Maximally Expressive GNNs for Outerplanar Graphs

Franka Bause*
Faculty of Computer Science
UniVie Doctoral School Computer Science
University of Vienna, Vienna, Austria
franka.bause@univie.ac.at

Fabian Jogl*
Center for Artificial Intelligence and Machine Learning
Machine Learning Research Unit
TU Wien, Vienna, Austria
fabian.jogl@tuwien.ac.at

Pascal Welke
Machine Learning Research Unit
TU Wien, Vienna, Austria
pascal.welke@tuwien.ac.at

Maximilian Thiessen
Machine Learning Research Unit
TU Wien, Vienna, Austria
maximilian.thiessen@tuwien.ac.at

Abstract

Most pharmaceutical molecules can be represented as *outerplanar* graphs. We propose a graph transformation that makes the Weisfeiler-Leman (WL) test and message passing graph neural networks maximally expressive on outerplanar graphs. While existing research predominantly focuses on enhancing expressivity of graph neural networks beyond the WL test on arbitrary graphs, our goal is to distinguish pharmaceutical graphs specifically. Our approach applies a linear time transformation, building on the fact that biconnected outerplanar graphs can be uniquely identified by their Hamiltonian adjacency list sequences. This pre-processing step can then be followed by any graph neural network. We achieve promising results on molecular benchmark datasets while keeping the pre-processing time low, in the order of seconds for common benchmarks.

1 Introduction

We study graph neural networks (GNNs) for the family of outerplanar graphs and devise a model that can distinguish all non-isomorphic outerplanar graphs. Most previous work relates the expressivity of GNNs to some k-dimensional Weisfeiler-Leman isomorphism test (k-WL), most famously showing that some GNNs can distinguish exactly all those graphs that k-WL can distinguish [Maron et al., 2019b; Morris et al., 2020; Xu et al., 2019]. In contrast, here we present an expressivity result for a well established class of graphs not defined by (generalized) message passing. We have chosen outerplanar graphs, since most pharmaceutical compounds are outerplanar [Droschinsky et al., 2017; Horváth and Ramon, 2010; Horváth et al., 2006]. In fact, most well-known benchmark datasets for graph-level tasks contain over 92% outerplanar graphs (see Table 2 in the appendix).

Relying on an isomorphism test for biconnected outerplanar graphs by Colbourn and Booth [1981], we develop a linear-time graph transformation [Jogl et al., 2023; Veličković, 2022]. This enables 1-WL, and thus message passing neural networks (MPNNs), to distinguish such graphs. We rely on *Hamiltonian adjacency lists* (HALs), which encode the structure of each biconnected component, and the fact that 1-WL can distinguish labeled trees. We discuss related work in Appendix D.

2 Preliminaries

A *graph* $G = (V, E, \mu, \nu)$ consists of a set of nodes V, a set of edges $E \subseteq V \times V$ and $\mu \colon V \to X$ and $\nu \colon E \to X$ arbitrary attributes for the nodes and edges, respectively. We refer to an edge from u to v by uv, and in case of undirected graphs $uv = vu$. The *in-neighbors* of a node $u \in V$ are denoted by $N_i(u) = \{v \mid vu \in E\}$. The *out-neighbors* of a node $u \in V$ are denoted by $N_o(u) = \{v \mid uv \in E\}$

*Equal contribution. For an extended version, see [Bause et al., 2023].

F. Bause et al., Maximally Expressive GNNs for Outerplanar Graphs (Extended Abstract). Presented at the Second Learning on Graphs Conference (LoG 2023), Virtual Event, November 27–30, 2023.

and in case of undirected graphs, $N_i = N_o$. We focus on undirected input graphs, and will transform them into directed ones. A (directed) cycle $(v_1, \dots, v_k)$ is a sequence of k distinct nodes, with $\forall i \in \{1, \dots, k-1\} : v_i v_{i+1} \in E$ and $v_n v_1 \in E$. Given a graph G, we denote the shortest path distance between v_i and v_j by $d_G(v_i, v_j)$. A graph is *outerplanar* if it can be drawn in the plane without edge crossings and with all nodes belonging to the exterior face (for more details, see, e.g., Felsner [2012]). We call an undirected graph with at least three vertices *biconnected* if the removal of any single node does not disconnect the graph. A *biconnected component* is a maximal biconnected subgraph and we call biconnected outerplanar components of a graph *blocks*. Two graphs G and H are isomorphic, if there exists a bijection $\phi \colon V(G) \to V(H)$, so that $\forall u, v \in V(G) \colon \mu(v) = \mu(\phi(v)) \wedge uv \in E(G) \Leftrightarrow \phi(u)\phi(v) \in E(H) \wedge \forall uv \in E(G) \colon \nu(uv) = \nu(\phi(u)\phi(v))$. An *in-tree* T is a connected, directed, acyclic graph with a distinct *root* with no outgoing edges and other nodes have one outgoing edge.

The 1-dimensional Weisfeiler-Leman algorithm (WL) iteratively assigns colors to nodes, starting with colors representing node labels (or a uniform coloring for unlabeled nodes). The color of $v \in V(G)$ is updated each iteration according to $c_{i+1}(v) = h\left(c_i(v), \{\!\{(\nu(uv), c_i(u)) \mid u \in N_i(v)\}\!\}\right)$, where h is an injective function and $c_0 = \mu$. The *unfolding tree* with height i of a node $v \in V(G)$ is defined as $F_i^v = (v, \{\!\{F_{i-1}^u \mid u \in N_i(v)\}\!\})$, where $F_0^v = (\{v\}, \emptyset)$. The unfolding trees F_i^v and F_i^w of two nodes v and w are isomorphic iff the colors of the nodes in iteration i are the same. WL has historically been used as a heuristic for graph isomorphism. Let $\mathrm{WL}(G) = \{\!\{c_\infty(v) \mid v \in V(G)\}\!\}$ be the multiset of node colors in the stable partitioning [Arvind et al., 2015]. Two graphs G and H are not isomorphic, if $\mathrm{WL}(G) \neq \mathrm{WL}(H)$. However, if $\mathrm{WL}(G) = \mathrm{WL}(H)$, G and H might not be isomorphic (WL for example cannot distinguish a 6-cycle from two triangles).

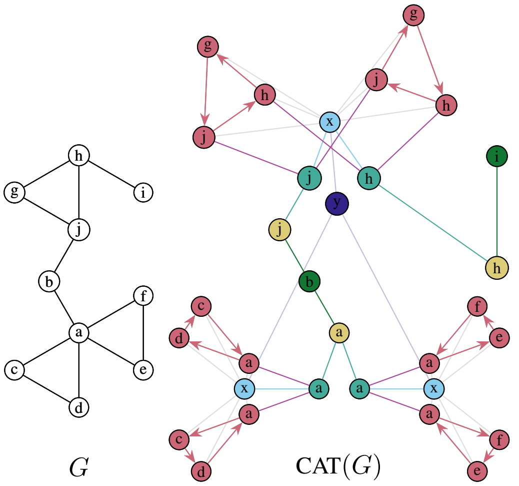

Figure 1: A graph and its CAT transformation. Original node labels are represented by letters, edge and node labels from the CAT transform are represented by colors. CAT(G) looks like a cat.

A Hamiltonian cycle is a cycle containing each node of the graph exactly once. Biconnected outerplanar graphs have a unique Hamiltonian cycle, that can be found in linear time [Mitchell, 1979]. Annotating each node with the sorted distances d_C to all its neighbors on the two directed variants of the Hamiltonian cycle C gives us Hamiltonian adjacency lists (HALs). Figure 3 in the Appendix shows two graphs annotated with their HALs in both directions of the Hamiltonian cycle. Following the Hamiltonian cycle in one direction and concatenating the HALs gives a sequence S (and a reverse sequence R, for the other direction). This HAL sequence uniquely identifies biconnected outerplanar graphs (if both directions and cyclic shift are considered):

Lemma 1 (Colbourn and Booth [1981])**.** *Two biconnected outerplanar graphs G and H with HAL and reverse sequences S_G, S_H and R_G, R_H are isomorphic, iff S_G is a cyclic shift of S_H or R_H.*

3 Identifying Outerplanar Graphs Using Weisfeiler-Leman

We develop a graph transformation called cyclic adjacency transform (CAT), that enables WL to distinguish all outerplanar graphs. We start by first introducing CAT*, enabling WL to distinguish any two biconnected outerplanar graphs, and then extend it in CAT to all outerplanar graphs.

Identifying biconnected outerplanar graphs using Weisfeiler-Leman. We first present a graph transformation called CAT*, that allows the Weisfeiler-Leman algorithm to distinguish any two *biconnected* outerplanar graphs. Figure 2 shows an example of CAT*.

Definition 1. *The CAT* transformation takes a biconnected outerplanar graph $G = (V, E, \mu, \nu)$ and yields a modified graph $G' = \text{CAT}^*(G) = (V', E', \mu', \nu')$ by performing the steps below.*

1. *Let $C = (v_1, \dots, v_n)$ be a (directed) Hamiltonian cycle of G and $\overleftarrow{C}$ be its reverse.*

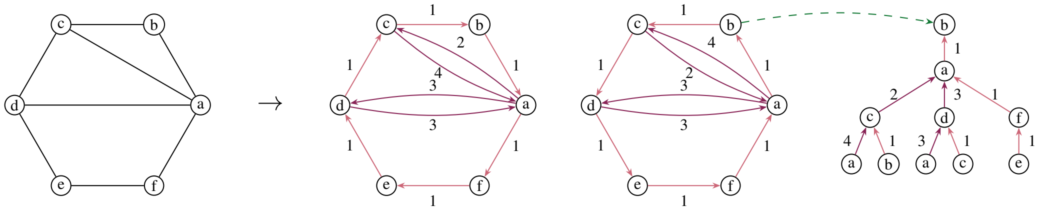

Figure 2: Biconnected outerplanar graph G, CAT$^*(G)$ and the unfolding tree of one of its nodes.

2. *Add node disjoint copies of C and $\overleftarrow{C}$ to G' and set $\nu'(e) = (1, \nu(e))$ for all edges in G'.*

3. *Let $D \subseteq E$ be the edges of G* not *on the (undirected) Hamiltonian cycle. Add edges in both directions to G' for the copies of C and $\overleftarrow{C}$ for each edge in D: $E(G') = E(G') \cup E_d \cup E_{\overleftarrow{d}}$ with $E_d = \bigcup_{\{v_i,v_j\}\in D}\{(v_i', v_j'), (v_j', v_i')\}$ and $E_{\overleftarrow{d}} = \bigcup_{\{v_i,v_j\}\in D}\{(v_i'', v_j''), (v_j'', v_i'')\}$ for copies v_i' of v_i in C (resp. v_i'' in $\overleftarrow{C}$). Set $\mu'(v_i') = \mu(v_i)$ and $\mu'(v_i'') = \mu(v_i)$ for the nodes in G'.*

4. *$\forall (v_i, v_j) \in E_d$ set $\nu'(v_i, v_j) = (d_C(v_j, v_i), \nu(v_i, v_j))$ and $\forall (v_i', v_j') \in E_{\overleftarrow{d}}$ set $\nu'(v_i', v_j') = (d_{\overleftarrow{C}}(v_j', v_i'), \nu(v_j, v_i))$.*

Theorem 1. *Two biconnected outerplanar graphs G and H are isomorphic, if and only if* $\mathrm{WL}($*CAT*$^*(G)) = \mathrm{WL}($*CAT*$^*(H))$.

Proof. Two graphs are distinguished by WL iff the multisets of node colors of their stable colorings differ. Trivially, $|V(G)| \neq |V(H)| \Rightarrow |V(\text{CAT}^*(G))| \neq |V(\text{CAT}^*(H))| \Rightarrow \mathrm{WL}(\text{CAT}^*(G)) \neq \mathrm{WL}(\text{CAT}^*(H))$, so we only focus on graphs with $|V(G)| = |V(H)|$. Two nodes only get the same color, if their unfolding trees are isomorphic. The first number in the HAL of each node is always 1, so it can be ignored, and the last number is always $|V(G)| - 1$, so this can simply be reconstructed by $|V(CAT^*(G))|$. The rest of the HAL sequence and the node labels of G can be reconstructed from the unfolding tree of any node in CAT$^*(G)$: Of course, each node has two direct neighbors in the Hamiltonian cycle. In the unfolding tree these are the parent and the single child with the 1-annotated edge. All other neighbors in the HAL can be reconstructed by looking at the weights of the edges that do not have weight 1. Figure 4 shows an example. Looking at any two biconnected outerplanar graphs with n nodes, Weisfeiler-Leman will be able to distinguish them after at most n iterations, iff they are non-isomorphic: Since the HAL sequence is encoded in the unfolding trees from all starting points (cyclic shift) and, because of the reverse copy, in both directions (reverse direction), this identifies isomorphism by Lemma 1. □

The CAT transformation. We define the CAT transformation by applying CAT* to the blocks of the graphs and adding nodes and edges to make outerplanar graphs distinguishable by WL.

Definition 2. *The CAT$(G) = G'$ transformation, maps a graph G to a new graph G' as follows:*

1. *Let $B_1, \ldots, B_\ell$ be the blocks of G and let F be the graph induced by the edges of G that are not in any block plus the nodes that are present in more than one block. Let $\{\perp, \square, \bowtie, \star, \triangle\}$ be distinct node labels not in X.*

2. *Add $\underline{F}$ to G' with labels $\underline{\mu'(v) = (\perp, \mu(v))}$ for $v \in F$.*

3. *For each block B_i in G:*

   3.1. *Add $\underline{(B_i', \overleftarrow{B_i'}) = \text{CAT}^*(B_i)}$ to G' (with labels).*

   3.2. *Let $A_i = V(B_i) \cap V(F)$ be the nodes of B_i in F.*

   3.3. *Let $\gamma_i : A_i \to V(B_i')$ map nodes of F to their copy in B_i' and $\overleftarrow{\gamma_i}$ to the copy in $\overleftarrow{B_i'}$.*

   3.4. *Add a node $\underline{b_i \text{ and (undirected) edges } \{b_i, v\}}$ for all $v \in V(B_i) \cup V(\overleftarrow{B_i'})$ to G'.*

   3.5. *For each $\underline{a \in A_i}$ add a node $\underline{p \text{ and edges } \{p, b_i\}, \{p, a\}, \{p, \gamma_i(a)\}, \{p, \overleftarrow{\gamma_i}(a)\}}$ to G'.*

   3.6. *Let $\underline{\mu'(b_i) = \square}$, and for each $a \in A_i$ let $\underline{\mu'(a) = (\bowtie, \mu(a))}$ and $\underline{\mu'(p) = (\star, \mu(a))}$ for the corresponding p.*

4. *Add a node g with $\mu'(g) = \triangle$ to G' and for all nodes b_i, add an edge $\{g, b_i\}$ to G'.*

5. *Let* CAT$(G) = G'$.

An example for a graph G and its corresponding graph $G' = \text{CAT}(G)$ can be seen in Figure 1.

Theorem 2. *Two outerplanar graphs G and H are isomorphic, iff* $\text{WL}(\textit{CAT}(G)) = \text{WL}(\textit{CAT}(H))$.

Proof. Following Theorem 1, each block will be uniquely identified by WL. Since the additional nodes have distinct labels, they will not cause WL to falsely report two blocks as isomorphic when they are not. The information about the entire HAL sequence of each block is stored in the b nodes after some iteration. The p nodes connect the block and b nodes to the rest of the graph, determining the orientation of the block. Note that the graph returned by CAT without the CAT* blocks and the node g is a tree. Relying on the labels of the p and b nodes, we can reconstruct the original graph from this tree. As WL can distinguish labeled trees [Arvind et al., 2015; Kiefer, 2020], it can thus distinguish non-isomorphic outerplanar graphs using CAT. For the other direction, note that CAT is permutation-invariant: for two isomorphic graphs G and H, the graphs CAT(G) and CAT(H) are isomorphic and WL will give the same output for both. □

Importantly, we can compute CAT(G) in linear time. The computational complexity is dominated by the computation of the blocks [Tarjan, 1972] and their Hamiltonian cycles [Mitchell, 1979], which both require linear time. Note that we only add a linear number of nodes and edges. From Xu et al. [2019] it follows, that MPNNs that are as expressive as 1-WL can distinguish CAT(G) and CAT(H) for non-isomorphic outerplanar graphs G and H. Thus, we propose to transform the input graphs using CAT and then use any MPNN on them. We can also apply CAT to non-outerplanar graphs. In this case, the biconnected outerplanar components are identified and the steps described in Definition 2 are performed. The remaining graph will simply be copied without modification. This never reduces expressivity but also is not guaranteed to distinguish general non-outerplanar graphs.

4 Experimental Evaluation

We investigate whether our proposed CAT can improve the predictive performance of the MPNN GIN [Xu et al., 2019] on two commonly used molecular benchmarks: ZINC [Gómez-Bombarelli et al., 2018; Sterling and Irwin, 2015] and ogbg-molhiv [Hu et al., 2020]. For this, we compare CAT+GIN against GIN on both datasets. We tune the hyperparameters on the validation sets and evaluate the best performing hyperparameters 10 times. For each dataset we track a commonly used evaluation metric for this dataset and report the mean and standard deviation of this metric in the epoch with the highest validation performance. More details can be found in Appendix C and in our code repository[2]. Table 1 shows the results of our experiments and we can see that CAT+GIN convincingly outperforms GIN on both datasets. To measure the speed of CAT we measure its runtime on the training splits of ZINC and MOLHIV averaged over 10 trials. CAT requires 38 ± 1s for ZINC and 133 ± 1s for MOLHIV.

Table 1: Test performance of different GNNs over 10 random seeds. Arrows indicate whether smaller (↓) or bigger (↑) results are better. **Bold** indicates the best performing model for a dataset.

Model	ZINC MAE ↓	MOLHIV ROC-AUC ↑
GIN	0.177 ± 0.006	76.5 ± 1.2
CAT+GIN	$\mathbf{0.125 \pm 0.006}$	$\mathbf{78.4 \pm 0.6}$

5 Conclusion

We proposed a graph transformation, which enables the Weisfeiler-Leman test to be maximally expressive on outerplanar graphs by building on the fact that biconnected outerplanar graphs can be uniquely identified by their Hamiltonian adjacency list sequences. Our transformation encodes these HAL sequences in unfolding trees. We achieve promising first empirical results on molecular benchmark datasets, while keeping the pre-processing time very low. Interesting further directions would be extending the transformation to non-outerplanar graphs and investigating the effect of the transformation on the issues of oversquashing and oversmoothing.

[2] https://github.com/ocatias/OuterplanarGNNs_LoG

Acknowledgements

This work was supported by the Vienna Science and Technology Fund (WWTF) through project VRG19-009 and through project ICT22-059.

References

Ralph Abboud, İsmail İlkan Ceylan, Martin Grohe, and Thomas Lukasiewicz. The surprising power of graph neural networks with random node initialization. In *IJCAI*, 2021. 8

Vikraman Arvind, Johannes Köbler, Gaurav Rattan, and Oleg Verbitsky. On the power of color refinement. In *Fundamentals of Computation Theory*, 2015. 2, 4

Franka Bause, Fabian Jogl, Patrick Indri, Tamara Drucks, David Penz, Nils Morten Kriege, Thomas Gärtner, Pascal Welke, and Maximilian Thiessen. Maximally expressive GNNs for outerplanar graphs. In *New Frontiers in Graph Learning Workshop (GLFrontiers@NeurIPS)*, 2023. 1

Beatrice Bevilacqua, Fabrizio Frasca, Derek Lim, Balasubramaniam Srinivasan, Chen Cai, Gopinath Balamurugan, Michael Bronstein, and Haggai Maron. Equivariant subgraph aggregation networks. In *ICLR*, 2021. 8

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com. 7

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yu Guang Wang, Pietro Liò, Guido Montúfar, and Michael Bronstein. Weisfeiler and Lehman go cellular: CW networks. In *NeurIPS*, 2021. 8

Charles J. Colbourn and Kellogg S. Booth. Linear time automorphism algorithms for trees, interval graphs, and planar graphs. *SIAM Journal on Computing*, 10(1):203–225, 1981. 1, 2, 7

Leonardo Cotta, Christopher Morris, and Bruno Ribeiro. Reconstruction for powerful graph representations. *NeurIPS*, 2021. 8

George Dasoulas, Ludovic Dos Santos, Kevin Scaman, and Aladin Virmaux. Coloring graph neural networks for node disambiguation. In *IJCAI*, 2020. 8

Andre Droschinsky, Nils Kriege, and Petra Mutzel. Finding largest common substructures of molecules in quadratic time. In *International Conference on Current Trends in Theory and Practice of Informatics*, 2017. 1

Stefan Felsner. *Geometric graphs and arrangements: some chapters from combinatorial geometry*. Springer, 2012. 2

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. 7

Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 2018. 4, 7

Tamás Horváth and Jan Ramon. Efficient frequent connected subgraph mining in graphs of bounded tree-width. *Theoretical Computer Science*, 411(31-33):2784–2797, 2010. 1, 8

Tamás Horváth, Jan Ramon, and Stefan Wrobel. Frequent subgraph mining in outerplanar graphs. In *KDD*, 2006. 1, 8

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open Graph Benchmark: Datasets for machine learning on graphs. In *NeurIPS*, 2020. 4, 7

Neil Immerman and Eric Lander. *Describing graphs: A first-order approach to graph canonization*. Yale University Press, 1990. 8

Fabian Jogl, Maximilian Thiessen, and Thomas Gärtner. Expressivity-preserving GNN simulation. In *NeurIPS*, 2023. 1

Sandra Kiefer. *Power and limits of the Weisfeiler-Leman algorithm*. PhD thesis, RWTH Aachen University, 2020. 4, 8

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. *NeurIPS*, 2019a. 8

Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. In *ICLR*, 2019b. 1

Sandra L. Mitchell. Linear algorithms to recognize outerplanar and maximal outerplanar graphs. *Information Processing Letters*, 9(5):229–232, 1979. 2, 4

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *AAAI*, 2019. 8

Christopher Morris, Gaurav Rattan, and Petra Mutzel. Weisfeiler and leman go sparse: Towards scalable higher-order graph embeddings. In *NeurIPS*, 2020. 1

Teague Sterling and John J. Irwin. Zinc 15 – ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 2015. 4, 7

Robert Endre Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal of Computing*, 1 (2):146–160, 1972. 4

Petar Veličković. Message passing all the way up. In *ICLR Workshop on Geometrical and Topological Representation Learning*, 2022. 1

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019. 1, 4, 8

Bohang Zhang, Shengjie Luo, Liwei Wang, and Di He. Rethinking the expressive power of GNNs via graph biconnectivity. In *ICLR*, 2023. 8

A Outerplanarity of Graph Datasets

Table 2 shows the percentage of outerplanar graphs in popular molecular datasets.

Table 2: Common benchmark datasets and the percentage of outerplanar graphs in them: NCI (`https://cactus.nci.nih.gov/`) ZINC [Gómez-Bombarelli et al., 2018; Sterling and Irwin, 2015] and all other datasets are from ogb [Hu et al., 2020].

Dataset	#Graphs	Outerplanar
NCI	250251	94%
ZINC	12000	98%
molhiv	41127	92%
moltox21	7831	96%
molesol	1128	97%
molbace	1513	93%
molclintox	1477	94%
molbbbp	2039	92%
molsider	1427	92%
moltoxcast	8576	96%
mollipo	4200	96%

B Example for Hamiltonian Adjacency Lists

Figure 3 shows two graphs and their directed Hamiltonian cycles. The nodes are annotated with their HAL, the list of distances on the Hamiltonian cycle to their neighbors.

C Experimental Evaluation

Our models are implemented in PyTorch-Geometric [Fey and Lenssen, 2019] and trained on a single NVIDIA GeForce RTX 3080 GPU. We use WandB [Biewald, 2020] for tracking. The used server has 64 GB of RAM, has an 11th Gen Intel(R) Core(TM) i9-11900KF CPU running at 3.50GHz and uses Fedora 38. For `ZINC` we train with a batch size of 128 and an initial learning rate of 10^{-3} that is halved whenever the validation metric does not improve for 20 epochs. Training stops after 500 epochs or after the learning rate dips below 10^{-5}. For `MOLHIV` we train with a batch size of 128 and a fixed learning rate of 10^{-3} for 100 epochs. Table 3 shows the hyperparameters for GIN and CAT+GIN. We used the same hyperparameters for both models. We used a smaller hyperparameter grid for `MOLHIV` than for `ZINC`, as `MOLHIV` is larger than `ZINC` meaning that training takes much longer.

More details on CAT. CAT adds an additional feature to each node which encodes the type of that node i.e., nodes from Hamiltonian cycles, block nodes, pooling nodes, articulation nodes and or global block nodes. Furthermore, we create additional edge features encoding the types of nodes incident to this edge i.e., an edge between two different nodes in a Hamiltonian cycle has a different type than an edge from a pooling node to the block node. For newly created nodes and edges we set their remaining features to the feature of the node / edge they are based on; for example, a pooling node will have the features of the node they are performing the pooling operation for. For nodes that

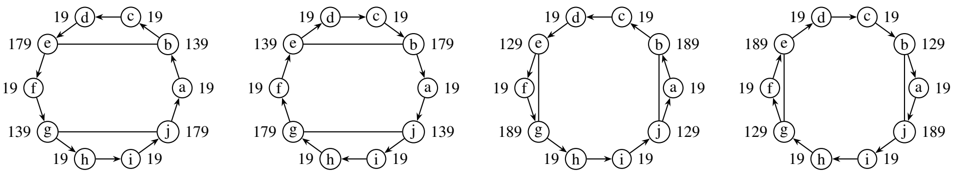

Figure 3: Two graphs and their directed Hamiltonian cycles. Nodes are annotated with their HALs, the distances on the Hamiltonian cycle to their neighbors [Colbourn and Booth, 1981].

have no natural representation in the graph (block and block pooling nodes) we set these features to 0. To ensure that only these nodes get assigned 0 features, we shift the values of these features for all other nodes by 1. Note that our GNN treats the distance on edges in blocks as a categorical feature. We are not sure whether this is advantageous and will experiment with treating these features as ordinals in the future.

Table 3: Hyperparamter grids for different datasets.

Parameter	GIN, CAT+GIN On ZINC	GIN, CAT+GIN On MOLHIV
Message passing layers	2, 3, 4, 5	4, 5
Final MLP layers	2	2
Pooling operation	mean, sum	mean, sum
Embedding dimension	64, 128, 256	64,128
Jumping knowledge	last	concat
Dropout rate	0, 0.5	0.5

D Discussion and Related Work

It is well known that the expressiveness of MPNNs is bounded by the 1-WL test [Morris et al., 2019; Xu et al., 2019]. The unlabled graphs corresponding to decalin and bicyclopentyl cannot be distinguished by any MPNN. As these two graphs are outerplanar this shows that MPNNs are not expressive enough for outerplanar graphs. Even some biconnected outerplanar graphs cannot be distinguished by MPNNs, see Fig. 3. The importance of biconnectivity in the context of GNNs was recently discussed by Zhang et al. [2023]. The fact that many pharmaceutical molecules are outerplanar is well known in the graph mining community [Horváth and Ramon, 2010; Horváth et al., 2006]. Cotta et al. [2021] discussed outerplanar graphs in the context of reconstruction with GNNs.

It is known that 3-WL is sufficient and necessary to distinguish all outerplanar graphs [Kiefer, 2020]. Hence, any GNN matching the expressivity of 3-WL, such as 3-IGN [Maron et al., 2019a] or 3-GNN [Morris et al., 2019], is capable of solving our main goal of distinguishing all outerplanar graphs. The 3-WL test, however, runs in $\mathcal{O}(n^3 \log n)$ time [Immerman and Lander, 1990; Kiefer, 2020], which is practically inefficient already for medium-sized graphs. Similarly, 3-GNN and 3-IGN run in roughly $\mathcal{O}(n^3)$ time, see Maron et al. [2019a]; Morris et al. [2019]. Contrary, we perform a linear-time pre-processing of the graph and run standard 1-WL, or an MPNN, with a much more efficient runtime of $\mathcal{O}(n \log n)$. This is the same asymptotic runtime as running 1-WL on the original graphs, as we only add a linear number of nodes and edges in our graph transformation.

Finally, there are other approaches towards building more expressive GNNs such as methods that extract subgraphs [Bevilacqua et al., 2021; Maron et al., 2019a] or lift the graph to regular cell complexes Bodnar et al. [2021]. The idea of using additional node labels to increase expressivity is also well known [Abboud et al., 2021; Dasoulas et al., 2020].

E Additional Figures

Fig 5 demonstrates the difference between two blocks overlapping in the same articulation node (top) or in different articulation nodes (bottom). Fig 6 demonstrates CAT on real-life molecular graphs. We would like to point out that one of the graphs looks like a frog.

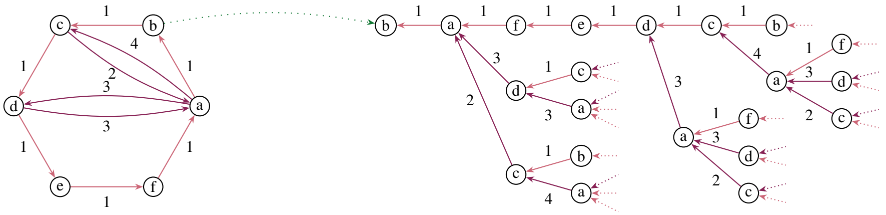

Figure 4: One part of the CAT* transformation of the graph from Figure 2 and an example unfolding tree of one of its nodes from which the HAL sequence of the original graph can be reconstructed.

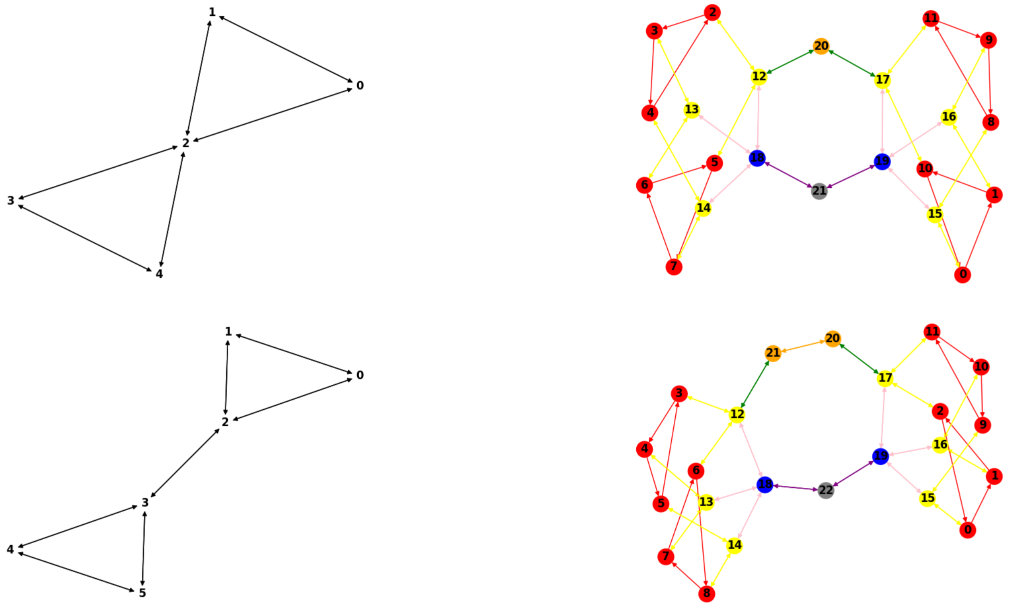

Figure 5: Left: example graphs; Right: Result of applying CAT to these graphs. Colors indicate the type of node: red nodes are from Hamiltonian cycles, blue nodes correspond to blocks, yellow nodes pool nodes from the Hamiltonian cycles, orange nodes correspond to articulation nodes and the grey node pools block nodes.

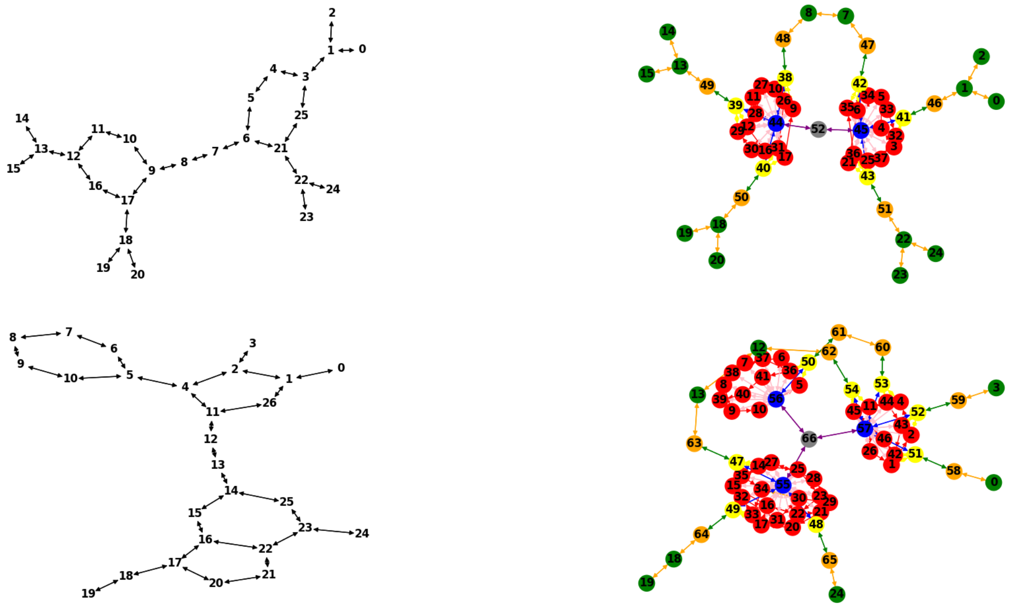

Figure 6: Left: example graphs from MOLHIV (top) and ZINC (bottom); Right: Result of applying CAT to these graphs. Colors indicate the type of node: red nodes are from Hamiltonian cycles, blue nodes correspond to blocks, yellow nodes pool nodes from the Hamiltonian cycles, orange nodes correspond to articulation nodes and the grey node pools block nodes.