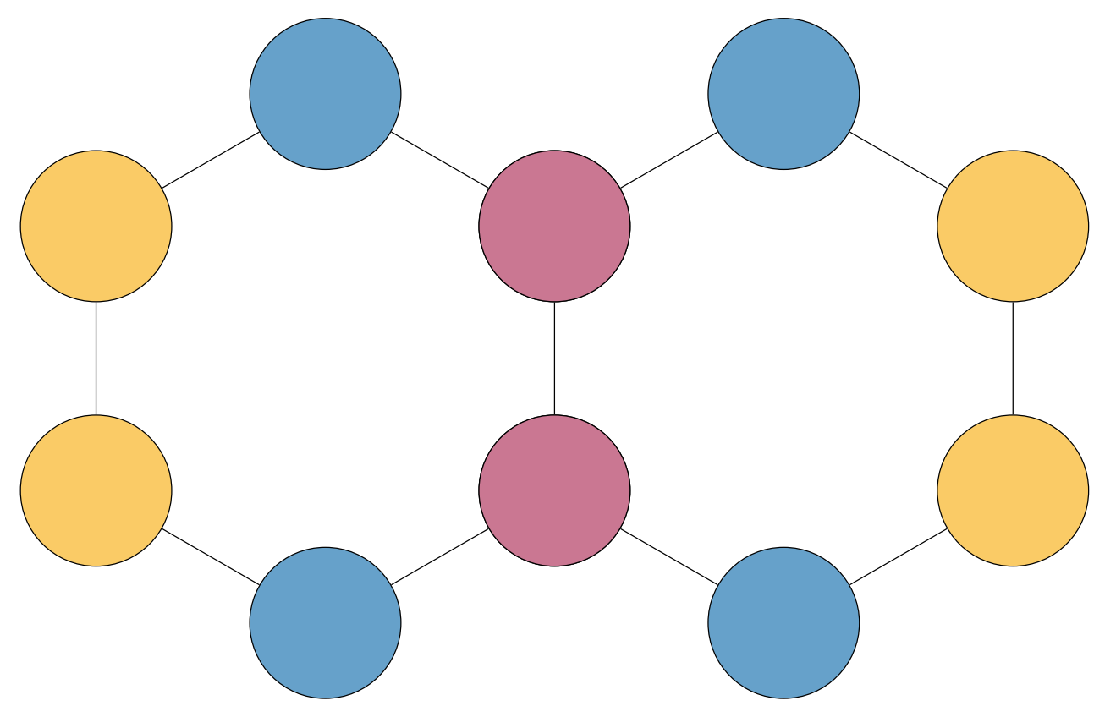
Comparison of Graph-Encoding Trees

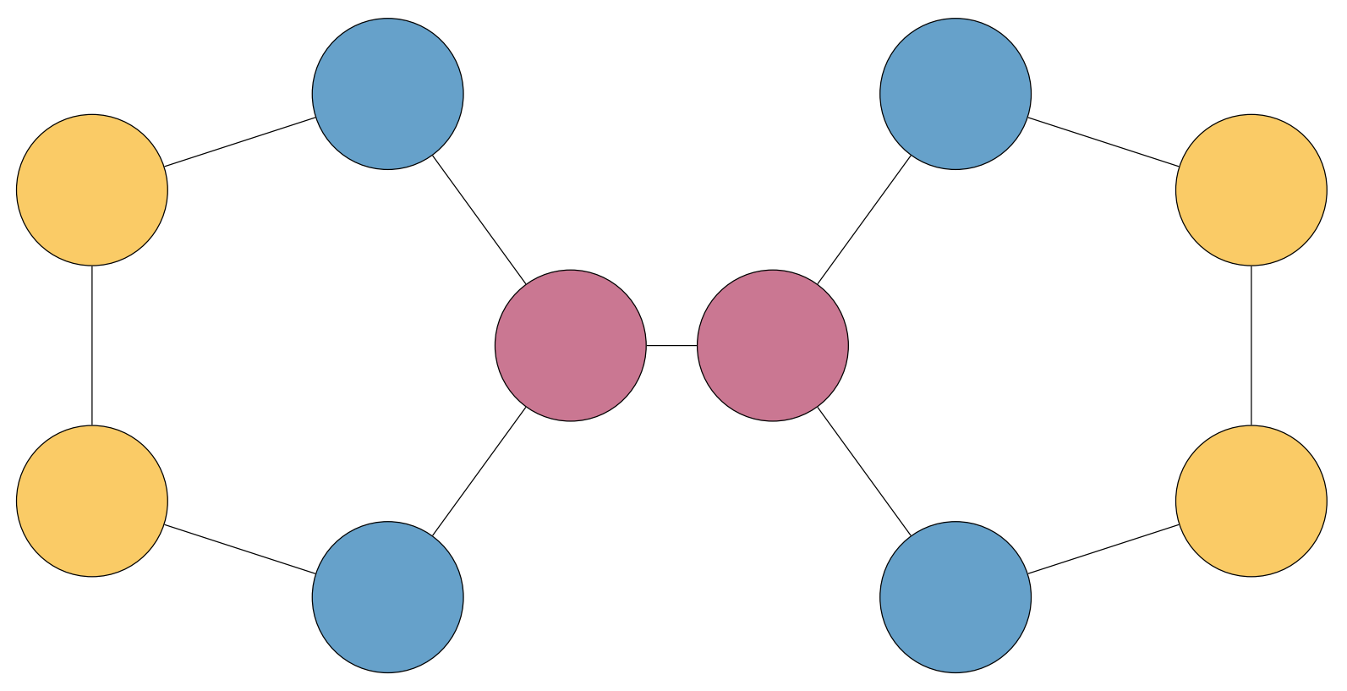
Ariana Fox

Supervisors: Assoz. Prof. Dr. Nils Kriege and Franka Bause

Introduction

The graph isomorphism problem is difficult:

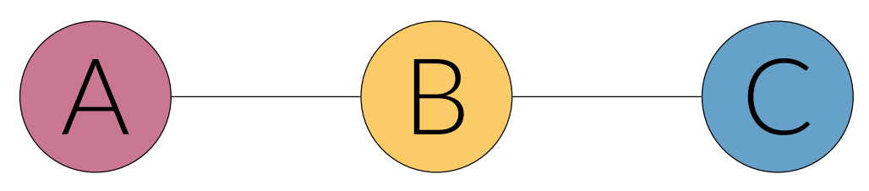

Declain Molecule[1]


Bicyclopentyl Molecule[1]

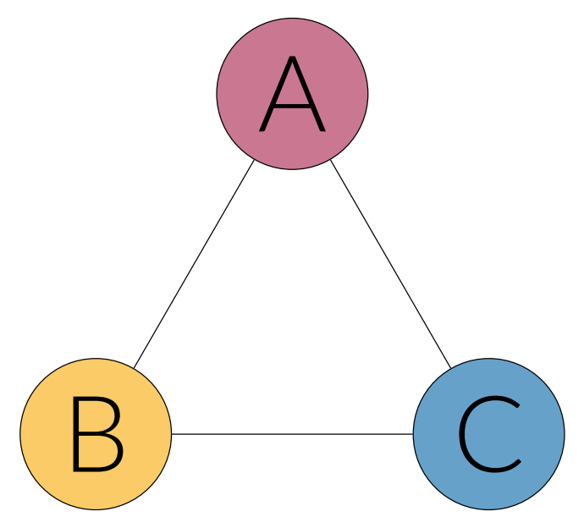
To address this, we derive acyclic trees, which encode neighborhoods and are easier to compare. These trees:

- provide tractable heuristics for graph isomorphism, and
- are fundamental for graph learning methods, e.g., GNNs.

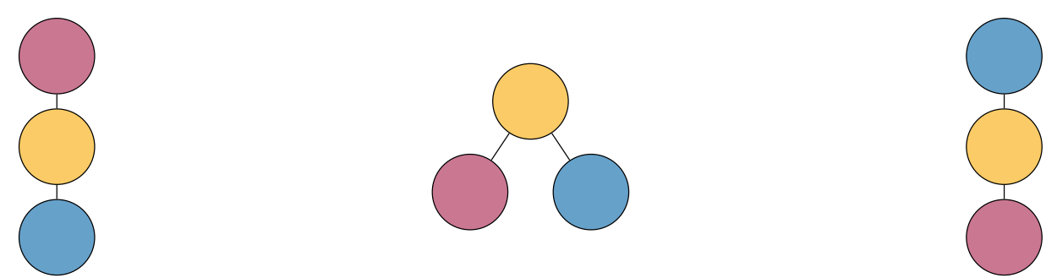
One tree is created for each node in the original graph. Differing tree sets imply graphs are not isomorphic, but identical tree sets don't guarantee isomorphism.

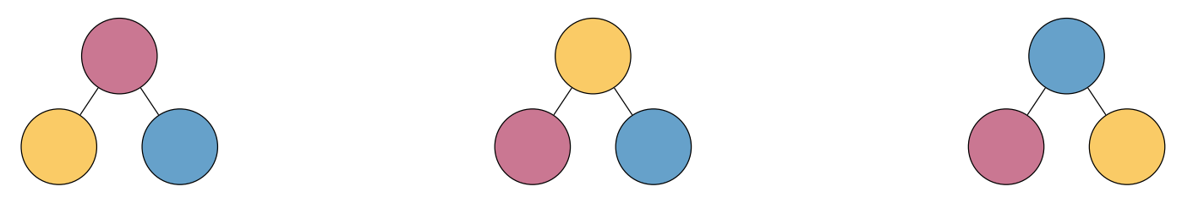


Graph I



Graph II


Encoding Tree Set for Graph I

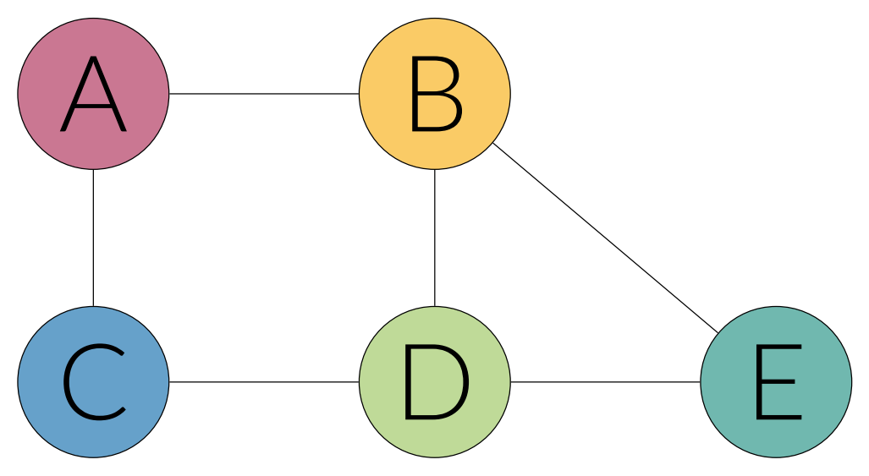

Encoding Tree Set for Graph II

Goal

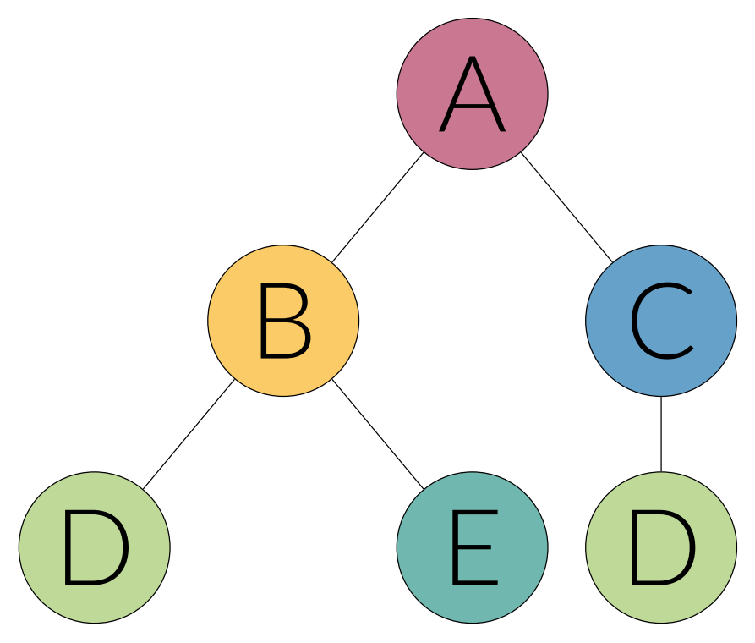
Experimentally evaluate the effectiveness and efficiency of two types of graph-encoding trees.

Graph-Encoding Trees

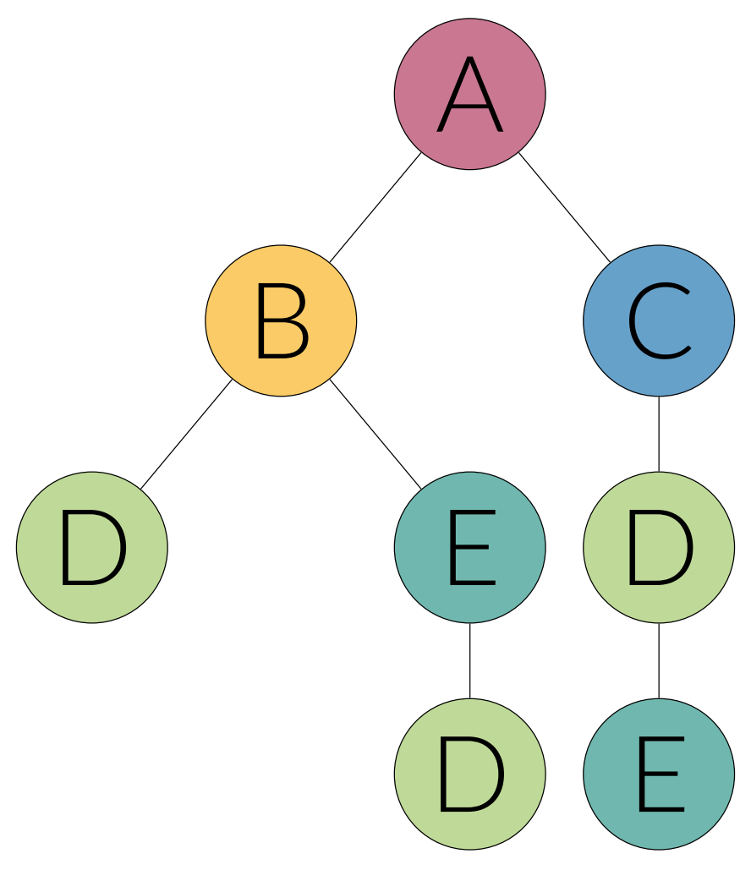
- The **k-Redundant Neighborhood Tree (k-NT)** captures all neighbors of the node permitting redundancy at each level up to k iterations before the current node.[2]
- The **Truncated ePath Tree (TPT)** captures all neighbor relationships where only previously unvisited nodes are allowed to be added onto the path, unless the last node is equal to the root node.[3]



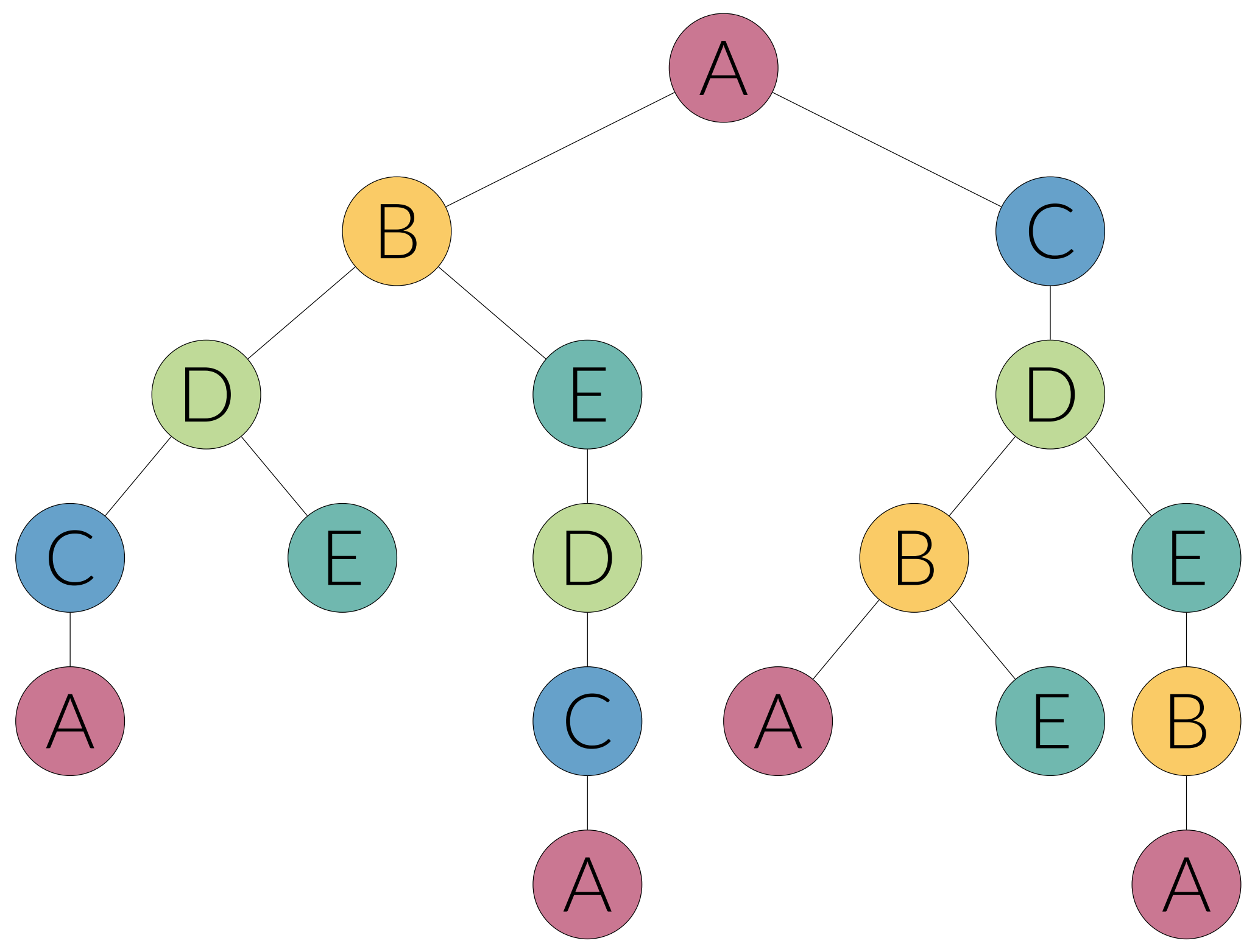
Original Graph



0-NT for Node A



1-NT for Node A

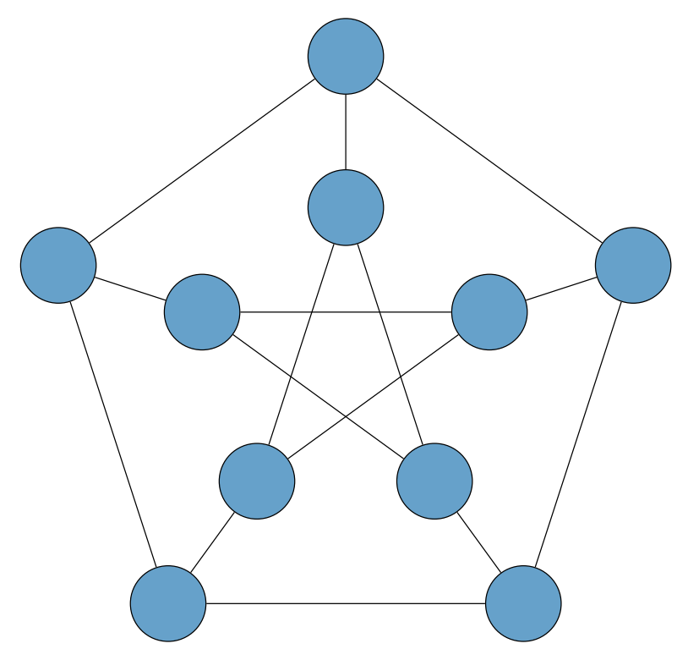


TPT for Node A

Graph Types

We tested on several different types of graphs, including simple, Eulerian, perfect, Hypohamiltonian, planar, self-complementary, and highly irregular.

The popular Weisfeiler-Leman color refinement algorithm is known to fail to distinguish strongly regular graphs,[1] so this graph type was of the most interest.


Strongly Regular Graph: srg(10,3,0,1) commonly known as the Petersen Graph

Expressiveness Results

- We found no cases where the TPT failed to distinguish unique graphs.
- The 1-NT only failed to distinguish strongly regular graphs.
- The 0-NT failed to distinguish several types of graphs beyond only strongly regular graphs.

Prior to this experimentation, there were no known cases where the 0-NT and 1-NT had different distinguishing capabilities.

Runtime and Memory Usage

The 0-NT consistently took the least time and used the least memory, followed by the 1-NT and then the TPT.

Graph Type (Num. of Nodes)	0-NT	1-NT	TPT
Simple (8)	5.580e00	1.385e01	1.318e02
Eulerian (9)	1.254e00	3.664e00	1.154e02
Perfect (8)	4.023e00	1.004e01	9.611e01

Table 1. Total Runtime

Graph Type (Num. of Nodes)	0-NT	1-NT	TPT
Simple (8)	10.76	30.04	231.54
Eulerian (9)	13.01	42.51	810.51
Perfect (8)	10.75	30.30	233.50

Table 2. Avg. Number of Nodes per Tree

References

[1] Nurudin Alvarez-Gonzalez, Andreas Kaltenbrunner, and Vicenç Gómez.
Beyond weisfeiler–lehman with local ego-network encodings.
Machine Learning and Knowledge Extraction, 5(4):1234–1265, 2023.

[2] Franka Bause, Christian Permann, and Nils M. Kriege.
Approximating the graph edit distance with compact neighborhood representations.
In Albert Bifet, Jesse Davis, Tomas Krilavičius, Meelis Kull, Eirini Ntoutsi, and Indrė Žliobaitė, editors, *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 300–318, Cham, 2024. Springer Nature Switzerland.

[3] Rongqin Chen, Shenghui Zhang, Leong Hou U, and Ye Li.
Redundancy-free message passing for graph neural networks.
In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.