FAS-GED: GPU-Accelerated Graph Edit Distance Computation.

Adel Dabah, Andreas Herten

Jülich Supercomputing Centre, Forschungszentrum Jülich, DE

a.dabah@fz-juelich.de, a.herten@fz-juelich.de

1) Motivation and Introduction

- **Graph Edit Distance (GED)**: Measures the similarity of graphs by quantifying the minimum cost of edit operations (insertion, deletion, substitution) needed to transform one graph into another.

Given graphs $g_1 = (V_1, E_1, \alpha_1, \beta_1)$ and $g_2 = (V_2, E_2, \alpha_2, \beta_2)$, the

$$GED(g_1, g_2) = \min_{\{o_1, o_2, \ldots, o_k\} \in \gamma(g_1, g_2)} \sum_{i=1}^{k} cost(o_i)$$

where $\gamma(g_1, g_2)$ denotes all possible edit paths.

- **Applications**: GED is used in fields like bioinformatics, computer vision, and pattern recognition, enabling tasks such as molecular structure comparison and image matching.
- **Challenge**: High complexity that increases exponentially with the size of graphs, making efficient GED computation crucial for large graphs.

Proposed Solution: FAS-GED

- **Goal**: Develop a Fast, Accurate, and Scalable approach for GED computation leveraging GPU architectures.
- **Key Features**:
  - High accuracy comparable to exact methods with significantly reduced computation time.
  - Efficient best-k elements retrieval on GPU to balance accuracy/scalability. A larger value of K prioritizes accuracy, while a smaller value of K enhances scalability.
  - Efficient utilization of GPU capabilities to handle large graphs with minimal host-device communication.

2) FAS-GED Methodology

FAS-GED explores the search tree level by level on GPUs and cuts off all nodes with a ranking greater than a given parameter (K) [1].

- **Three-Phase GPU Process**:
  - **Branching Phase**:
    - Apply edit operations to expand the nodes at a given tree level.
    - Efficiently leverage GPU parallelism for simultaneous node expansion and evaluation of the partial edit distance.
  - **Ranking Phase**:
    - Rank the expanded nodes using local and global ranking mechanisms within GPU blocks, exploiting atomic operations without host intervention.
  - **Update Phase**:
    - Update primary data structures with the best K nodes' data based on the ranking.
- **Benefits**:
  - Reduces host-device communication, significantly improving performance.
  - Scales efficiently with graph size in a linear complexity.

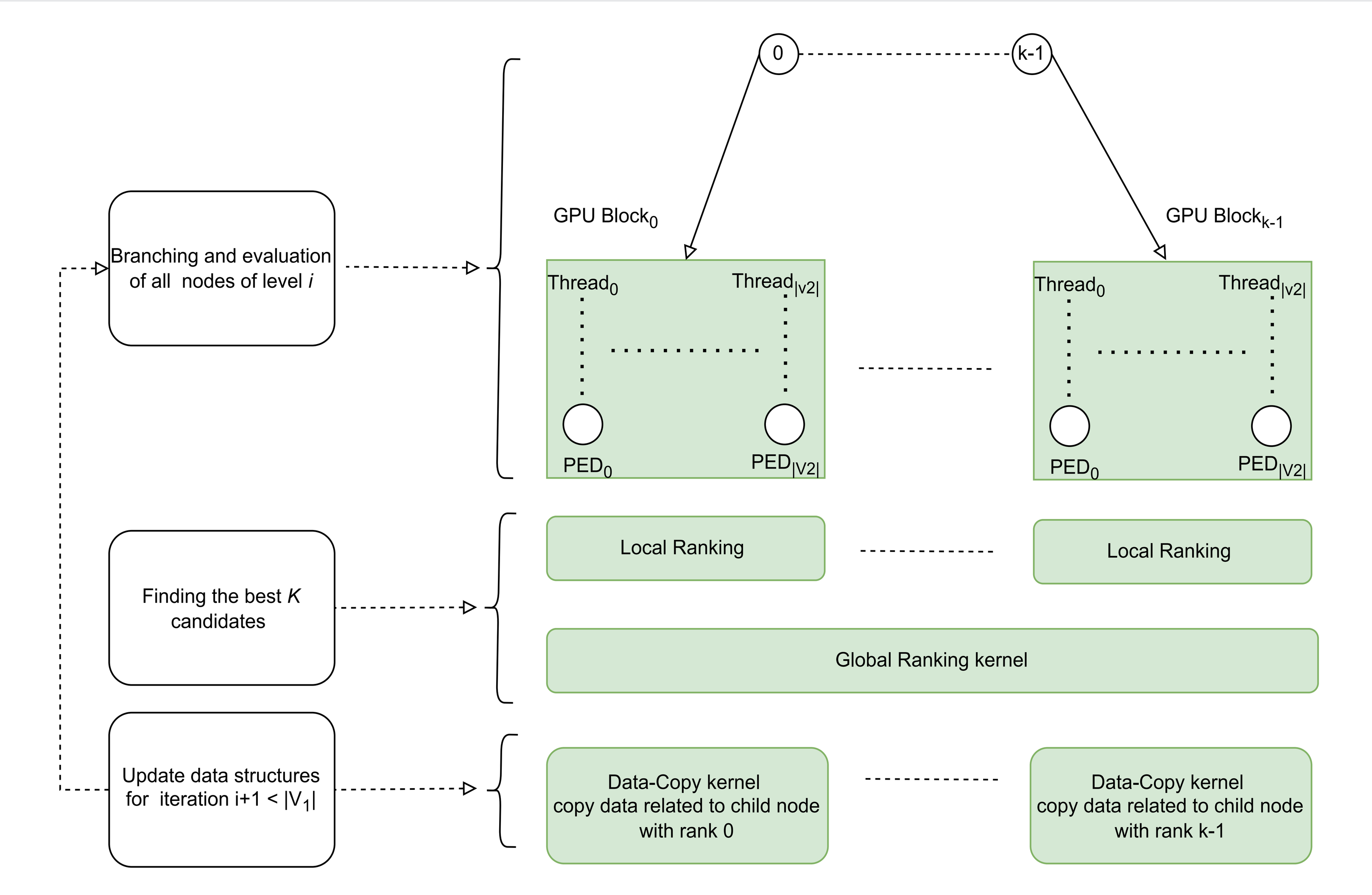

Figure 1: FAS-GED GPU implementation.

3) Performance Evaluation

- **Accuracy**: FAS-GED achieves the optimal edit distance in over 90% of cases with a deviation less than 0.5% using synthetic graphs with varying sizes and densities.
- **Speedup**: Up to 55× speedup over NetworkX library [2] for small size graphs.
- **Scalability**: Support graph sizes up to 1000 vertices, with an excellent accuracy–complexity trade-off.
- **Approximate results**: Demonstrates robust performance across diverse real-world datasets compared to Beam Search (BS) and Depth First Search (DFS) state-of-the-art methods.

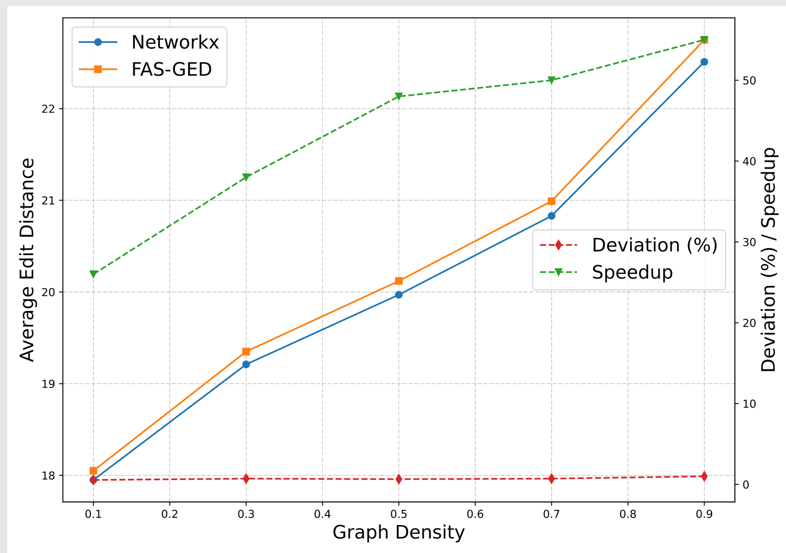

Figure 2: FAS-GED vs. NetworkX open source library.

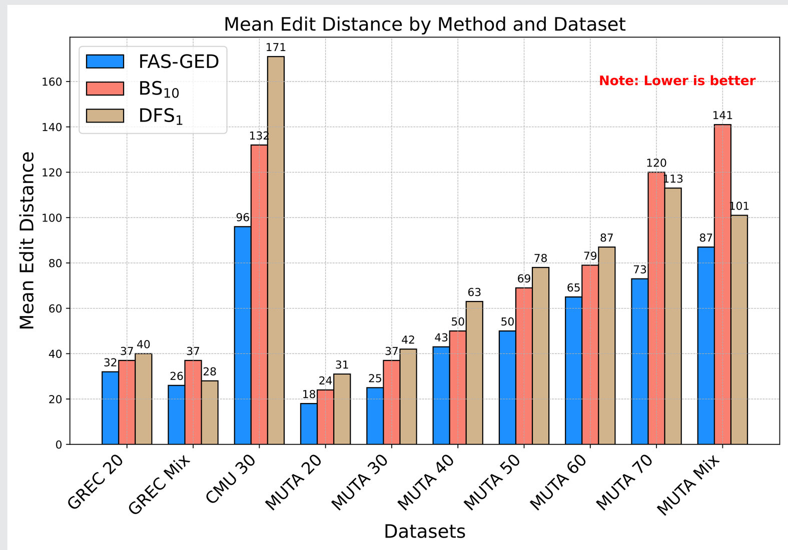

Figure 3: FAS-GED compared to BS and DFS$_1$ results using real-world datasets.

4) Optimization of FAS-GED

Hardware Optimization

- Improve the data layout for efficient memory access, up to 50% improvement over the non-optimized version.
- FAS-GED achieves a 300× speedup over its baseline CPU version on a 48-core AMD Epyc CPU.
- **Bottleneck**: Synchronization overhead in top-k search, accounting for 80% of execution time and explaining performance of A100 to H100 GPUs.

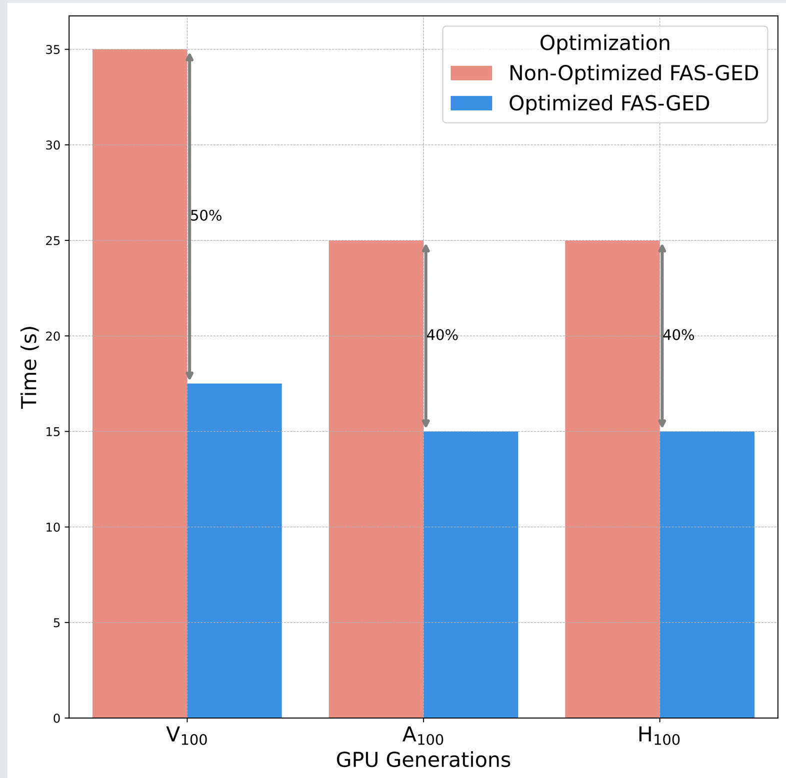

Figure 4: FAS-GED Optimization and hardware scaling.

5) Application in Classification

- Enabling facst accurate GED measurement for K-Nearest Neighbor (KNN) classifier in graph space.
- KNN/FAS-GED reaches similar accuracy compared to sophisticated Graph Neural Network approaches (GNN_NDP [3] and GNN_MEWISPool [4]) on Mutagenicity dataset.

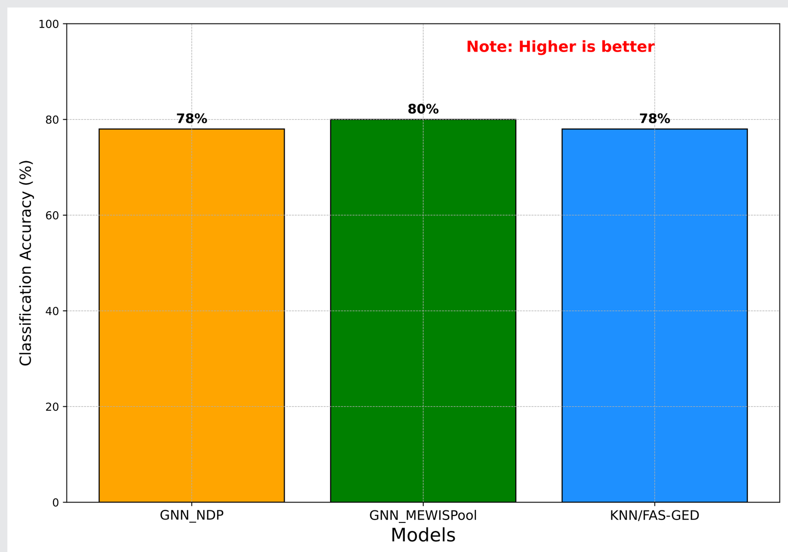

Figure 5: KNN/FAS-GED vs. GNN for Graph Classification.

6) Conclusions

- **Summary**: FAS-GED significantly advances GED computation by balancing speed, accuracy, and scalability on GPU architectures.
- **Future Work**: Focuses on optimization and extending the approach to very large graph sizes, while focusing on the application side.

References:

(1) A. Dabah, I. Chegrane et al., *Pattern Recognit. Lett.*, 2021, **134**, 46–57.

(2) A. A. Hagberg, D. A. Schult et al., *NetworkX*, Web Page, Accessed: 2024-05-20, 2008.

(3) A. Nouranizadeh, M. Matinkia et al., *arXiv preprint arXiv:2107.01410*, 2021.

(4) F. M. Bianchi, D. Grattarola et al., *IEEE Trans. on Neural Networks Learn. Syst.*, 2020, **33**, 2195–2207.