Expressivity and Generalization: Fragment-Biases for Molecular GNNs

Tom Wollschläger [*1 2] **Niklas Kemper** [*1] **Leon Hetzel** [1 3] **Johanna Sommer** [1 2] **Stephan Günnemann** [1 2]

Abstract

Although recent advances in higher-order Graph Neural Networks (GNNs) improve the theoretical expressiveness and molecular property predictive performance, they often fall short of the empirical performance of models that explicitly use fragment information as inductive bias. However, for these approaches, there exists no theoretic expressivity study. In this work, we propose the *Fragment-WL* test, an extension to the well-known Weisfeiler & Leman (WL) test, which enables the theoretic analysis of these fragment-biased GNNs. Building on the insights gained from the Fragment-WL test, we develop a new GNN architecture and a fragmentation with infinite vocabulary that significantly boosts expressiveness. We show the effectiveness of our model on synthetic and real-world data where we outperform all GNNs on Peptides and have 12% lower error than all GNNs on ZINC and 34% lower error than other fragment-biased models. Furthermore, we show that our model exhibits superior generalization capabilities compared to the latest transformer-based architectures, positioning it as a robust solution for a range of molecular modeling tasks.

1. Introduction

A common issue with Graph Neural Networks (GNNs) is their lack of expressiveness, including their inability to recognize substructures, which could limit their empirical performance (Morris et al., 2021). In chemistry and machine learning for chemistry, frequently occurring substructures, or fragments, are commonly used to improve expressivity, as they can be powerful predictors of the functional properties of molecules (Merlot et al., 2003). The beneficial effect of fragments becomes even more apparent in larger systems, such as proteins, where individual components often resemble whole residues (Singh & Saha, 2003).

To give GNNs the predictive power of substructures, recent work has introduced more powerful methods that relate expressive power to the ability to distinguish isomorphic graphs. For these higher-order GNNs, the Weisfeiler & Leman (WL) test serves as a measure of expressivity. Empirical evaluations analyze the ability of these models to count substructures and predict molecular properties (Zhang et al., 2023; Morris et al., 2021). However, many of these works focus primarily on theoretical expressiveness analysis, often neglecting practical implications.

Recently, Campi et al. (2023) showed that these models suffer from poor generalization to data that do not perfectly fit the training distribution. This often results in sub-par performance on real-world data (Maron et al., 2019; Gasteiger et al., 2020). At the same time, models that explicitly incorporate fragment information as inductive biases perform better overall (Fey et al., 2020), but are often limited to a single substructure and lack a theoretical analysis of their expressiveness, leading to a mismatch between theory and practical performance (Zhu et al., 2022; Fey et al., 2020; Zang et al., 2023).

In this work, we address the gap between theory and practical performance by introducing the Fragment-WL test, an extension of the standard WL test that enables a unified analysis of existing (fragment-biased) models. In addition, we introduce a new powerful model that directly leverages graph fragments within its message-passing framework. Thereby, our model becomes more robust to varying graph structures and can generalize better to out-of-distribution data. Lastly, our model enables a new fragmentation to represent molecular graphs with an infinite vocabulary consisting only of basic building blocks. We show the usefulness of our approach across a range of molecular datasets and tasks. We analyze both short- and long-range interactions, where we achieve state-of-the-art performance compared to graph-based models and even outperform transformer-based architectures in some scenarios.[1]

[*]Equal contribution [1]School of Computation, Information & Technology, Technical University of Munich. [2]Munich Data Science Institute, Germany. [3]Helmholtz Center for Computational Health, Munich, Germany. Correspondence to: Tom Wollschläger <t.wollschlaeger@tum.de>, Niklas Kemper <niklas.kemper@tum.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1]Find our code at cs.cit.tum.de/daml/fragment-biased-gnns/

Our core contributions can be summarized as follows:

- We provide a more fine-grained hierarchy on the expressiveness for a multitude of models that incorporate substructures as inductive bias, such as including them as node features, learning an individual representation, or performing operations on higher-level structures.
- We propose a new architecture that performs message passing along substructures and improves expressivity and generalization while retaining linear complexity.
- With our new architecture, we can propose a novel fragmentation for molecules that handles an infinite number of substructures based on simple yet flexible building blocks but still generalizes well.
- We study predictive power, long-range performance, and generalization through extensive experiments.

2. Background

Notation. A graph $G := (\mathcal{V}, \mathcal{E}, \boldsymbol{X})$ consists of a set of vertices $\mathcal{V}$, a set of (undirected) edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ and d node features $\boldsymbol{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ for every node $v \in \mathcal{V}$. The set of nodes that are adjacent to v is denoted by $\mathcal{N}(v)$. Two graphs $G^1 = (\mathcal{V}^1, \mathcal{E}^1, \boldsymbol{X}^1)$ and $G^2 = (\mathcal{V}^2, \mathcal{E}^2, \boldsymbol{X}^2)$ are *isomorphic* if there exists a bijection $b : \mathcal{V}^1 \to \mathcal{V}^2$ that preserves edges and node features, that is, $\{v, w\} \in \mathcal{E}^1 \Leftrightarrow \{b(v), b(w)\} \in \mathcal{E}^2$ and $\boldsymbol{X}^1_v = \boldsymbol{X}^2_w$. For a subset of nodes $\mathcal{U} \subseteq \mathcal{V}$, we denote the induced subgraph with respect to these nodes by $G[\mathcal{U}]$.

Expressiveness. We can classify the expressiveness of functions over graphs by their capability to distinguish non-isomorphic graphs. We say that a function f is (in parts) *more powerful* than a function g if there exist two non-isomorphic graphs G^1, G^2 such that $f(G^1) \neq f(G^2)$ whereas $g(G^1) = g(G^2)$. The function f is *strictly more powerful* than g (we write $f > g$) if f is more powerful than g and g is not (in parts) more powerful than f.

Weisfeiler & Leman. The Weisfeiler & Leman graph isomorphism test is an iterative graph coloring algorithm that bounds the expressive power of MPNNs (Kiefer & Neuen, 2022). In each iteration, it produces a color for each node based on its neighboring nodes' colors. Starting with a vertex color based only on features $c_v^0 = \mathrm{HASH}(\boldsymbol{X}_v)$, we calculate the update for the color c of node v in iteration t:

$$c_v^{(t)} = \mathrm{HASH}\left(c_v^{(t-1)}, \{\!\{c_w^{(t-1)} \mid w \in \mathcal{N}(v)\}\!\}\right). \quad (1)$$

The algorithm terminates once the set of unique colors does not increase. Two non-isomorphic graphs can be distinguished if the multiset of colors differs at the end. As this test cannot distinguish all non-isomorphic graphs, it can be extended to strictly more powerful versions, k-WL, incorporating k-tuples of nodes to determine the color. For more background information, we refer to Morris et al. (2021). Importantly, 2-WL is equivalent to the previously described WL test (Huang & Villar, 2021).

Fragmentations. A vocabulary $\mathcal{Y}$ is a set of graphs (potentially including node features) representing important substructures, e.g., cycles. A fragment of a graph G is an induced subgraph $G[f]$ isomorphic to a graph from the vocabulary. We will identify a fragment simply by the subset of nodes $f \subseteq \mathcal{V}$. All fragments f that are isomorphic to the same graph of the vocabulary have the same type(f). A fragmentation scheme $\mathcal{F}$ is a permutation invariant function that maps a graph G to a set of fragments $\mathcal{F}(G) =: F$, which is called a fragmentation. Note that there might exist subgraphs isomorphic to a graph in $\mathcal{Y}$ that are *not* in $\mathcal{F}(G)$. For example, even if $\mathcal{Y}$ contains 5-cycles, not all 5-cycles in G need to be in $\mathcal{F}(G)$. If, for all graphs G, $\mathcal{F}(G)$ includes *every* subgraph isomorphic to a graph $V \in \mathcal{Y}$, we say that the fragmentation scheme $\mathcal{F}$ recovers V.

3. Related work

Expressiveness of GNNs. Message-Passing Neural Networks (MPNN)[2] are limited in their expressiveness. Their ability to distinguish between non-isomorphic graphs is confined to the 2-WL algorithm, restricting their discriminative power (Xu et al., 2019). Moreover, when it comes to recognizing substructures, MPNNs are unable to accurately count almost all types of substructures (Chen et al., 2020). This limitation stems from their reliance on purely local messages, which—despite facilitating excellent linear space and time complexity—renders them blind to higher-level structural information within graphs.

Higher-order GNNs. In response to the inability to effectively learn substructures, the introduction of more powerful GNN architectures aims to overcome this limitation and enable comprehensive substructure learning. Morris et al. (2021) draw inspiration from the multidimensional k-WL algorithm and diverge from learning node-specific representations by considering each k-tuple of nodes instead. Although this improves expressiveness, its complexity increases exponentially. Subgraph GNNs comprise an alternative to improve substructure identification, decomposing a graph into smaller subgraphs for GNN application. The resulting subgraph representations are pooled before a final graph level representation is derived (Huang et al., 2022; Frasca et al., 2022). With some strategies for extracting subgraphs, subgraph GNNs can identify basic substructures such as 4-cycles (Huang et al., 2022). Puny et al. (2023) extend the WL test for higher-order GNNs to the graph

[2]We use MPNNs and GNNs interchangeably.

polynomial counting problem as a new expressivity measure, highlighting the importance of more fine-grained tests for GNNs. However, the limitations of higher-order GNNs lie in their inability to effectively learn more intricate substructures, accompanied by an increase in time complexity. Recent findings also suggest susceptibility to adversarial attacks and out-of-distribution data, hinting at challenges in robustly learning substructures (Campi et al., 2023).

Fragment-Biased GNNs. Another line of work provides fragment information to GNNs as an explicit inductive bias. These fragment-biased models vary not only in their vocabulary but also in the way fragmentation information is integrated into the model. *Node features*: Bouritsas et al. (2023) introduce GSN-v, which uses the number of cycles or cliques as an additional node feature. *Learned fragment representation*: Instead of treating fragmentation information as a fixed feature, other models learn representations for each fragment by aggregating information from the corresponding nodes. Zhu et al. (2022) use a vocabulary of only cycles whereas Zang et al. (2023) present HiMol, which fragments a molecular graph, based on chemical properties. *Higher-level graph*: A natural extension of the learned fragment representation is a higher-level graph of fragments where neighboring fragments influence each other. Thiede et al. (2021) use equivariant computations along the paths of length 3 to 6 and cycles of sizes 5 and 6. Fey et al. (2020) build a higher-level junction tree using rings and edges. Yet, none of the existing works compare—theoretically or experimentally—how to encode and use substructure information in the model. Additionally, most works only focus on a single substructure that does not allow to fragment the complete graph.

Topological GNNs use higher-level topological structures such as simplicial complexes (Bodnar et al., 2021) or CW-Networks (Bodnar et al., 2022; Giusti et al., 2023) in their message-passing schemes. While coming from a different theoretical direction than substructure-biased GNNs, in practice, they use cliques or cycles as learned fragment representations or in a higher-level graph.

Graph Transformer. Recently, models such as Graph Transformers (Ying et al., 2021; Ma et al., 2023; Geisler et al., 2023) and ViT/MLP-Mixers (He et al., 2023) for graphs adapted successful models from other domains to graph data. Their ability to recognize substructures depends on the positional encoding used. Almost all recent models use random walk encodings, which can help to discover simple substructures like cycles.

Fragmentation Schemes. Fragmentation methods in the chemical domain aim to divide a molecular graph into subgraphs with distinct structures or properties. There are various strategies to achieve this, such as separating probable reactants (Degen et al., 2008a), categorizing molecules into distinct structural classes (Bemis & Murcko, 1996), or breaking apart acyclic bonds (Maziarz et al., 2022; Jin et al., 2019). Unlike these methods, data-driven approaches like those outlined in Kong et al. (2022) and Geng et al. (2023) focus on deriving subgraphs directly from a dataset without relying on predefined rules for decomposition.

4. Weisfeiler & Leman Go Fragments

Existing fragment-biased MPNNs vary in their underlying fragmentation scheme and how the fragment information is incorporated into the model. This variability makes a direct comparison of the expressiveness of these models difficult. To address this challenge, we propose a new, more fine-grained version of the WL test, called *Fragment-WL* test, that incorporates detailed structural elements. We derive a hierarchy of tests that capture how fragmentation information is incorporated into existing substructure-biased models, while leaving out all variability that does not influence the expressivity.

Our Fragment-WL test also subsumes existing WL variants designed for simplicial complexes and CW-cells (Bodnar et al., 2021; 2022), providing a more unified framework for assessing the expressiveness of both substructure-biased and topological GNNs. Furthermore, our proposed Fragment-WL test highlights the significance of how fragment information is incorporated into the model, emphasizing that the integration methodology plays a crucial role for determining the model's expressive power.

Fragment-WL entails multiple variants with increasing expressiveness in distinguishing isomorphic graphs. In the following, we first provide a general framework and then define the individual Fragment-WL versions that perform the original WL test on different augmented graphs. We start with a definition of WL tests on augmented graphs:

Definition 4.1. A g-WL test is a function that performs the WL test on the augmented graph $g(G)$, i.e.

$$g\text{-WL}(G) := \text{WL}(g(G))$$

where g is a function mapping from graphs to graphs, i.e., $g : (\mathcal{V}, \mathcal{E}, \boldsymbol{X}) \mapsto (\mathcal{V}', \mathcal{E}', \boldsymbol{X}')$.

There are three ways in which a fragmentation F is used in existing fragment-biased GNNs: as an additional node feature, as learned fragment representation, and as a higher-level graph. We instantiate g with the corresponding functions to augment the graph with the respective features. First, we use additional node features. We extend the individual node features with the information of the fragments that the node is contained in. We concatenate this information to the already existing features. Formally, we define this augmentation function in the following way.

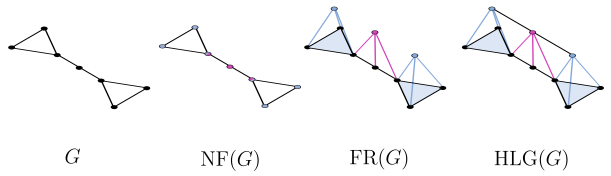

Figure 1. Example graph G with corresponding augmented variants. NF(G) includes node features, FR(G) also includes a representation for each fragment and HLG(G) also has connections between neighboring fragment represenations.

Definition 4.2. We define the node feature function as $\text{NF}(\mathcal{V}, \mathcal{E}, \boldsymbol{X}) = (\mathcal{V}, \mathcal{E}, \boldsymbol{X}^{\text{NF}})$ with:

$$\boldsymbol{X}_v^{\text{NF}} = X_v \parallel \lambda\big(\{\!\{\text{type}(f) \mid v \in f, f \in F\}\!\}\big),$$

where λ represents any injective function and $\parallel$ indicates the concatenation operation. We instantiate g with NF to create the NF-WL test.

Another prominent way is using representations for each fragment and messages that are flowing from the lower level nodes to their entailing fragment and backwards. This means that we introduce a new vertex for each fragment and connect it to all its corresponding vertices in the original graph. We depict this graph FR(G) in Figure 1. We define this augmentation function in the following.

Definition 4.3. We define the fragment representation function as $\text{FR}(\mathcal{V}, \mathcal{E}, \boldsymbol{X}) = (\mathcal{V}^{\text{FR}}, \mathcal{E}^{\text{FR}}, \boldsymbol{X}^{\text{FR}})$ with:

$$\begin{aligned}
&\mathcal{V}^{\text{FR}} := \mathcal{V} \cup F,\\
&\mathcal{E}^{\text{FR}} := \mathcal{E} \cup \big\{\{f, v\} \mid \forall f \in F, \forall v \in f\big\},\\
&\boldsymbol{X}_i^{\text{FR}} := \begin{cases} \boldsymbol{X}_i & i \in \mathcal{V} \\ \text{type}(i) & i \in F \end{cases}
\end{aligned}$$

Lastly, we allow messages to be exchanged between neighboring fragments, thus creating a higher-level graph on which information can flow. To this end, we add edges between fragments that have neighboring nodes in G and thus construct a graph of the higher-level fragments, see HLG(G) in Figure 1.

Definition 4.4. The higher-level graph augmentation functions is $\text{HLG}(\mathcal{V}, \mathcal{E}, \boldsymbol{X}) = (\mathcal{V}^{\text{HLG}}, \mathcal{E}^{\text{HLG}}, \boldsymbol{X}^{\text{HLG}})$ with:

$$\begin{aligned}
&\mathcal{V}^{\text{HLG}} := \mathcal{V}^{\text{FR}}, \quad \boldsymbol{X}^{\text{HLG}} := \boldsymbol{X}^{\text{FR}},\\
&\mathcal{E}^{\text{HLG}} := \mathcal{E}^{\text{FR}} \cup \big\{\{f, k\} \mid f, k \in F, f \cap k \neq \emptyset\big\}
\end{aligned}$$

Equipped with the formal definitions, we will now compare the power of performing the WL test on these transformed graphs to the original graph[3]. The power of the Fragment-WL test depends on the fragmentation scheme $\mathcal{F}$. With a sufficiently advanced fragmentation scheme, even NF-WL can become arbitrarily powerful.

Theorem 4.5. *There exist fragmentation schemes such that NF-WL, FR-WL and HLG-WL are all strictly more powerful than k-WL for any k.*

However, in practice, mostly fragmentation schemes with a vocabulary of rings, paths, and cliques are used for fragment-biased GNNs (Fey et al., 2020; Bouritsas et al., 2023; Zhu et al., 2022). So, from now on, we will restrict ourselves to such fragmentation in our theoretical analysis. Next, we show that it matters how to incorporate fragment information and that the WL variants become more powerful through higher-level abstraction.

Integrating fragment information from any non-trivial substructure as an additional node feature already increases expressiveness beyond 2-WL.

Theorem 4.6. *NF-WL is strictly more powerful than 2-WL for fragmentation schemes $\mathcal{F}$ that recover any substructure with more than two nodes.*

This shows that the classical 2-WL test cannot reveal differences in the expressivity of fragment-biased GNNs since using any substructure as a node feature already increases expressivity beyond 2-WL. Our Fragment-WL test, however, provides a more fine-grained alternative that reveals that higher-level abstraction through a learned fragment representation strictly increases the expressivity compared to node features:

Theorem 4.7. *FR-WL is strictly more powerful than NF-WL for fragmentation schemes $\mathcal{F}$ recovering 3-cycles.*

Building a higher-level graph of fragments further increases the expressivity. Figure 2 shows an example of two graphs that are indistinguishable by 2-WL, NF-WL, and FR-WL but distinguishable by HLG-WL. Formally, we express this in the following theorem.

Theorem 4.8. *HLG-WL is strictly more powerful than FR-WL for fragmentation schemes $\mathcal{F}$ recovering 3-cycles.*

Hence, we have shown that the expressivity increases strictly monotonically from 2-WL to HLG-WL for fragmentation schemes recovering 3-cycles:

$$\text{2-WL} < \text{NF-WL} < \text{FR-WL} < \text{HLG-WL} \tag{2}$$

Regarding the higher-dimensional k-WL hierarchy, our Fragment-WL hierarchy cannot be bounded by 3-WL if the fragmentation can recover 5 cycles.

Theorem 4.9. *HLG-WL is in parts more powerful than 3-WL for fragmentation schemes $\mathcal{F}$ recovering 5-cycles.*

Our developed Fragment-WL hierarchy models the different ways in which fragment information is used in most

[3]All proofs are detailed in Appendix A

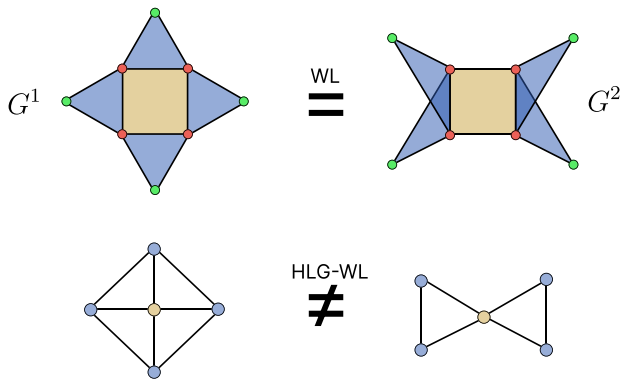

Figure 2. Graphs G^1 and G^2 with their corresponding higher-level graph of fragments. The edges of the fragment representation to the vertices of G^1 and G^2 are omitted. G^1 and G^2 are indistinguishable by WL, NF-WL and FR-WL but distinguishable by HLG-WL as the higher-level graphs exhibit different connections from the 3-ring nodes to the 4-ring node.

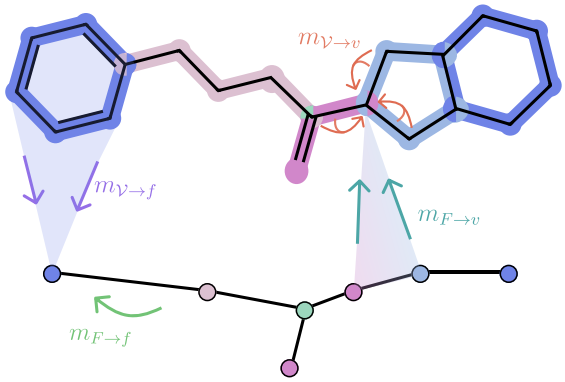

Figure 3. Overview of our model and our fragmentation. The molecular graph is fragmented with our rings-paths fragmentation into three cycles, three paths, and a junction node. The figure shows the messages $m^t_{F\to f}$, $m^t_{\mathcal{V}\to f}$ to one fragment f, and the messages $m^t_{\mathcal{V}\to v}$, $m^t_{F\to v}$ to one vertex v.

fragment-biased and topological GNNs. This new measure of expressiveness allows the comparison and ordering of these existing methods; see Table 7 for an overview.

In summary, our Fragment-WL test provides a new alternative measure of expressivity compared to the original WL test. Our developed hierarchy reveals that it matters how to incorporate fragmentation information, i.e., higher-level abstraction increases expressivity. Additionally, it allows for a comparison of the expressiveness of most existing fragment-biased and topological GNNs.

5. Fragment Graph Neural Network

Based on the insights of the previous section, we propose our new model architecture and a new fragmentation scheme with infinite vocabulary consisting of only basic building blocks. Given the higher expressiveness, our model can differentiate complex substructures given only these basic building blocks, as it is able to learn the dependencies on the higher-level graph. We empirically confirm this in Section 6.

5.1. Model

Building on the theoretical findings that a higher-level graph (see Theorem 4.8) improves expressiveness, we propose FragNet, a general model for any fragmentation F that performs message-passing on the original graph *and* a higher-level graph of fragments that are also connected, i.e., we are using the HLG augmented graph.

The learned representation h^t_v at step or layer t of a node receives a message from neighboring nodes $m^t_{\mathcal{V}\to v}$ and fragments $m^t_{F\to v}$. Similarly, the learned representation h^t_f of a fragment receives a message from neighboring fragments $m^t_{F\to f}$ and nodes $m^t_{\mathcal{V}\to f}$.

$$
\begin{aligned}
m^t_{\mathcal{V}\to v} &= \mathrm{AGG}\big(\{\!\{\mathrm{MLP}(h^{t-1}_u, h^{t-1}_e) \mid e = \{u, v\} \in \mathcal{E}\}\!\}\big)\\
m^t_{F\to f} &= \mathrm{AGG}\big(\{\!\{h^{t-1}_g \mid g \in N_F(f)\}\!\}\big)\\
m^t_{F\to v} &= \mathrm{AGG}\big(\{\!\{h^{t-1}_f \mid v \in f, f \in F\}\!\}\big)\\
m^t_{\mathcal{V}\to f} &= \mathrm{AGG}\big(\{\!\{h^{t-1}_v \mid v \in f, v \in \mathcal{V}\}\!\}\big)
\end{aligned}
$$

where $N_F(f)$ denotes the neighbors of fragment f in the higher-level graph of fragments. The hidden representations are updated by combining the incoming messages with the previous hidden representation:

$$
\begin{aligned}
h^t_v &= \mathrm{MLP}(h^{t-1}_v, m^t_{\mathcal{V}\to v}, m^t_{F\to v})\\
h^t_f &= \mathrm{MLP}(h^{t-1}_f, m^t_{\mathcal{V}\to f}, m^t_{F\to f})\\
h^t_e &= \mathrm{MLP}(h^{t-1}_e, h^{t-1}_u, h^{t-1}_v)
\end{aligned}
$$

The final graph-level readout after T layers is computed by aggregating the multiset of node representation, edge representations, and fragment representations:

$$
\mathrm{OUT}(\{\!\{h^T_v \mid v \in \mathcal{V}\}\!\}, \{\!\{h^T_e \mid e \in \mathcal{E}\}\!\}, \{\!\{h^T_f \mid f \in F\}\!\})
$$

Note that the complexity of our FragNet model is linear in the number of nodes and fragments (assuming that each node is only part of a constant number of fragments).

Additionally, our FragNet model achieves the highest expressiveness in our Fragment-WL hierarchy and also compared to other fragment-biased GNNs.

Theorem 5.1. *FragNets are at most as powerful as HLG-WL. Additionally, when using injective neighborhood aggregators and a sufficient number of layers, FragNets are as powerful as HLG-WL.*

5.2. Molecular fragmentation

Apart from the question of how to use a fragmentation, there is the equally important question of how to fragment the graph in the first place. A fragmentation F has to fulfill two seemingly conflicting goals:

1. F should contain all *important* substructures.
2. F should facilitate generalization.

On the one hand, if the fragmentation is too coarse-grained, important structural features that the model cannot learn may be missed. On the other hand, if the fragmentation is too fine-grained, it is harder to find similarities between graphs, and it exposes the model to the risk of overfitting. Of course, the right level of detail and the right fragmentation scheme critically depend on the application domain. Existing approaches to fragment molecules focus either only on a single substructure (Zhu et al., 2022) or fragment the molecule based on chemical properties (Degen et al., 2008b), which requires a huge vocabulary.

Our approach is capable of fragmenting the complete graph with only two classes of substructures: rings and paths. At a conceptual level, first, all minimal rings are extracted. Second, the remaining edges are connected at nodes of degree two to form paths. Third, whenever three or more fragments meet at a node, a junction node is introduced in the higher-level graph and the fragments are only connected to the junction nodes. This prevents most cycles in the higher-level graph. Figure 3 shows our fragmentation on an example molecule.

Since our rings-paths fragmentation scheme recovers 5-cycles, we derive the following Corollary from Theorem 4.9.

Corollary 5.2. *FragNet with our rings-paths fragmentation scheme is in parts more powerful than 3-WL.*

While this fragmentation increases the expressiveness and makes it possible to fragment the complete graph, it could require a very large vocabulary, as it has to include paths and cycles of any size found in the dataset. To still effectively encode the fragment type, we introduce a novel ordinal encoding.

Ordinal encoding. Previous works either use no encoding for the types of fragments (Fey et al., 2020) or a simple one-hot encoding (Bouritsas et al., 2023). However, to facilitate the generalization capabilities of a model, the encodings of similar fragments should also be similar. In our approach, we introduce an ordinal fragment encoding, which accomplishes this by incorporating two embeddings: one for the fragment class (i.e., $\text{class}(f) \in \{\text{path}, \text{cycle}, \text{junction}\}$) and another that is proportionally scaled based on the fragment size; see Figure 4. More formally:

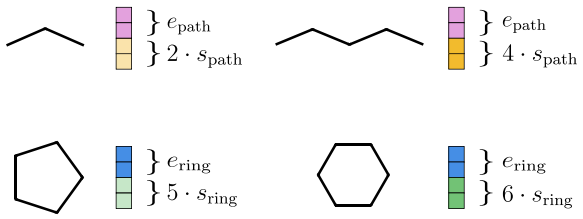

Figure 4. Ordinal encoding applied to a 2-path, 4-path, 5-ring, and 6-ring. The encoding comprises two components: one learned embedding e for every fragment class (i.e., path, cycle, or junction) and another learned embedding s that is proportionally scaled based on the fragment size.

$$h_f^0 = (e_{\text{class}(f)}, |f| \cdot s_{\text{class}(f)}),$$

where s and e are different learned embeddings for the classes of cycle, path, and junction fragments. This approach enables the encoding of an infinite vocabulary, accommodating even completely unseen fragments while concurrently supporting effective model generalization.

5.3. Long-range interaction

Many graph-related tasks demand the capability to capture long-range interactions among distant nodes (Dwivedi et al., 2023). However, GNNs suffer from a phenomenon called *over-squashing*, where nodes are insensitive to information from distant nodes. This occurs because, in GNNs, the computational graph is equal to the input graph, requiring L layers to exchange information between nodes separated by a distance of L. Di Giovanni et al. (2023) show that increasing the number of layers does not help prevent this phenomenon, as GNNs suffer from vanishing gradients. Additionally, their work introduces the commute time between nodes as an indicator for over-squashing.

Our rings-paths fragmentation, together with the higher-level graph, create shortcuts that can help mitigate over-squashing. Messages can traverse the smaller, higher-level graph of fragments, while the message path still has semantic meaning. In Section 6, we demonstrate that our rings-paths fragmentation reduces the commute time in both synthetic and real-world datasets, enhances the recoverability of messages from distant nodes, and shows strong performance on benchmarks requiring long-range interactions.

6. Results

While we have theoretically demonstrated that our model attains the highest expressiveness within our Fragment-WL hierarchy, we also empirically evaluate its expressiveness by

Table 1. Our model predicting the number of different substructures occurring in ZINC test dataset.

Fragment					
Counts	5629	3904	1799	1772	1071
Accuracy	0.986	0.99	0.999	1.0	0.997

examining its ability to count substructures. Additionally, we explore its ability to communicate over long distances, its overall predictive effectiveness, and its capacity to generalize.

Expressiveness. To evaluate how well our model can learn to recognize chemically *important* substructures in molecular graphs, we first identify the most common substructures in the ZINC 10k dataset (Gómez-Bombarelli et al., 2017) using a chemically-inspired fragmentation scheme, specifically MAGNet (Hetzel et al., 2023). Subsequently, we train our model to predict substructure counts. Our model is able to identify all substructures close to perfection as demonstrated in Table 1 on a subset of fragments; performance details for all substructures are available in Appendix B. Notably, our model achieves high accuracy even for intricate substructures not present in our vocabulary. This underscores that our fragmentation, based solely on rings and paths, together with our ordinal encoding and the higher-level message passing, proves sufficient for the model to recognize more complex substructures. This is essential for application in, e.g., fragment-based molecule generation, as the task of the encoder is to encode information about such substructures.

Long-Range Interactions. Inspired by Di Giovanni et al. (2023), we investigate the long-range capacity by training models to recover messages sent from a designated node to another node in a graph. As over-squashing is particularly prone to happen at bottlenecks (Topping et al., 2022), we are considering a graph consisting of two cycles connected by a bottleneck in the form of a path. This graph structure also matches many substructures found in molecules.

Figure 5 shows the recovery rates for a GNN using no fragmentation, a ring fragmentation, as used by CIN (Bodnar et al., 2022), and our rings-paths fragmentation. A standard GNN is able to recover messages from the neighborhood perfectly, but the performance deteriorates with increasing distance. After the path bottleneck, the performance is equivalent to random guessing. A model with rings fragmentation exhibits a behavior very similar to the baseline. In particular, it performs worse on one node on the path. We assume that this happens due to the additional messages on the rings, creating even more messages that traverse the bottleneck. For our fragmentation with the higher-level graph,

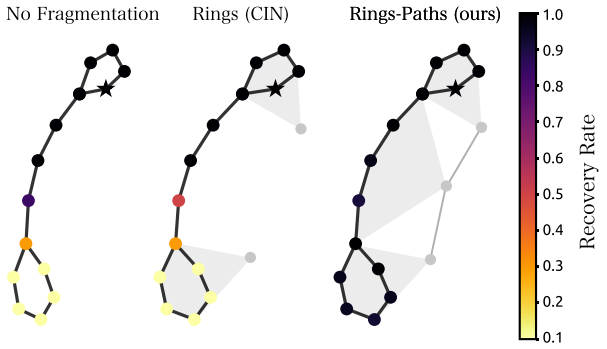

Figure 5. Recovery rate of messages sent from the star node to all other nodes. A recovery rate of 0.1 corresponds to random guessing. The first graph has no fragmentation, the second one a rings fragmentation (like in CIN/CIN++), the third a rings and paths fragmentation (like in our model).

we observe that all nodes can recover the messages nearly perfectly. The results closely mirror the commute times on those graphs (see Figure 7 in Appendix B), validating the theoretical work by Di Giovanni et al. (2023) that proposed the commute time as an indicator for over-squashing.

Predictive Performance. To evaluate the predictive performance on real-world molecular dataset, we use the long-range peptides benchmark (Dwivedi et al., 2022) and the large-scale molecular benchmark ZINC (Sterling & Irwin, 2015). We compare our model against standard GNNs, like **GCN** (Kipf & Welling, 2017), **GIN** (Xu et al., 2019) or **GatedGCN** (Bresson & Laurent, 2018), higher-order GNNs such as **RingGNN** (Chen et al., 2023) or **3WL-GNN** (Maron et al., 2019), topological GNNs, especially **CIN** and **CIN++** (Bodnar et al., 2021; 2022) and other fragment-biased GNNs such as **HIMP** (Fey et al., 2020) and **GSN** (Bouritsas et al., 2023). While our main focus lies on the evaluation against other message-passing GNNs, especially the group of fragment-biased GNNs, we compare against Transformer architectures for completeness. We test **Graphormer** (Ying et al., 2021), **GPS** (Rampášek et al., 2023), **GT** (Dwivedi & Bresson, 2021), **SAN** (Kreuzer et al., 2021) and **GRIT** (Ma et al., 2023).

A summary of the used hyperparameters of our model and the experimental details for each experiment can be found in Appendix C. All our models adhere to the 500k parameter budget for both datasets. We do not use any additional feature augmentation, such as positional encodings.

To further investigate the long-range capabilities of our model, we empirically measure the performance on the Long-Range Graph Benchmark (LRGB), where the regression performance depends on these capabilities. The datasets we use are the LRGB Peptides-struct and Peptides-func (Dwivedi et al., 2022). The peptides datasets comprise

Table 2. Predictive performance for multiple models on Peptides-struct and func. Best Transformer and best GNN are highlighted.

Type	Model	Peptides-	
		Struct (MAE ↓)	**Func** (AP ↑)
Transformer	GPS	0.2500 ± 0.0012	0.6535 ± 0.0041
	SAN+LapPE	0.2683 ± 0.0043	0.6384 ± 0.0121
	SAN+RWSE	0.2545 ± 0.0012	0.6439 ± 0.0075
	GRIT	**0.2460** ± 0.0012	**0.6988** ± 0.0082
Basic GNNs	GCN	0.3496 ± 0.0013	0.5930 ± 0.0023
	GIN	0.3547 ± 0.0045	0.5498 ± 0.0079
	GatedGCN	0.3420 ± 0.0013	0.5864 ± 0.0035
	GatedGCN+RWSE	0.3357 ± 0.0006	0.6069 ± 0.0035
Topological	CIN++	0.2523 ± 0.0013	0.6569 ± 0.0117
Fragment-Biased	HIMP	0.2503 ± 0.0008	0.5668 ± 0.0149
	FragNet (ours)	**0.2462** ± 0.0021	**0.6678** ± 0.005

15535 peptides, each with an average of 150 nodes, that exhibit long amino acid chains with a lower average degree. As the molecules have a strong variability in their diameters, the models must generalize well to graphs of different sizes. These datasets are commonly utilized to benchmark the long-range capabilities of graph transformers against GNNs.

The peptides-struct task is a multi-label graph regression dataset where the objective is to predict various structural properties of the 3-dimensional molecules, including spatial lengths. Note that the node and edge features of the graphs do not contain any 3D information. Performance is evaluated using the mean absolute error (MAE). The peptides-func dataset consists of the same graphs, but requires the models to predict functional properties of the peptides. It is a multi-label classification task with 10 classes such as *Antiviral* or *Antibacterial*. The performance metric is the unweighted mean average precision (AP).

The results presented in Table 2 demonstrate that our model achieves state-of-the-art performance for GNNs on both datasets. Notably, our model surpasses almost all graph transformers, even though transformers have an advantage in capturing long-range interactions due to their consideration of messages between all pairs of nodes, resulting in quadratic complexity. GRIT stands out as the only model that surpasses our performance, displaying exceptional results across all datasets.

Furthermore, we evaluate our model on ZINC (Sterling & Irwin, 2015), a collection of chemical compounds. The benchmark consists of a dataset that contains only a subset of 10,000 molecules and a dataset consisting of all 250,000 molecules. For both, we predict the penalized logP, which characterizes the drug-likeness of a molecule (Gómez-Bombarelli et al., 2017), and measure performance using MAE.

Table 3. Predictive performance for multiple models on ZINC 10k and ZINC full. Best Transformer and best GNN are highlighted.

Type	Model	ZINC	
		10k (MAE ↓)	**Full** (MAE ↓)
Transformer	Graphormer	0.122 ± 0.006	0.052 ± 0.005
	GPS	0.070 ± 0.006	-
	GT	0.226 ± 0.014	-
	SAN	0.139 ± 0.006	-
	Graphormer-URPE	0.086 ± 0.007	0.028 ± 0.002
	Graphormer-GD	0.081 ± 0.009	0.025 ± 0.004
	GRIT	**0.059** ± 0.002	**0.023** ± 0.001
Basic GNNs	GCN	0.367 ± 0.011	0.113 ± 0.002
	GIN	0.526 ± 0.051	0.088 ± 0.002
	GAT	0.384 ± 0.007	0.111 ± 0.002
	GraphSAGE	0.398 ± 0.002	0.126 ± 0.003
Higher-order	RingGNN	0.353 ± 0.019	-
	3WLGNN	0.303 ± 0.068	-
Topological	CIN-Small	0.094 ± 0.004	0.044 ± 0.003
	CIN++	**0.077** ± 0.004	0.027 ± 0.007
Fragment-Biased	HIMP	0.151 ± 0.006	0.036 ± 0.002
	GSN	0.115 ± 0.012	-
	Autobahn	0.106 ± 0.004	0.029 ± 0.001
	FragNet (ours)	**0.0775** ± 0.005	**0.0237** ± 0.00

As shown in Table 3, basic GNNs, which lack both high expressiveness and strong long-range capacity, exhibit significantly inferior performance. All fragment-biased approaches outperform the higher-order GNNs despite being less computationally demanding. Our model shows the best performance among all GNNs, with very similar results to CIN++ on ZINC-subset. In addition, our model outperforms most transformer architectures as well and achieves results comparable to those of GRIT on the full dataset.

Generalization. To test the generalization capabilities of our model with the ordinal fragment encoding, we use a test set containing out-of-distribution molecules with completely unseen fragments. For this, we use the ZINC dataset and remove all molecules containing a 7-ring from the training data. After training, we test on all molecules from the test set, thus also containing 7-rings that were not seen during training. The results in Table 4 demonstrate that our model achieves an error 1.8 times lower than GRIT, showcasing the superior generalization capabilities. Our better generalization capabilities can also be seen in the normal ZINC benchmark. In Table 5, we group the ZINC dataset into groups based on the

Table 4. We remove all molecules that contain 7-rings from the training set and test on all molecules, i.e., also the ones with 7-rings.

Model	ZINC 10k	
	training (MAE ↓)	**test** (MAE ↓)
GRIT	0.02	0.61
FragNet (ours)	0.08	**0.34**

Table 5. Comparison of the MAE of GRIT and our model on ZINC-full. Graphs are grouped by the frequency of their rarest fragment.

Frequency of rarest Fragment	$< 0.1\%$	$< 1\%$	$< 10\%$	$\geq 10\%$
HIMP (MAE)	14.4	0.48	0.15	0.030
GRIT (MAE)	9.5	0.26	0.026	0.018
FragNet (ours) (MAE)	5.3	0.15	0.045	0.021

frequency of the rarest fragment. Our model outperforms HIMP everywhere and GRIT for graphs containing rare fragments.

Lastly, we test our model's capability to transfer the knowledge to a completely different dataset. We train on ZINC and predict the penalized logP on QM9 (Wu et al., 2017). In Table 6, we see that FragNet achieves the lowest MAE of 1.12, outperforming GRIT and HIMP, suggesting that our model generalizes better to unseen data distributions due to our inductive bias and corresponding fragmentation. In summary, we showcased the generalization capabilities of our model on both a completely unseen dataset and a slightly shifted data distribution. The generalization capabilities also help our model perform better on rare fragments.

Table 6. Zero shot generalization to QM9

Model	**QM9** (MAE ↓)
HIMP	3.43
GRIT	1.22
FragNet (ours)	**1.12**

7. Limitations

While our method has strong predictive performance, it can further be improved: first, our fragmentation method and ordinal encoding is tailored towards molecules and not applicable to large densely connected graphs such as citation or social networks as one will find vast amount of meaningless fragments, introducing a lot of noise. Second, while our generalization experiments show superior performance over GRIT, Table 5 also shows that GRIT performs slightly better on molecules with frequent fragments. We leave it for future work to further improve fragment-biased models on those data.

8. Conclusion

In this work, we proposed a new expressivity measure, the Fragment-WL test, which provides a hierarchy on existing fragment-biased GNNs. Based on these insights, we proposed an expressive new model, which, together with our new fragmentation, outperforms all GNN approaches and most transformer architectures. Moreover, our model demonstrates predictive performance comparable to that of the best transformer model, while surpassing it in terms of generalization capabilities, despite having only linear complexity. This positions our approach as a robust solution for a variety of molecular modeling tasks.

We believe that our work lays the foundation for several promising future directions. These include extending the expressivity hierarchy to, for example, incorporate orbit information (Bouritsas et al., 2023). In addition, future work could seek to further improve the predictive performance on frequent data, or using fragment-biases in multi-task or meta-learning settings.

Acknowledgements

This project is supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy with funds from the Hightech Agenda Bayern. Additionally, it is supported by the Helmholtz Association under the joint research school "Munich School for Data Science - MUDS" and by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036B.

Impact Statement

Among other contributions, this work presents an approach for predicting the properties of molecules. In the area of machine learning for drug discovery, such methods can sometimes be used for harmful purposes. This also applies to our research, since it might help to discover or create dangerous substances. Despite these concerns, we believe that the benefits of our work outweigh the risks.

References

Bemis, G. W. and Murcko, M. A. The properties of known drugs. 1. molecular frameworks. *Journal of Medicinal Chemistry*, 1996.

Bodnar, C., Frasca, F., Wang, Y. G., Otter, N., Montúfar, G., Liò, P., and Bronstein, M. Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks, 2021.

Bodnar, C., Frasca, F., Otter, N., Wang, Y. G., Liò, P., Montúfar, G., and Bronstein, M. Weisfeiler and Lehman Go Cellular: CW Networks, 2022.

Bouritsas, G., Frasca, F., Zafeiriou, S., and Bronstein, M. M. Improving Graph Neural Network Expressivity via Subgraph Isomorphism Counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Bresson, X. and Laurent, T. Residual gated graph convnets, 2018.

Campi, F., Gosch, L., Wollschläger, T., Scholten, Y., and

Günnemann, S. Expressivity of Graph Neural Networks Through the Lens of Adversarial Robustness, 2023.

Chen, Z., Chen, L., Villar, S., and Bruna, J. Can Graph Neural Networks Count Substructures?, 2020.

Chen, Z., Villar, S., Chen, L., and Bruna, J. On the equivalence between graph isomorphism testing and function approximation with gnns, 2023.

Degen, J., Wegscheid-Gerlach, C., Zaliani, A., and Rarey, M. On the art of compiling and using 'drug-like' chemical fragment spaces. *ChemMedChem*, 2008a.

Degen, J., Wegscheid-Gerlach, C., Zaliani, A., and Rarey, M. On the art of compiling and using 'drug-like' chemical fragment spaces. *ChemMedChem*, 2008b.

Di Giovanni, F., Giusti, L., Barbero, F., Luise, G., Lio', P., and Bronstein, M. On Over-Squashing in Message Passing Neural Networks: The Impact of Width, Depth, and Topology, 2023.

Du, Y., Fu, T., Sun, J., and Liu, S. Molgensurvey: A systematic survey in machine learning models for molecule design. *arXiv preprint arXiv:2203.14500*, 2022.

Dwivedi, V. P. and Bresson, X. A Generalization of Transformer Networks to Graphs, 2021.

Dwivedi, V. P., Joshi, C. K., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X. Benchmarking Graph Neural Networks, 2022.

Dwivedi, V. P., Rampášek, L., Galkin, M., Parviz, A., Wolf, G., Luu, A. T., and Beaini, D. Long range graph benchmark, 2023.

Falcon, W. and The PyTorch Lightning team. PyTorch Lightning, 2019.

Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Fey, M., Yuen, J.-G., and Weichert, F. Hierarchical Inter-Message Passing for Learning on Molecular Graphs, 2020.

Frasca, F., Bevilacqua, B., Bronstein, M. M., and Maron, H. Understanding and Extending Subgraph GNNs by Rethinking Their Symmetries, 2022.

Gasteiger, J., Groß, J., and Günnemann, S. Directional message passing for molecular graphs. *International Conference on Learning Representations*, 2020.

Geisler, S., Li, Y., Mankowitz, D., Cemgil, A. T., Günnemann, S., and Paduraru, C. Transformers meet directed graphs, 2023.

Geng, Z., Xie, S., Xia, Y., Wu, L., Qin, T., Wang, J., Zhang, Y., Wu, F., and Liu, T.-Y. De novo molecular generation via connection-aware motif mining. *International Conference on Learning Representations*, 2023.

Giusti, L., Reu, T., Ceccarelli, F., Bodnar, C., and Liò, P. CIN++: Enhancing Topological Message Passing, 2023.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules, 2017.

He, X., Hooi, B., Laurent, T., Perold, A., Lecun, Y., and Bresson, X. A Generalization of ViT/MLP-Mixer to Graphs. In *International Conference on Machine Learning*, 2023.

Hetzel, L., Sommer, J., Rieck, B., Theis, F. J., and Günnemann, S. MAGNet: Motif-Agnostic Generation of Molecules from Shapes. *arXiv*, 2023.

Hu, W., Liu, Y., Chen, X., Chai, W., Chen, H., Wang, H., and Wang, G. Deep learning methods for small molecule drug discovery: A survey. *IEEE Transactions on Artificial Intelligence*, 2023.

Huang, N. and Villar, S. A short tutorial on the weisfeiler-lehman test and its variants. *International Conference on Acoustics Speech and Signal Processing*, 2021.

Huang, Y., Peng, X., Ma, J., and Zhang, M. Boosting the cycle counting power of graph neural networks with I 2-GNNs. *International Conference on Learning Representations*, 2022.

Inae, E., Liu, G., and Jiang, M. Motif-aware attribute masking for molecular graph pre-training. *arXiv preprint arXiv:2309.04589*, 2023.

Jin, W., Barzilay, R., and Jaakkola, T. Junction Tree Variational Autoencoder for Molecular Graph Generation, 2019.

Jin, W., Barzilay, R., and Jaakkola, T. Hierarchical generation of molecular graphs using structural motifs. In *International Conference on Machine Learning*. PMLR, 2020.

Kiefer, S. and Neuen, D. The power of the weisfeiler–leman algorithm to decompose graphs. *SIAM Journal on Discrete Mathematics*, 2022.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks, 2017.

Kong, X., Huang, W., Tan, Z., and Liu, Y. Molecule Generation by Principal Subgraph Mining and Assembling, 2022.

Kreuzer, D., Beaini, D., Hamilton, W. L., Létourneau, V., and Tossou, P. Rethinking graph transformers with spectral attention, 2021.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.

Ma, L., Lin, C., Lim, D., Romero-Soriano, A., Dokania, P. K., Coates, M., Torr, P., and Lim, S.-N. Graph Inductive Biases in Transformers without Message Passing, 2023.

Maron, H., Ben-Hamu, H., Serviansky, H., and Lipman, Y. Provably powerful graph networks. *Advances in neural information processing systems*, 2019.

Maziarz, K., Jackson-Flux, H., Cameron, P., Sirockin, F., Schneider, N., Stiefl, N., Segler, M., and Brockschmidt, M. Learning to Extend Molecular Scaffolds with Structural Motifs, 2022.

Merlot, C., Domine, D., Cleva, C., and Church, D. J. Chemical substructures in drug discovery. *Drug Discovery Today*, 2003.

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks, 2021.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*. 2019.

Puny, O., Lim, D., Kiani, B. T., Maron, H., and Lipman, Y. Equivariant polynomials for graph neural networks, 2023.

Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. Recipe for a General, Powerful, Scalable Graph Transformer, 2023.

Singh, R. and Saha, M. Identifying structural motifs in proteins. *Pacific Symposium on Biocomputing*, 2003.

Sommer, J., Hetzel, L., Lüdke, D., Theis, F. J., and Günnemann, S. The Power of Motifs as Inductive Bias for Learning Molecular Distributions, 2023.

Sterling, T. and Irwin, J. J. Zinc 15 – ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 2015.

Thiede, E. H., Zhou, W., and Kondor, R. Autobahn: Automorphism-based graph neural nets. *ArXiv*, 2021.

Topping, J., Di Giovanni, F., Chamberlain, B. P., Dong, X., and Bronstein, M. M. Understanding over-squashing and bottlenecks on graphs via curvature, 2022.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. S. Moleculenet: A benchmark for molecular machine learning. *arXiv: Learning*, 2017.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How Powerful are Graph Neural Networks?, 2019.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T.-Y. Do Transformers Really Perform Bad for Graph Representation?, 2021.

Zang, X., Zhao, X., and Tang, B. Hierarchical molecular graph self-supervised learning for property prediction. *Communications Chemistry*, 2023.

Zhang, B., Fan, C., Liu, S., Huang, K., Zhao, X., Huang, J., and Liu, Z. The expressive power of graph neural networks: A survey. *arXiv preprint arXiv:2308.08235*, 2023.

Zhang, Z., Liu, Q., Wang, H., Lu, C., and Lee, C.-K. Motif-based graph self-supervised learning for molecular property prediction. *Advances in Neural Information Processing Systems*, 2021.

Zhu, J., Wu, K., Wang, B., Xia, Y., Xie, S., Meng, Q., Wu, L., Qin, T., Zhou, W., Li, H., and Liu, T.-Y. $\mathcal{O}$-GNN: incorporating ring priors into molecular modeling, 2022.

A. Proofs

This chapter presents the proofs for the theorems introduced in Section 4, and the expressiveness analysis of existing fragment-biased and topological GNNs. We will first introduce general concepts that will later help to bound the expressiveness of different models and our Fragment WL test.

A.1. Color refinement and expressiveness: Useful definitions and lemmas

To prove the power of different graph coloring algorithms, it will be useful to first introduce the definition of color refinement. The intuition is that a "finer" coloring contains more information than a "corser" coloring.

Definition A.1. Let c, d be colorings of a graph G. The coloring c refines d (we write $c \sqsubseteq d$) if there exists a function h such that $h(c_v) = d_v$ for all $v \in \mathcal{V}$.

We will sometimes write $c_v \sqsubseteq d_v$ if $c \sqsubseteq d$ and the set of vertices $\mathcal{V}$ is clear from the context.

Example A.1. Let $c^{(t)}$ be the coloring of iteration t of the WL test. Then one can easily show that $c_v^{(t)} \sqsubseteq c_v^{(l)}$ for $l \leq t$ as $c_v^{(t)}$ contains the information of all previous colorings $c_v^{(l)}$.

Note that an alternative definition of color refinement (Bodnar et al., 2021) is: $c \sqsubseteq d$ iff $c_v = c_w$ implies $d_v = d_w$ for all $v, w \in \mathcal{V}$. It is easy to see that the two definitions are equivalent. Next, we extend our definition to arbitrary functions and not just colorings.

Definition A.2. Let a, b be two functions over the set of vertices $\mathcal{V}$. Then $a \sqsubseteq b$ if there exists a function h such that $h(a(v)) = b(v)$ for all $v \in \mathcal{V}$.

Intuitively, $a \sqsubseteq b$ means that we can compute $b(v)$ from the result of $a(v)$. So $a(v)$ contains more or the same information as $b(v)$. Again, we will sometimes simply write $a_v \sqsubseteq b_v$ with $a_v := a(v)$, $b_v := b(v)$ if $a \sqsubseteq b$ and if the set of vertices $\mathcal{V}$ is clear from the context.

Example A.2. Let $c^{(t)}$ be the coloring of iteration t of the WL test. Then, because of the injectiveness of the hash function HASH in Equation (1):

$$c_v^{(t)} \sqsubseteq \{\!\{c_w^{(t-1)} \mid w \in \mathcal{N}(v)\}\!\}.$$

Note that we use the simplified notation here. The right and left-hand side are actually the functions that map from $v \in \mathcal{V}$ to these terms. Intuitively, this shows that one can compute the previous color of all neighbors from the color of a node.

It is easy to see that the refinement relation is transitive, i.e., $a \sqsubseteq b$ and $b \sqsubseteq c$ imply $a \sqsubseteq c$.

We will now formally define the expressive power of an algorithm with respect to the ability to distinguish non-isomorphic subgraphs.

Definition A.3. A function f is (in parts) *more powerful* than a function g if there exist two non-isomorphic graphs G^1, G^2 such that f can distinguish them

$$f(G^1) \neq f(G^2)$$

whereas g cannot distinguish them

$$g(G^1) = g(G^2).$$

Note that this relation is *not* anti-symmetric, i.e. f can be (in parts) more powerful than g, and conversely, g can also be (in parts) more powerful than f. Hence, we introduce the following stronger anti-symmetric relation:

Definition A.4. A function f is *strictly more powerful* than g (denoted as $f > g$) if

1. f is more powerful than g
2. and g is not (in parts) more powerful than f.

Additionally, we write we write $g \leq f$ if a function g is not more powerful than f.

Next, we will prove a connection between color refinement and expressiveness: a function that always produces a finer coloring cannot be less powerful than a function with a coarser coloring.

Lemma A.5. *Let f, g be functions with $f \sqsubseteq g$ for all graphs. Then, g is not more powerful than f, i.e., $g \leq f$.*

Proof. Assume for the sake of contradiction that there exist non-isomorphic graphs G^1, G^2 that can be distinguished by g but not by f. Let d be the coloring obtained with g, and c be the coloring obtained with f. The multiset of colors d has to differ for G^1 and G^2, i.e. there exists a color α with

$$D^1_\alpha := \{v \mid d_v = \alpha,\ v \in \mathcal{V}^1\}$$
$$D^2_\alpha := \{v \mid d_v = \alpha,\ v \in \mathcal{V}^2\}$$

such that

$$|D^1_\alpha| \neq |D^2_\alpha|. \tag{3}$$

Since c refines d no node v in D^1_α and D^2_α can share a color c_v with another node not in D^1_α and D^2_α. Hence, the set of colors c of nodes in D^1_α and D^2_α is disjoint from the set of colors c for the other nodes in the graph. But because of 3 there has to exist a color β with

$$C^1_\beta := \{v \mid c_v = \beta,\ v \in \mathcal{V}^1\} \subseteq D^1_\alpha$$
$$C^2_\beta := \{v \mid c_v = \beta,\ v \in \mathcal{V}^2\} \subseteq D^2_\alpha$$

and

$$|C^1_\beta| \neq |C^2_\beta|.$$

This contradicts the initial assumption. □

Note that all our augmentation functions introduced in Section 4 only add information to the graph, or more formally:

Definition A.6. A function g from graphs to graphs is called additive if the set of nodes and edges does not decrease, i.e., $g(\mathcal{V}, \mathcal{E}, \boldsymbol{X}) = (\mathcal{V}', \mathcal{E}', \boldsymbol{X}')$ with $\mathcal{V} \subseteq \mathcal{V}'$ and $\mathcal{E} \subseteq \mathcal{E}'$.

But by adding too much information, one could completely destroy the initial structure of the graph (e.g., make every graph a complete graph). Hence, we need the additional condition that it should be possible to recover the original graph from the augmented graph.

Definition A.7. An additive function g from graphs to graphs (i.e, $g(G) = G'$) is called reversible if there exists a function h for vertices $v \in \mathcal{V}'$ such that

$$h(\boldsymbol{X}'_v) = \begin{cases} \boldsymbol{X}_v & v \in \mathcal{V} \\ \perp & v \notin \mathcal{V} \end{cases} \tag{4}$$

and a function ε for edges $e = \{u, v\} \in \mathcal{E}'$ such that

$$\varepsilon(\boldsymbol{X}'_u, \boldsymbol{X}'_v) = \begin{cases} 1 & e \in \mathcal{E} \\ 0 & e \notin \mathcal{E}. \end{cases} \tag{5}$$

By only adding such reversible information the WL test cannot become less powerful:

Lemma A.8. *Let g be a reversible function. Then g-WL is not less powerful than WL.*

Proof. Let $c^{(t)}$ be the coloring obtained by the t-th iteration of the WL-test, and $d^{(t)}$ the coloring of g-WL. We will show by induction that there exists a function h such that $h(d^{(t)}_v) = c^{(t)}_v$ for $v \in \mathcal{V}$, i.e $d^{(t)}_v \sqsubseteq c^{(t)}_v$. For $t = 0$, this follows immediately from Equation (4). For the induction step, note

$$d^{(t)}_v \sqsubseteq \{\!\{d^{(t-1)}_u \mid u \in \mathcal{N}_{G'}(v)\}\!\}.$$

Now note that the function ε (Equation (5)) makes it possible to reconstruct the neighborhood in G based on the features $\boldsymbol{X}'$ and neighborhood in G'. Since $d^{(t)}$ only refines the features $\boldsymbol{X}'$, we can also reconstruct the neighborhood in G based on $d^{(t)}$ and, hence,

$$\{\!\{d_u^{(t-1)} \mid u \in \mathcal{N}_{G'}(v)\}\!\} \sqsubseteq \{\!\{d_u^{(t-1)} \mid u \in \mathcal{N}_G(v)\}\!\}$$

and by induction hypothesis

$$\{\!\{d_u^{(t-1)} \mid u \in \mathcal{N}_G(v)\}\!\} \sqsubseteq \{\!\{c_u^{(t-1)} \mid u \in \mathcal{N}_G(v)\}\!\}.$$

So, taken together, we have

$$d_v^{(t)} \sqsubseteq \{\!\{c_u^{(t-1)} \mid u \in \mathcal{N}_G(v)\}\!\}. \tag{6}$$

Additionally, note that

$$d_v^{(t)} \sqsubseteq d_v^{(t-1)} \overset{\text{IH}}{\sqsubseteq} c_v^{(t-1)}. \tag{7}$$

By combining Equation (6) and Equation (7), we get

$$d_v^{(t)} \sqsubseteq \left(c_v^{(t-1)},\ \{\!\{c_u^{(t-1)} \mid u \in \mathcal{N}_G(v)\}\!\}\right) \sqsubseteq c_v^{(t)}.$$

Hence, d refines c, and by Lemma A.5 g-WL is not less powerful than WL. □

Now that we have found a lower bound of the expressiveness, we will also give an upper bound of the expressiveness. The idea is that a graph augmentation function does not increase the power of a coloring algorithm f (e.g., the WL test) if all the information added by g can also be computed from the coloring obtained with f. Or, to put it differently, a graph augmentation function g does not increase the expressiveness if it is possible to compute the set of colors on $g(G)$ from the set of colors on G.

Lemma A.9. *Let g be a function that augments a graph, i.e., a function from graphs to graphs. Let f be a coloring function. If there exists a function h such that*

$$f(g(G)) = h(f(G))$$

then $f \circ g$ is not more powerful than f.

Proof. Let G^1, G^2 be two non-isomorphic graphs that are distinguishable by $f \circ g$. Then

$$f(g(G^1)) \neq f(g(G^2))$$
$$h(f(G^1)) \neq h(f(G^2))$$

It follows that $f(G^1) \neq f(G^2)$. Hence, $f \circ g$ is not more powerful than f. □

Note that the condition in the lemma is similar to the definition of color refinement. However, we cannot use color refinement directly because the function g could add or delete nodes, making a direct comparison of nodes between G^1 and G^2 impossible. Consequently, we have to use a more global view rather than the more localized approach of color refinement.

A.2. Graph augmentation functions

We will now analyze the change in expressiveness with some graph augmentation functions that model message-passing schemes that are frequently used in practice.

A.2.1. Fragment augmentations

We will first give the proofs for the augmentation functions from Section 4 that incorporate fragment information.

Proof of Theorem 4.5

Theorem 4.5. *There exist fragmentation schemes such that NF-WL, FR-WL and HLG-WL are all strictly more powerful than k-WL for any k.*

Proof. Consider a fragmentation scheme that decomposes a graph into every possible subgraph (so the vocabulary is the set of all possible graphs). It is known that for every k there exist two non-isomorphic graphs G^1 and G^2 that are indistinguishable by k-WL. But as G^1 and G^2 are themselves part of the vocabulary and the fragmentation, they are trivially distinguishable by NF-WL, FR-WL, and HLG-WL. □

Proof of Theorem 4.6

Theorem 4.6. *NF-WL is strictly more powerful than 2-WL for fragmentation schemes $\mathcal{F}$ that recover any substructure with more than two nodes.*

Proof. Chen et al. (2020) showed that the WL test cannot count the number of (induced) subgraphs with at least three nodes, i.e. for any substructure S with more than three nodes, there exist non-isomorphic graphs G^1 and G^2 such that 2-WL cannot distinguish them but they have a different count of S. So, let X be the substructure that $\mathcal{F}$ recovers. Then, WL cannot count the number of occurrences of X, whereas NF-WL can trivially count the number of X in a graph. Hence, NF-WL is more powerful than WL. Since NF is a reversible function, NF-WL is by Lemma A.8 also not less powerful than WL, making NF-WL strictly more powerful than WL. □

Proof of Theorem 4.7 Before coming to the proof of Theorem 4.7, we will prove the following useful lemma:

Lemma A.10. *Two graphs $G^1 = (\mathcal{V}^1, \mathcal{E}^1, \boldsymbol{X}^1)$, $G^2 = (\mathcal{V}^2, \mathcal{E}^2, \boldsymbol{X}^2)$ are undistinguishable by WL if the set of node features is the same*

$$\boldsymbol{X}^1 = \boldsymbol{X}^2 := \boldsymbol{X}$$

and all nodes with the same node feature have the same neighborhood

$$\begin{aligned}&\forall i, j \in \mathcal{V}^1 \cup \mathcal{V}^2 : \\ &\boldsymbol{X}_i = \boldsymbol{X}_j \Rightarrow \{\boldsymbol{X}_n \mid n \in \mathcal{N}(i)\} = \{\boldsymbol{X}_m \mid m \in \mathcal{N}(j)\}.\end{aligned} \tag{8}$$

Proof. We will show by induction over t that the color of all nodes with the same node features is the same:

$$\begin{aligned}&\forall i, j \in \mathcal{V}^1 \cup \mathcal{V}^2 : \\ &\boldsymbol{X}_i = \boldsymbol{X}_j \Rightarrow c_i^t = c_j^t.\end{aligned}$$

For $t = 0$, this follows immediately from $c_i^0 = \text{HASH}(\boldsymbol{X}_i)$.
For $t > 0$, we have for nodes i, j with $X_i = X_j$

$$\begin{aligned}c_i^t &= \text{HASH}\big(c_i^{t-1}, \{c_n^{t-1} \mid n \in N(i)\}\big) \\ &= \text{HASH}\big(c_j^{t-1}, \{c_n^{t-1} \mid n \in N(i)\}\big) && \text{(by IH)} \\ &= \text{HASH}\big(c_j^{t-1}, \{c_m^{t-1} \mid m \in N(j)\}\big) && \text{(by 8 and IH)} \\ &= c_j^t\end{aligned}$$

As both graphs have the same node features $\boldsymbol{X}^1 = \boldsymbol{X}^2$ the set of colors is also the same in each iteration t of the WL-test. Hence, the graphs are indistinguishable by the WL-test □

Theorem 4.7. *FR-WL is strictly more powerful than NF-WL for fragmentation schemes $\mathcal{F}$ recovering 3-cycles.*

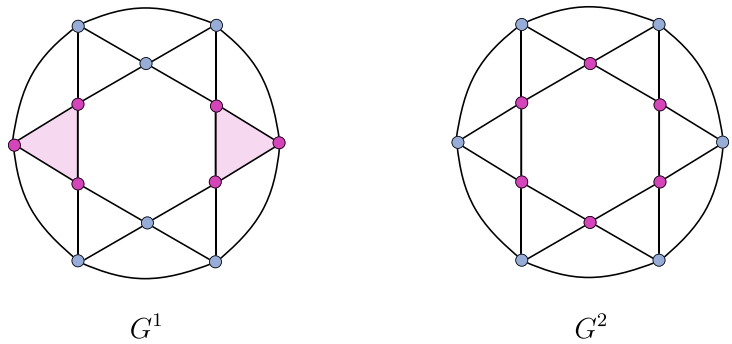

Figure 6. Graph G^1 and graph G^2 that are indistinguishable by NF-WL but distinguishable by FR-WL. Node features are represented by the color of the nodes.

Proof. With Lemma A.10 we can now prove Theorem 4.7: We first show that FR-WL is more powerful than NF-WL. Consider the two graphs G^1, G^2 depicted in Figure 6 with two different node features colored violet and blue. Note that when not considering the node features the graphs are identical and each node is isomorphic to every other node. As every fragmentation scheme has to be permutation invariant, every node is assigned the same additional node feature in NF (which, thus, holds no additional information to distinguish the graphs). Now, observe that the two graphs fulfill the conditions of Lemma A.10. Therefore, the graphs are indistinguishable by NF-WL. In contrast, FR-WL can distinguish the two graphs as G^1 contains 3-cycles with three violet nodes whereas G^2 does not contain such 3-cycles. Hence, the fragment representation for these 3-cycles differs and distinguishes the two graphs.

Now, it remains to show that NF-WL is not more powerful than FR-WL. Let c_v^t, d_v^t be the colorings of the t-th iteration of FR-WL and NF-WL, respectively. We will show by induction over t that $c_v^{t+1} \sqsubseteq d_v^t$. To simplify the notation, let $\mathcal{N}_\uparrow(v) := \{f \mid v \in f,\ f \in F\}$ be the set of fragments that v is part of. For $t = 0$, note that after the first iteration of the FR-WL test each vertex $v \in G$ receives information about the fragment it is part of

$$\begin{aligned}
c_v^1 &= \text{HASH}\big(c_v^0, \{\!\{c_n^0 \mid n \in \mathcal{N}_G(v) \uplus \mathcal{N}_\uparrow(v)\}\!\}\big) \\
&= \text{HASH}\big(c_v^0, \{\!\{c_n^0 \mid n \in \mathcal{N}(v)\}\!\} \uplus \{\!\{c_f^0 \mid f \in \mathcal{N}_\uparrow(v)\}\!\}\big) \\
&\sqsubseteq \big(c_v^0, \{\!\{c_f^0 \mid f \in \mathcal{N}_\uparrow(v)\}\!\}\big) \\
&= \big(\boldsymbol{X}_v, \{\!\{\text{type}(f) \mid f \in \mathcal{N}_\uparrow(v)\}\!\}\big) \\
&\sqsubseteq d_v^0
\end{aligned}$$

For the induction step $(t-1) \to t$, we have

$$\begin{aligned}
c_v^t &= \text{HASH}\big(c_v^{t-1}, \{\!\{c_n^{t-1} \mid n \in \mathcal{N}_G(v) \uplus \mathcal{N}_\uparrow(v)\}\!\}\big) \\
&\sqsubseteq \big(c_v^{t-1}, \{\!\{c_n^{t-1} \mid n \in \mathcal{N}_G(v)\}\!\}\big) \\
&\sqsubseteq \big(d_v^{t-2}, \{\!\{d_n^{t-2} \mid n \in \mathcal{N}_G(v)\}\!\}\big) \\
&\sqsubseteq d_v^{t-1}
\end{aligned}$$

This concludes the induction step. Hence, we have $c \sqsubseteq d$, and by Lemma A.8 NF-WL is not more powerful than FR-WL. □

Proof of Theorem 4.8

Theorem 4.8. *HLG-WL is strictly more powerful than FR-WL for fragmentation schemes $\mathcal{F}$ recovering 3-cycles.*

Proof. Consider the two graphs G^1, G^2 depicted in Figure 2 with two different node features colored green and red. Note that the two graphs fulfill the conditions of Lemma A.10. Furthermore, even the graphs FR(G^1) and FR(G^2) fulfill these conditions. Hence, the graphs G^1, G^2 cannot be distinguished by FR-WL.

In the higher level graph of G^1 each 3-cycle representation is connected to two other 3-cycle representations. Contrarily, in the higher level graph of G^2 each 3-cycle representation is connected to only one 3-cycle representations. Hence, the coloring will differ and HLG-WL distinguishes the two graphs.

Now, it remains to show that FR-WL is not more powerful than HLG-WL. This follows immediately from Lemma A.8 and the fact that the change in the graph from FR to HLG is reversible. □

Proof of Theorem 4.9

Theorem 4.9. *HLG-WL is in parts more powerful than 3-WL for fragmentation schemes $\mathcal{F}$ recovering 5-cycles.*

Proof. The proof follows a similar proof by Bodnar et al. (2022). Consider the Rook's 4×4 and Shrikhande graph (both in the family of strongly regular graphs $\mathrm{SR}(16, 6, 2, 2)$). The Shrikhande graph possesses 5-cycles while the Rook's graph does not. Hence, HLG-WL can trivially disitnguish those two graphs with a fragmentation recovering 5-cycles. However, it is known that 3-WL cannot distinguish those two graphs (Bodnar et al., 2022). □

A.3. Additional graph augmentation function

We will now consider two additional graph augmentation functions that are often used in practice: a learned representation for each edge and a learned representation for the complete graph that is connected to all other nodes. While these augmentations might be beneficial in practice, we will show that they do not increase expressiveness.

Edge representation We will first formally define the edge representation augmentation (ER). For every edge, we introduce a new node that is connected to its two endpoints.

Definition A.11. The edge representation graph augmentation function is $\mathrm{ER}(\mathcal{V}, \mathcal{E}, \boldsymbol{X}) = (\mathcal{V}^{\mathrm{ER}}, \mathcal{E}^{\mathrm{ER}}, \boldsymbol{X}^{\mathrm{ER}})$ with

$$\begin{aligned}
&\mathcal{V}^{\mathrm{ER}} := \mathcal{V} \cup \mathcal{E}, \\
&\mathcal{E}^{\mathrm{ER}} := \mathcal{E} \cup \big\{\{e, v\} \mid e = \{u, v\} \in \mathcal{E}\big\} \\
&\boldsymbol{X}_i^{\mathrm{ER}} := \begin{cases} \boldsymbol{X}_i & i \in \mathcal{V} \\ \alpha & i \in \mathcal{E} \end{cases}
\end{aligned}$$

where α is some new label.

Now, we can show that this augmentation does not increase expressiveness compared to the WL test.

Lemma A.12. *ER-WL is as powerful as WL.*

Proof. We will first show that ER-WL is not more powerful than WL: Let $c^{(i)}, d^{(i)}$ be the colorings of the WL test and of the ER-WL test, respectively. Then we will show that

We will use Lemma A.9 and give a function h that maps a coloring $c^{(i)}$ of the WL test without edge representation to a coloring $d^{(i)}$ of ER-WL: Note that with the color $c_v^{(i)}$ of a node v one can compute the colors $c_u^{(i-1)}$ of all neighboring nodes u. Hence, we can determine from $c^{(i)}$ the following multiset

$$\{\!\{(c_v^{(i)}, c_u^{(i-1)}) \mid e = (u, v) \in \mathcal{E}\}\!\}$$

which allows us to compute the corresponding edge representations $d_e^{(i)}$.

We will now show that ER-WL is not less powerful than WL. This follows directly from Lemma A.8 as ER is a reversible function. □

Graph representation We will now formally define the learned graph representation (GR), sometimes called virtual node.

Definition A.13. The graph rerpesentation augmentation function is $\mathrm{GR}(\mathcal{V}, \mathcal{E}, \boldsymbol{X}) = (\mathcal{V}^{\mathrm{GR}}, \mathcal{E}^{\mathrm{GR}}, \boldsymbol{X}^{\mathrm{GR}})$ with

$$\begin{aligned}
&\mathcal{V}^{\mathrm{GR}} := \mathcal{V} \cup \{g\}, \\
&\mathcal{E}^{\mathrm{GR}} := \mathcal{E} \cup \big\{\{v, g\} \mid v \in \mathcal{V}\big\} \\
&\boldsymbol{X}_i^{\mathrm{GR}} := \begin{cases} \boldsymbol{X}_i & i \in \mathcal{V} \\ \alpha & i = g \end{cases}
\end{aligned}$$

where α is some new label.

Table 7. Overview of the vocabulary and expressiveness of existing topological and fragment-biased models. The bounds for GSN-v, $\mathcal{O}$-GNN, and HIMP are tight, i.e. when using a sufficient number of layers and injective neighborhood aggregators, the models are as powerful as the corresponding Fragment-WL test.

Model	**Bounded by**	**Vocabulary**
GSN-v[4]	$\leq$ NF-WL	Cliques or Rings
$\mathcal{O}$-GNN	$\leq$ FR-WL	Rings
HIMP	$\leq$ HLG-WL	Rings
MPSN	$\leq$ HLG-WL	Simplicial complexes (in practice cliques)
CIN	$\leq$ HLG-WL	CW complexes (in practice rings & edges)
CIN++	$\leq$ HLG-WL	CW complexes (in practice rings & edges)

Similar to the edge representation, the graph representation does not increase expressiveness:

Lemma A.14. *GR-WL is as powerful as WL.*

Proof. We will first show that GR-WL is not more powerful than WL. We will use Lemma A.9 and give a function h that maps a coloring $c^{(i)}$ of the WL test without graph representation to a coloring $d^{(i)}$ of GR-WL: We will show this by induction over i. For $i = 0$, this follows immediately from the definition of GR. For the induction step, assume that there exists such a function from $c^{(t-1)}$ to $d^{(t-1)}$. Note that the graph representation $d_g^{(t)}$ is computed as:

$$\begin{aligned} d_g^{(t)} &= \text{HASH}\big(d_g^{(t-1)}, \{\!\{d_v^{(t-1)} \mid v \in \mathcal{N}_{G'}(g)\}\!\}\big) \\ &= \text{HASH}\big(d_g^{(t-1)}, \{\!\{d_v^{(t-1)} \mid v \in \mathcal{V}\}\!\}\big). \end{aligned}$$

which can be derived from $c^{(t-1)}$ by induction hypothesis. With this we can trivially compute $d_v^{(t)}$ from $c^{(t)}$ for all other nodes v, too.

Since GR is a reversible function, by Lemma A.8 GR-WL is not less powerful than WL. Hence, GR-WL is as powerful as WL. □

A.4. Expressiveness of existing models

We will use our Fragment-WL tests to compare the expressiveness of existing fragment-biased GNN models. Table 7 gives an overview of the vocabulary of existing fragment-biased and topological GNNs. Additionally, it shows the expressiveness in our Fragment-WL hierarchy.

A.4.1. GSN-v

GSN-v (Bouritsas et al., 2023) incorporate fragment information as an additional node feature. The additional node features consist of the counts of fragment types a node is part of. Their framework also differentiates between different (non-symmetric) positions inside the fragment (e.g., first node in path vs. second node in path) that correspond to different orbits. While their framework can use any fragmentation scheme, in all real-world experiments, they only use rings or cliques. Note that for rings and cliques, no different orbits exist, i.e., each node in the substructure has the same orbit. Hence, this information becomes irrelevant.

Theorem A.15. *GSN-v using rings and/or cliques as vocabulary is at most as powerful as NF-WL. Additionally, when using injective neighborhood aggregators and a sufficient number of layers, GNSs are as powerful as NF-WL with a fragmentation scheme based on rings and cliques.*

[4]We evaluate the GSN-v that is used in practice, i.e. with a vocabulary of cliques and rings. Note that the theory of the authors allows for potentially more expressive instantiations.

Proof. GSN-v appends the node features by the counts of substructures and the respective orbits each node is part of. After that, a standard GNN is applied to the graph.

Note that in a ring or a clique, each node has exactly the same orbit. So, for a vocabulary based on cliques and rings the appended information degenerates to solely the substructure counts. Further, note that this substructure count function is an injective function λ as defined in Definition 4.2. Hence, when using injective neighborhood aggregators and an MLP update function with a sufficiently large number of layers such that it can approximate the HASH function, GSN-v exactly models the NF-WL test. Hence, GSN-v is exactly as powerful as NF-WL. □

A.4.2. $\mathcal{O}$-GNNs

Besides representations for nodes, $\mathcal{O}$-GNNs (Zhu et al., 2022) use explicit representation for rings, edges, and the whole graph.

Theorem A.16. *$\mathcal{O}$-GNNs (Zhu et al., 2022) are at most as powerful as FR-WL. Additionally, when using injective neighborhood aggregators and a sufficient number of layers, $\mathcal{O}$-GNNs are as powerful as FR-WL with a fragmentation scheme based solely on rings.*

Proof. $\mathcal{O}$-GNNs (when using injective neighborhood aggregators instead of their original sum aggregators and an MLP with a sufficiently large number of layers such that it can approximate the HASH function) models performing the WL test on an FR, ER, and GR augmented graph. As shown in Lemmas A.12 and A.14, the edge representation and the graph representation do not influence the expressivity. Hence, $\mathcal{O}$-GNNs are exactly as powerful as FR-WL with a fragmentation scheme based solely on rings. □

A.4.3. HIMP

HIMP (Fey et al., 2020) builds a higher-level junction tree based on rings and edges for message passing on the original graph, the higher-level junction tree, and between those two.

Theorem A.17. *HIMP is at most as powerful as HLG-WL. Additionally, when using injective neighborhood aggregators and a sufficient number of layers, HIMP is as powerful as HLG-WL with a fragmentation scheme based on rings and edges.*

Proof. HIMP (when using injective neighborhood aggregators instead of their original sum aggregators and an MLP with a sufficiently large number of layers such that it can approximate the HASH function) exactly models performing the WL test on an HLG augmented graph. Hence, HIMP is exactly as powerful as FR-WL with a fragmentation scheme based on rings and edges. □

A.4.4. FragNet

Next, we consider the expressiveness of our FragNet model:

Theorem 5.1. *FragNets are at most as powerful as HLG-WL. Additionally, when using injective neighborhood aggregators and a sufficient number of layers, FragNets are as powerful as HLG-WL.*

Proof. For the proof, we rely on Lemma A.12, the finding that an explicit edge representation does not augment expressiveness. Notice that our model, when using injective neighborhood aggregators and an MLP with a sufficiently large number of layers such that it can approximate the HASH function, exactly models performing the WL test on an HLG and ER augmented graph. As shown in Lemma A.12, ER does not change the expressiveness. Hence, our model is exactly as powerful as HLG-WL. □

A.4.5. Topological GNNs

We will now consider topological GNNs. We will start by comparing HLG-WL with CWL, a variant of the WL test operating on CW complexes (Bodnar et al., 2022). In the CWL framework, every graph is (permutation invariantly) mapped to a set of cells $\mathcal{X}$, a CW complex (using a skeleton-preserving lifting map). Let $\mathcal{X}_i$ denote the set of cells with dimension i. Then, $\mathcal{X}_0$ corresponds to all vertices $\mathcal{V}$ and $\mathcal{X}_1$ to all edges $\mathcal{E}$. For higher dimensions, the results depend on the particular cellular lifting map. For instance, $\mathcal{X}_2$ could correspond to all cycles.

Theorem A.18. *HLG-WL is not less powerful than CWL, with a fragmentation scheme $\mathcal{F}$ that corresponds to the cellular lifting map used by CWL.*

Proof. Let F be the fragmentation that corresponds to $\mathcal{X}$ without the vertices: $F = \mathcal{X} \setminus \mathcal{V}$. Let $c^{(t)}, b^{(t)}$ be the coloring of iteration t of HLG-WL and CWL, respectively. We will show that $b^{(t)} \sqsubseteq c^{(2t)}$ which implies that $b \sqsubseteq c$ after termination.

We will show this by induction over t. For $t = 0$, this follows immediately from the fact that the node features in HLG-WL are finer than the features of cells in CWL.

Now we will show $c^{(t)} \sqsubseteq b^{(2t)}$ assuming $c^{(t-1)} \sqsubseteq b^{(2t-2)}$:

The idea of the induction step is that the hash update function HASH receives more information in HLG-WL compared to CWL. Let us first consider vertices, i.e., $\mathcal{X}_0$. The update function in CWL for $v \in \mathcal{V} = \mathcal{X}_0$ is:

$$b_v^{(t)} = \text{HASH}\big(b_v^{(t-1)}, \{\!\{(b_w^{(t-1)}, b_e^{(t-1)}) \mid e = \{v, w\} \in \mathcal{E}\}\!\}\big)$$

Now note that the update function in HLG-WL for $v \in \mathcal{V}$ is:

$$\begin{aligned}
c_v^{(2t)} &= \text{HASH}\big(c_v^{(2t-1)}, \{\!\{c_w^{(2t-1)} \mid w \in \mathcal{N}_{\text{HLG}(G)}(v)\}\!\}\big) \\
&\sqsubseteq \text{HASH}\big(c_v^{(2t-1)}, \{\!\{c_e^{(2t-1)} \mid e = \{v, w\} \in \mathcal{E}\}\!\}\big) \\
&\sqsubseteq \text{HASH}\big(c_v^{(2t-1)}, \{\!\{(c_e^{(2t-2)}, c_w^{(2t-2)}) \mid e = \{v, w\} \in \mathcal{E}\}\!\}\big) \\
&\sqsubseteq \text{HASH}\big(b_v^{(t-1)}, \{\!\{(b_e^{(t-1)}, b_w^{(t-1)}) \mid e = \{v, w\} \in \mathcal{E}\}\!\}\big) \\
&= b_v^{(t)}
\end{aligned}$$

The first step follows from $e \in \mathcal{N}_{\text{HLG}(G)}(v)$ as the edges are part of the fragmentation F. The second step follows from $c_e^{(2t-1)} \sqsubseteq (c_e^{(2t-2)}, c_w^{(2t-2)}, c_v^{(2t-2}$. The third step uses the induction hypothesis.

Now, we will consider a cell $x \in \mathcal{X}_k \subseteq F$. The update function in CWL is

$$b_x^{(t)} = \text{HASH}\big(b_x^{(t-1)}, \{\!\{(b_u^{(t-1)}, b_o^{(t-1)}) \mid x \prec u,\ o \prec u\}\!\}, \{\!\{b_l^{(t-1)} \mid l \prec x\}\!\}\big)$$

where $x \prec y$ means that x with dimension k is part of the cell y of dimension $k+1$. For example, $e \prec r$ if e is an edge in a ring $r \in X_{k+1}$. For details, we refer to Bodnar et al. (2021).

The update function in HLG-WL for a fragment $x \in F \cap \mathcal{X}_k$ in $G' := \text{HLG}(G)$ is:

$$\begin{aligned}
c_x^{(2t)} &= \text{HASH}\big(c_x^{(2t-1)}, \{\!\{c_w^{(2t-1)} \mid w \in \mathcal{N}_{G'}(x)\}\!\}\big) \\
&\sqsubseteq \text{HASH}\big(c_x^{(2t-1)}, \{\!\{c_u^{(2t-1)} \mid u \in \mathcal{N}_{G'}(x) \cap \mathcal{X}_{k+1}\}\!\}, \{\!\{c_l^{(2t-1)} \mid l \in \mathcal{N}_{G'}(x) \cap \mathcal{X}_{k-1}\}\!\}\big) \\
&\sqsubseteq \text{HASH}\big(c_x^{(2t-1)}, \{\!\{c_u^{(2t-1)} \mid x \prec u\}\!\}, \{\!\{c_l^{(2t-1)} \mid l \prec x\}\!\}\big) \\
&\sqsubseteq \text{HASH}\big(c_x^{(2t-1)}, \{\!\{(c_u^{(2t-2)}, c_o^{(2t-2)}) \mid x \prec u,\ o \prec u\}\!\}, \{\!\{c_l^{(2t-1)} \mid l \prec x\}\!\}\big) \\
&\sqsubseteq \text{HASH}\big(b_x^{(t-1)}, \{\!\{(b_u^{(t-1)}, b_o^{(t-1)}) \mid x \prec u,\ o \prec u\}\!\}, \{\!\{b_l^{(t-1)} \mid l \prec x\}\!\}\big) \\
&= b_x^{(t)}
\end{aligned}$$

The steps are very similar to the vertex case above. This concludes the proof that c refines b. By Lemma A.5 this implies that HLG-WL is not less powerful than CWL using a fragmentation that corresponds to the cellular complex. □

As CWL bounds the expressiveness of CIN (Bodnar et al., 2022) and CIN++ (Giusti et al., 2023), we get the following corollary:

Corollary A.19. *CIN and CIN++ are at most as powerful as HLG-WL with a fragmentation scheme that corresponds to the cellular lifting map.*

Additionally, CWL subsumes the WL version, SWL, introduced by Bodnar et al. (2021) for simplicial complexes. The cellular complex just corresponds to all cliques of the graph.

Corollary A.20. *HLG-WL, with a fragmentation scheme recovering cliques, is not less powerful than SWL.*

As MPSNs (Bodnar et al., 2021) are bounded by SWL, we have the following result for MPSNs:

Corollary A.21. *MPSNs are at most as powerful as HLG-WL with a fragmentation scheme based on cliques.*

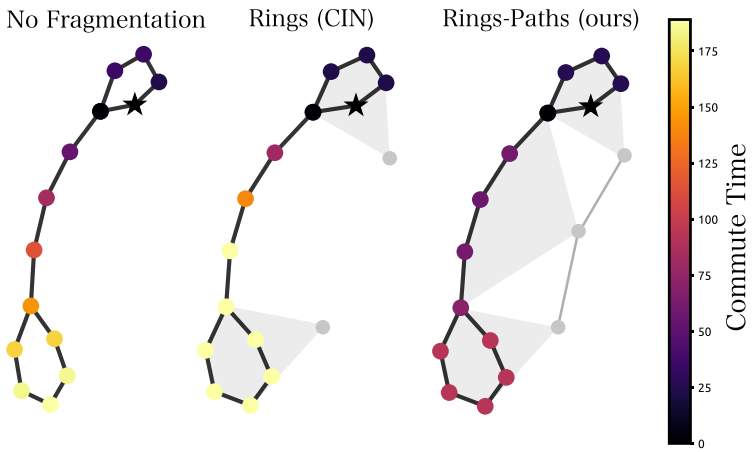

Figure 7. Commute time from the star node to all other nodes. The first graph has no fragmentation, the second one a rings fragmentation (like in CIN/CIN++), the third a rings and paths fragmentation (like our model).

B. Further experiments

B.1. Long-Range tests

We provide more experiments to measure the long-range capabilities of FragNet.

Commute time

In addition to the recovery rates in Figure 5, we also consider commute times. The commute time between the nodes a and b is the expected time for a random walker from a to reach b and return again to a. Di Giovanni et al. (2023) have proposed the commute time as a measure for over-squashing. To compute and compare commute times across different fragmentations, we connected all nodes in each fragmentation that could exchange a message within one layer. Figure 7 shows the commute from the star node to every other node for the same graph as in Figure 5. The close alignment between commute time and recovery rate supports the theoretical findings by Di Giovanni et al. (2023) and further emphasizes the potentially enhanced long-range capabilities of our model. Additionally, we also compute commute times on a molecule from the ZINC dataset that contains more fragments (see Figure 8).

Quantitatively, we compute average commute times on a random sample of molecules from the peptides dataset for a model without any fragmentation and for FragNet (RingsPaths fragmentation with a higher-level graph). We observe that the addition of a higher-level graph reduces commute times by 16%.

Table 8. Average commute times between all nodes on a random sample of 50 molecular graphs from the peptides dataset with and without the higher-level graph.

Normal Molecular Graph	5056
Molecular Graph + HLG	4253

B.2. Distribution of Fragments

Figure 9 illustrates the distribution of fragment sizes, i.e., path lengths and ring lengths, extracted by our RingsPaths fragmentation method across the ZINC-10k, ZINC-full, and peptides datasets. It is worth noting that the peptides dataset features some exceptionally large rings.

B.3. Ablation Studies

In the following, we test the design choices of our model and fragmentation. First, we test FragNet without the different fragment information or ordinal encoding on ZINC and Peptides. We show the results in Table 9. We observe that a reduction

in expressiveness generally leads to a reduction in performance. An exception is the use of fragment representations (FR-WL, i.e. FragNet - Higher-level graph) which shows a higher error on ZINC 10k and Peptides Struct. This is similar to the pattern that we observe in the message reconstruction toy experiment we show in Figure 5 where the additional fragment representations increase the importance of the current substructure and do not contain a message from other parts of the molecule.

Table 9. Ablation of different expressivity choices for FragNet. Additionally, we ablate the ordinal encoding.

Model	**ZINC**	**Peptides**	
	10k (MAE ↓)	**Struct** (MAE ↓)	**Func** (AP ↑)
FragNet (=HLG-WL)	**0.0775** ± 0.004	**0.246** ± 0.002	**0.668** ± 0.003
− Higher-level graph (=FR-WL)	0.0872 ± 0.004	0.256 ± 0.003	0.661 ± 0.005
− Fragment representation (=NF-WL)	0.0994 ± 0.007	0.247 ± 0.003	0.654 ± 0.005
− All fragment information (=WL)	0.1609 ± 0.003	0.249 ± 0.001	0.652 ± 0.005
FragNet − ordinal encoding	0.0945 ± 0.006	0.249 ± 0.001	0.666 ± 0.004

Furthermore, we compare different fragmentation schemes in combination with FragNet. In Table 10, we observe that our RingsPath fragmentation scheme performs the best across the different datasets.

Table 10. Performance of FragNet with different fragmentation schemes.

Fragmentation Scheme	**ZINC**	**Peptides**	
	10k (MAE ↓)	**Struct** (MAE ↓)	**Func** (AP ↑)
BBB	0.127	0.252 ± 0.002	0.637 ± 0.003
BRICS	0.127	0.247 ± 0.008	0.658 ± 0.011
Magnet	0.098	-	-
Rings	0.078	0.249 ± 0.001	0.659 ± 0.007
RingsPaths (ours)	**0.077**	**0.246** ± 0.002	**0.668** ± 0.005

In Figure 10, we look at how large the vocabulary size has to be per fraction of fragmented atoms. That is, for an increasing vocabulary, we observce how many atoms belong to a fragment. The steeper the increase the better. We can observe that on ZINC-10k BBB, BRICS and Rings are not able to assign a fragment to each atom no matter how large the vocabulary size. Magnet achieves full fragmentation but slower compared to our RingsPaths which is the most vocabulary efficient. We further show the necessary vocabulary size on ZINC-Full and Peptides for RingsPaths in Table 11.

Table 11. Vocabulary sizes for RingsPaths on different datasets.

	ZINC-10k	**ZINC-Full**	**Peptides**
Vocabulary Size	18	28	100

C. Experimental details

In the following, we will describe details for all our experiments. Unless otherwise stated, for our FragNet, we use a 2-layer fully connected neural network with ReLU activations and batch norm as the MLP update function. For the aggregation method AGG, we use a sum aggregation for messages within the original or higher-level graph and a mean aggregation for messages between the original graph and the higher-level graph. For training, we use the AdamW (Loshchilov & Hutter, 2017) optimizer and gradient clipping with a value of 1. The model has been implemented in PyTorch (Paszke et al., 2019) using the PyTorchGeometric (Fey & Lenssen, 2019) and the PyTorch Lightning (Falcon & The PyTorch Lightning team, 2019) library. It is in parts adapted from HIMP (Fey et al., 2020). All results for other methods are taken from Rampášek et al. (2023) and Giusti et al. (2023).

	peptides-struct	peptides-func	ZINC-10k	ZINC-full
num_layers	3	2	5	3
hidden_channels	110	128	64	120
num_layers_out	3	3	3	2
frag-reduction	sum	sum	max	max
out-reduction	mean	mean	mean	mean
dropout	0.05	0.15	0	0
lr	0.001	0.001	0.001	0.001
weight decay	0	0	0.001	0
ema decay	0.99	0.99	0.99	0.99
scheduler	Cosine	ReduceonPlateau	Cosine	ReduceonPlateau
patience	-	30	-	15
factor	-	0.5	-	0.9
batchsize	32	128	32	128
max epochs	300	400	2000	1000
num parameters	440K	440K	221K	494K

Table 12. Hyperparameter configuration of our model for the ZINC and peptides benchmarks

Table 13. Fragment counts for the 42 most common MagNet fragments in ZINC and accuracy scores of our model in predicting the counts.

Fragment														
Count	12862	7548	6198	5629	3904	2204	1799	1772	1348	1330	1071	741	573	375
Accuracy	1.0	0.997	0.999	0.986	0.99	0.969	0.999	1.0	0.963	0.997	0.997	0.933	0.999	0.954
Fragment														
Count	208	204	176	156	113	113	90	80	77	66	54	45	37	32
Accuracy	0.999	0.988	0.983	0.982	0.996	0.995	0.993	0.991	0.998	0.995	0.996	0.996	0.996	0.998
Fragment														
Count	32	31	28	25	23	19	19	18	18	17	15	15	15	13
Accuracy	0.998	0.998	0.996	0.997	1.0	0.998	0.997	0.998	1.0	1.0	0.998	0.998	1.0	0.999

C.1. ZINC and peptides

The hyperparameters of our model for ZINC (10k and full) and peptides (struct and func) can be found in 12. Note that we adhere to the 500K parameter budget. Each experiment is repeated over three different seeds except for the ZINC-full experiment, where we only have a single run because of computational and time limitations.

C.2. Expressiveness

We use the MagNet (Hetzel et al., 2023) fragmentation to fragment all graphs in the ZINC-subset dataset. We sort the fragments by number of occurrences in the training set. For each of the 28 most common substructure we train our model to predict the counts of these substructures. As model parameters we use three layers of message passing with a hidden dimension of 120. For the final readout function we use a sum aggregation and a two layer MLP. We train our model using the MAE loss for 200 epochs with a learning rate of 0.001 and a reduce-on-plateau learning rate scheduling. We report the accuracy (percentage of graphs where rounded prediction equals the ground-truth count) on the test set. Table 13 shows the complete table of all substructures.

C.3. Long-range Interaction: Recovery rate

In our synthetic long-range experiment, we consider a graph consisting of two rings connected by a path Figure 5. One node in the graph is the designated source node (marked by a star). The feature of the source node is initialized with one-hot-encoding of one of 10 different classes. All other node features are initialized with a constant encoding. For every node t in the graph, we train a separate model to predict the class of the source node s, i.e. the target node t has to reconstruct a message from the source node. The number of layers of the models is $\max(d(s,t),3)$, ensuring that the target can receive messages from the source. We train the model with the cross entropy loss between the prediction at the target node and the true class of the source. We compare the results of models that have no fragmentation, a ring fragmentation and a ring-path fragmentation. We use a our model without batchnorm and a hidden dimension of 64. We train the model for a maximum of 200 epochs with a starting learn rate of 0.001 and average the results over at least five seeds.

C.4. Generalization: Rarity

For the experiment in Table 5, we report the MAE of the ZINC-full validation set grouped by the frequency of the rarest fragment in the molecule. The frequency of a fragment is defined as the fraction of molecules that contain the fragment. As the fragmentation scheme, we use the simple Rings fragmentation.

C.5. Generalization: QM9

To perform our generalization experiment on QM9, we transform the edge and node features of the molecular graphs in QM9 so that they have the same node features and edge features as the graphs in the ZINC dataset. Additionally, we do not use any molecular graphs that contain atom types that do not appear in the ZINC dataset. We calculate penalized logP as ground truth. Then, we trained our model and GRIT on ZINC-full and tested them on the transformed QM9 dataset.

D. Downstream tasks using substructures

Many other molecular tasks beyond property prediction can benefit from substructure information, highlighting the broader potential applications of our model.

Motifs for Drug Discovery Motifs and specific substructures are important inductive biases in molecular generation, optimization, and scaffolding tasks (Hu et al., 2023; Sommer et al., 2023; Du et al., 2022). Employing a set of fragments can simplify the generation process and increase the chemical validity of the generated molecules. Given a fragmentation procedure, the fragments are aggregated into a vocabulary of motifs through complete enumeration of the dataset (Jin et al., 2019; 2020; Geng et al., 2023), top-k selection (Kong et al., 2022; Maziarz et al., 2022) or consolidation into Murcko scaffolds (Hetzel et al., 2023). Encoders for molecule generation often integrate motifs as node features or via additional higher-level encoder networks, as the decoder is explicitly tasked with reconstructing the set of motifs from a given embedding.

Pretaining In the context of using GNNs for drug discovery, incorporating motifs as part of a pretraining phase has been shown to improve representation learning capabilities. Zang et al. (2023) integrates higher-level structures as nodes in a graph and leverages the graph's hierarchy for self-supervised pretraining. Similarly, Zhang et al. (2021) propose a GNN that operates on a two-tiered graph and predicts the sequence of motifs during network pretraining. To improve the encoding of higher-level structures, Inae et al. (2023) suggest a motif-aware pretraining technique, which masks entire motifs during the pretraining phase.

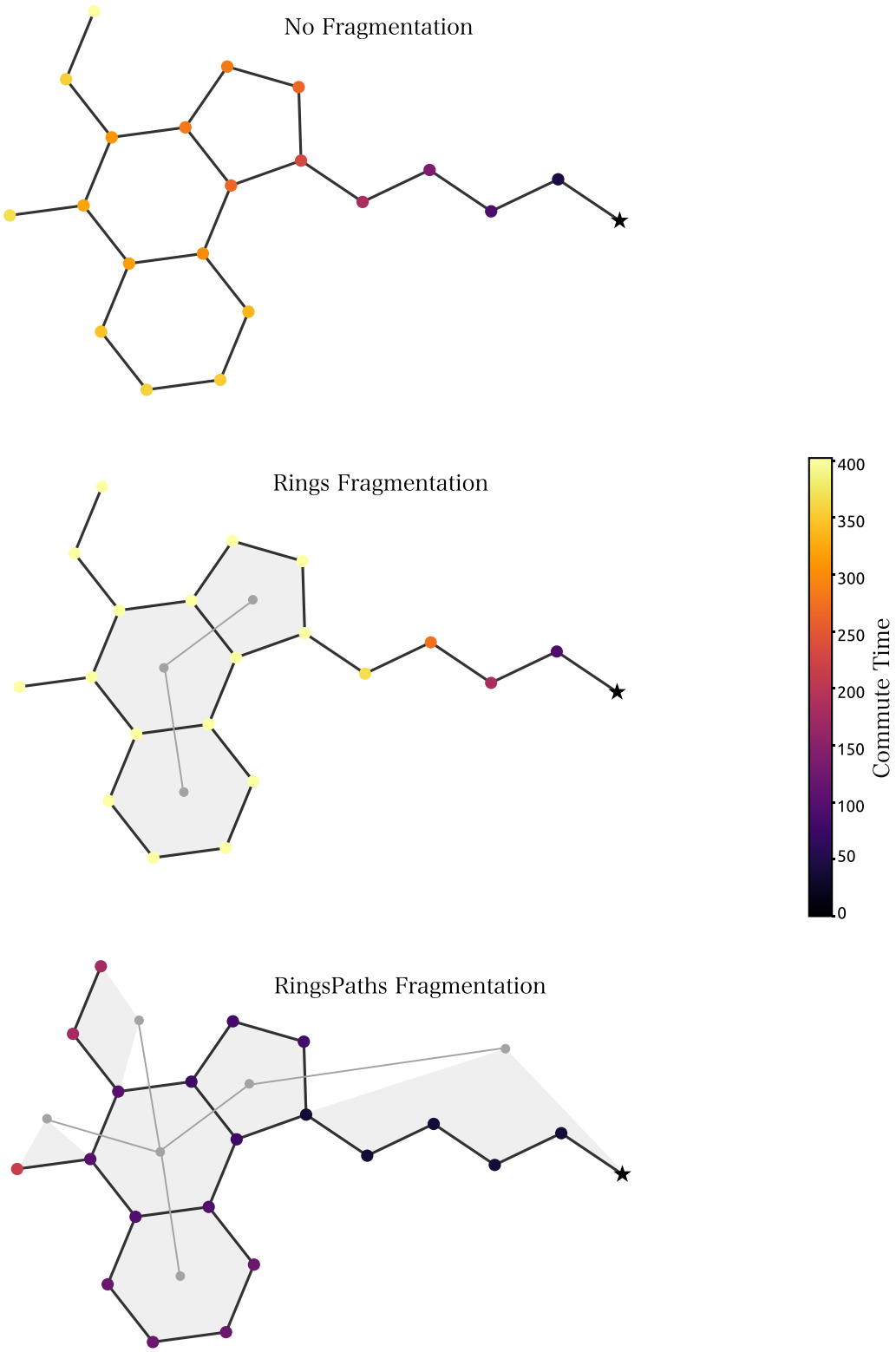

Figure 8. Commute time from the star node to all other nodes. The first graph has no fragmentation, the second one a rings fragmentation (like in CIN/CIN++), the third a rings and paths fragmentation (like our model).

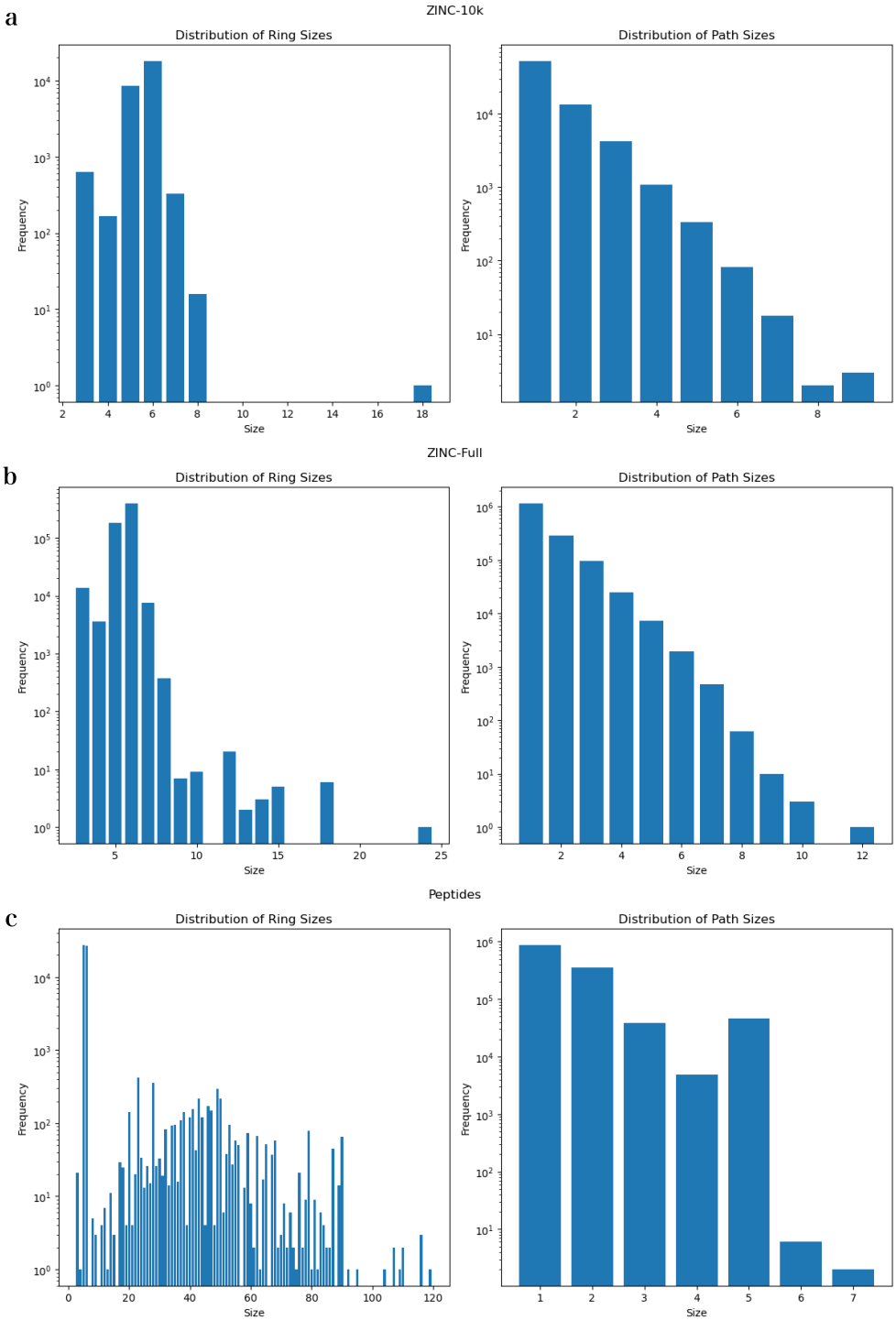

Figure 9. Distribution of sizes of path and ring fragments for the a) ZINC-10k, b) ZINC-full, and c) Peptides dataset.

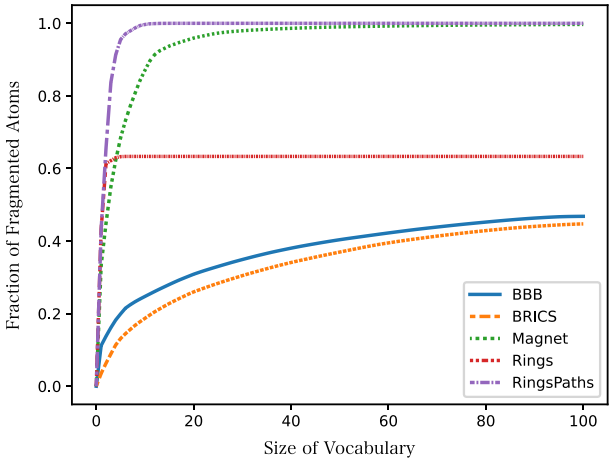

Figure 10. Fraction of atoms in ZINC-10k dataset that are part of a fragment as a function of vocabulary size. A fraction of 1 indicates that all molecules in the dataset can be completely fragmented. We compare the chemically inspired fragmentation schemes BBB, BRICS, and MagNet with a fragmentation based just on rings and our RingsPaths fragmentation. The substructures in the vocabulary are sorted by the frequency in which they appear in the molecules.