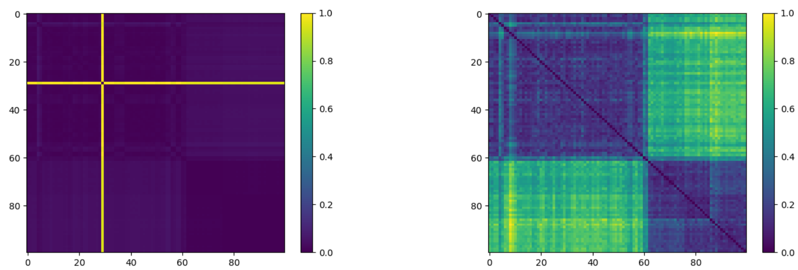


(d) $\mathcal{F} = \{T \mid \mathrm{ord}(T) = 8\}$ and $f = \mathrm{RWPE}(\cdot)$

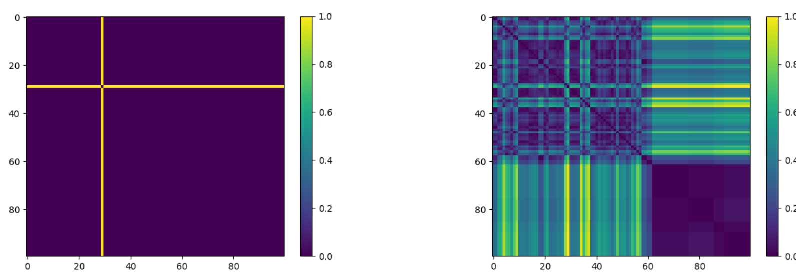
Fig. 3: Class C containing CFI graphs. The distance matrices $d_{\mathrm{DH}}(\cdot)$ with parameters $\sigma = 10$ and $\gamma = 10$ are shown. The top row shows the evaluation on SPEs Figure 3a and Figure 1b and the bottom on RWPEs Figure 3c and Figure 3d.

Table 5: The MAE for graph regression on `ZINC-12k` (without edge features) on 500K parameters. We use the best performing class $\mathcal{F}^* = \{T \mid \mathrm{ord}(T) = 9\}$ with the previous best approaches. Our method *outperforms* previous approaches when adding Sub in GIN and is very close (within std) in GCN. Using Hom outperforms the homomorphism sampling approach $\mathrm{Hom} + F$.

	GAT		GCN		GIN	
Base	0.380	± 0.009	0.282	± 0.007	0.246	± 0.019
Spasm$^{\circledast}$ [30]	**0.155**	**± 0.008**	**0.166**	**± 0.005**	0.143	± 0.004
Hom + F [62]	-		0.207	± 0.008	0.174	± 0.005
$d_{HD}(G,\cdot)_{\mathcal{F}^*}$	0.217	± 0.004	0.217	± 0.004	0.184	± 0.005
$d_{HD}(G,\cdot)_{\mathcal{F}^*}$+`Hom`	**0.171**	**± 0.002**	0.179	± 0.008	0.152	± 0.004
$d_{HD}(G,\cdot)_{\mathcal{F}^*}$+`Sub`	**0.170**	**± 0.003**	0.167	± 0.005	**0.130**	**± 0.005**

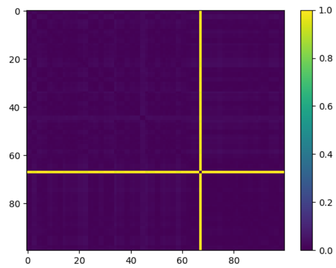


(a) $\mathcal{F} = \{C_3, \cdots, C_8,\}$ and $f = \mathrm{SPE}(\cdot)$ (b) $\mathcal{F} = \{T \mid \mathrm{ord}(T) = 8\}$ and $f = \mathrm{SPE}(\cdot)$

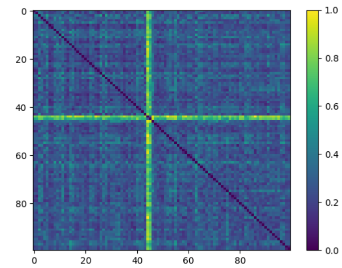


(c) $\mathcal{F} = \{C_3, \cdots, C_8,\}$ and $f = \mathrm{RWPE}(\cdot)$ (d) $\mathcal{F} = \{T \mid \mathrm{ord}(T) = 8\}$ and $f = \mathrm{RWPE}(\cdot)$

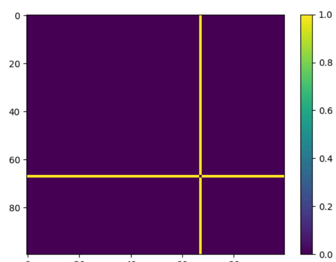
Fig. 4: Class E containing extension graphs. The distance matrices $d_{\mathrm{DH}}(\cdot)$ with parameters $\sigma = 10$ and $\gamma = 10$ are shown. The top row shows the evaluation on SPEs Figure 4a and Figure 4b and the bottom on RWPEs Figure 4c and Figure 4d.



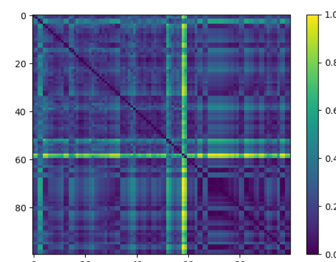
(a) $\mathcal{F} = \{C_3, \cdots, C_8,\}$ and $f = \mathrm{SPE}(\cdot)$



(b) $\mathcal{F} = \{T \mid \mathrm{ord}(T) = 8\}$ and $f = \mathrm{SPE}(\cdot)$

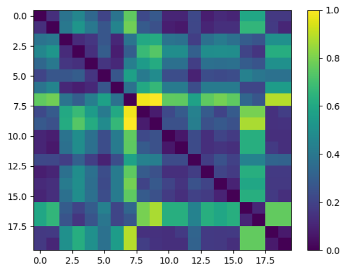


(c) $\mathcal{F} = \{C_3, \cdots, C_8,\}$ and $f = \mathrm{RWPE}(\cdot)$

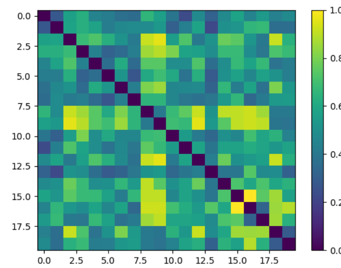


(d) $\mathcal{F} = \{T \mid \mathrm{ord}(T) = 8\}$ and $f = \mathrm{RWPE}(\cdot)$

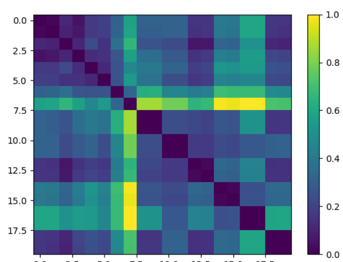
Fig. 5: Class R containing regular graphs. The distance matrices $d_{\mathrm{DH}}(\cdot)$ with parameters $\sigma = 10$ and $\gamma = 10$ are shown. The top row shows the evaluation on SPEs Figure 5a and Figure 5b and the bottom on RWPEs Figure 5c and Figure 5d.



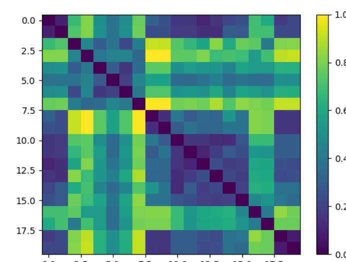
(a) $\mathcal{F} = \{C_3, \cdots, C_8,\}$ and $f = \mathrm{SPE}(\cdot)$



(b) $\mathcal{F} = \{T \mid \mathrm{ord}(T) = 8\}$ and $f = \mathrm{SPE}(\cdot)$



(c) $\mathcal{F} = \{C_3, \cdots, C_8,\}$ and $f = \mathrm{RWPE}(\cdot)$



(d) $\mathcal{F} = \{T \mid \mathrm{ord}(T) = 8\}$ and $f = \mathrm{RWPE}(\cdot)$

Fig. 6: Class D containing distance-regular graphs. The distance matrices $d_{\mathrm{DH}}(\cdot)$ with parameters $\sigma = 10$ and $\gamma = 10$ are shown. The top row shows the evaluation on SPEs Figure 6a and Figure 6b and the bottom on RWPEs Figure 6c and Figure 6d.