Approximating the Graph Edit Distance with Compact Neighborhood Representations

Franka Bause, Christian Permann, Nils M. Kriege

universität wien

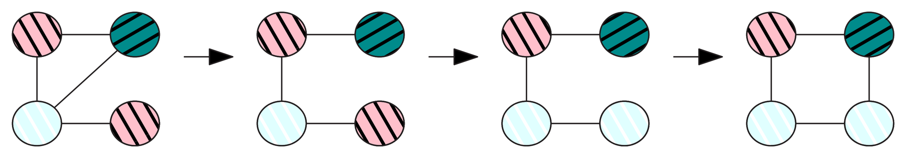
Motivation and Idea

Goal: Approximate graph edit distance

- Use **bipartite graph matching**
- Find better cost function for vertex assignment
- Incorporate **neighborhood information**
- **Efficient** computation

Graph Edit Distance (GED)

- **Distance measure** for **graphs**
- Transform one graph into the other (**edit path**)
- Cost of transformation ≙ GED



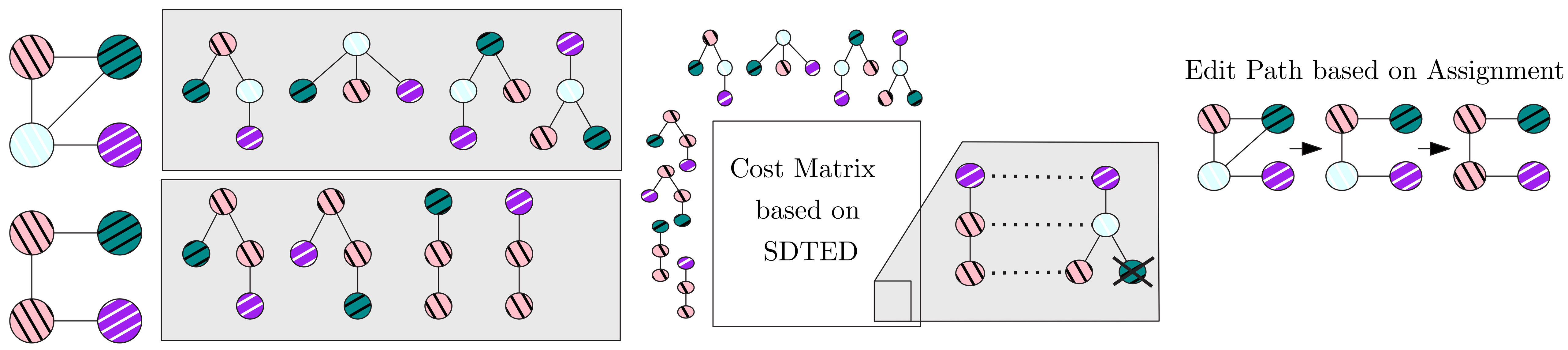
- **NP-hard** problem

Bipartite Graph Matching [Riesen and Bunke, 2009]

- Compute minimal assignment between vertices of the two graphs
  - Create cost matrix
  - Solve assignment problem (cubic runtime)
- Derive edit path from assignment (linear runtime)
- **Approximated GED** ≙ cost of (sub-optimal) edit path

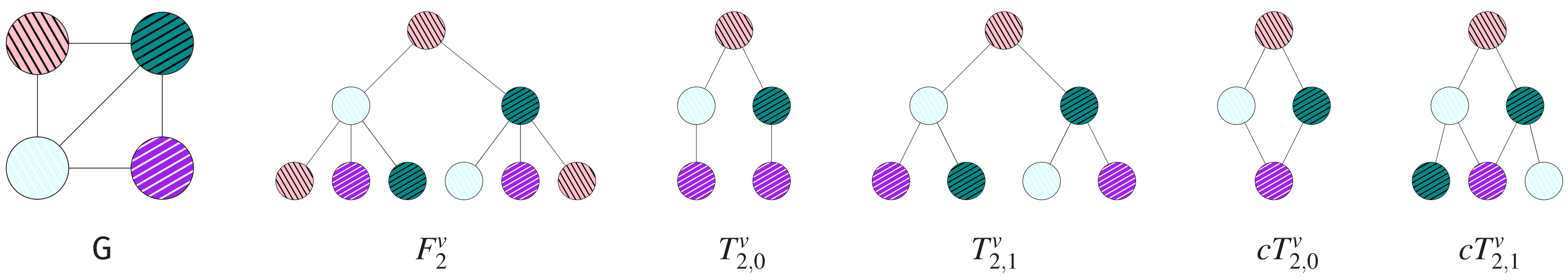
Our Contribution

- Tree structures encoding node neighborhoods and tree edit distance as cost function in **BGM framework**
- **Neighborhood trees**: compact variation of Weisfeiler-Leman unfolding trees
- Tree edit distance in $O(n^2\Delta)$ for two trees with n vertices and maximum degree Δ
- Caching and compression to **accelerate computation**



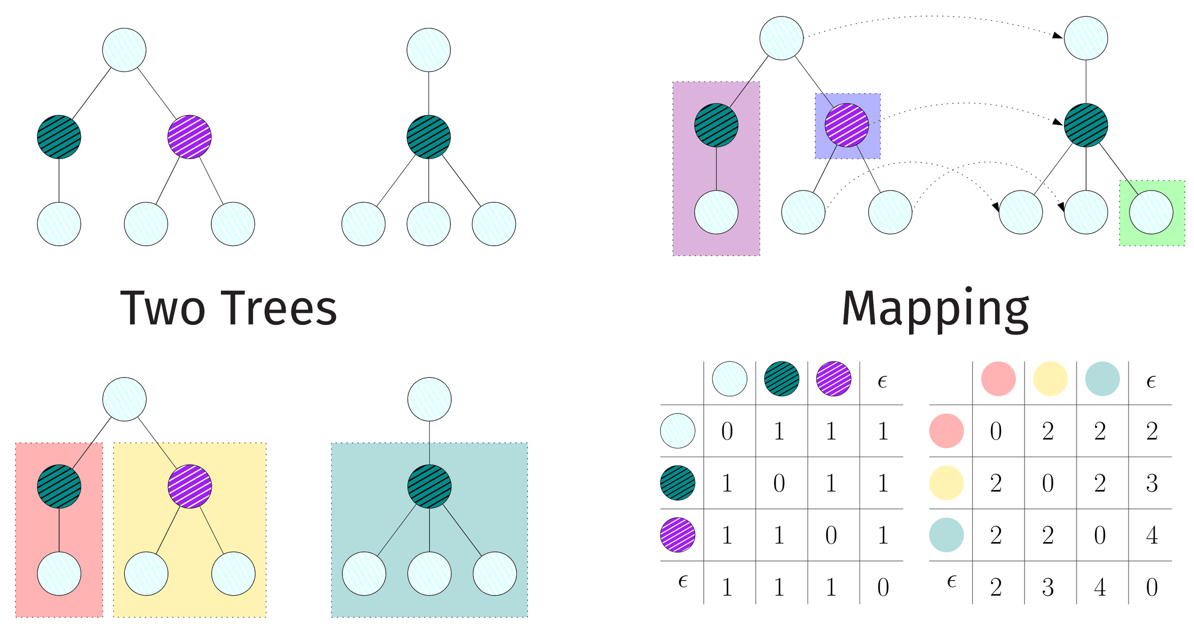
k-Redundant Neighborhood Trees (k-NTs)

- $T^v_{i,k}$: k-NT of $v \in V(G)$ with height i
- **Subtree of unfolding tree** F^v_i
- Node can occur only **up to** k layers after first occurrence
- Compressed versions (cT) with **bounded size** $O(|E(G)| \cdot (k+1))$ and maximum height $\mathrm{diam}(G) + k$



Structure- and Depth-Preserving Tree Edit Distance (SDTED) [Schulz et al., 2022]

- **Distance function** for **rooted trees**
- Find **minimal cost mapping**, so that:
  - Roots are mapped to each other
  - Two nodes are mapped ⇒ parents are mapped
- Find mapping **recursively** by solving optimal assignments on children
- Computation in $O(nn'\Delta)$ time (for two trees with n and n' vertices and maximum degree Δ)



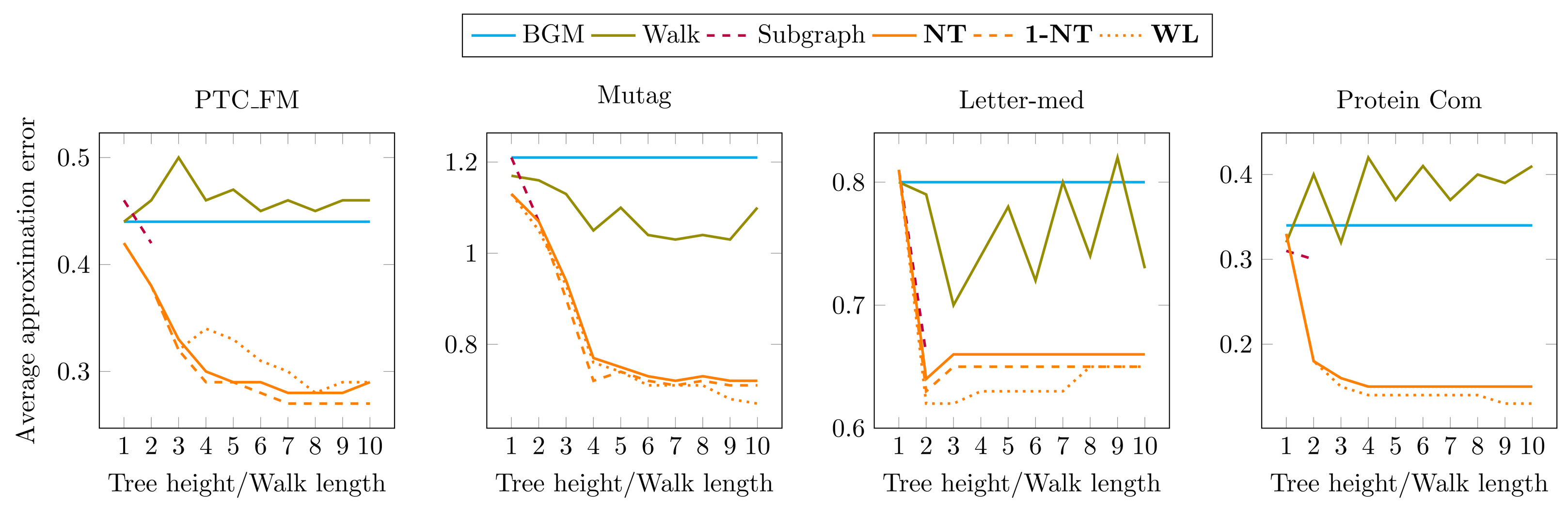
Theoretical Time Complexity (for Creating Cost Matrix)

- Δ: maximum degree
- h: radius of subgraphs/walk length
- ω: exponent of matrix multiplication (typically $\omega \approx 2.81$)
- Bounded-degree graphs: Δ bounded by constant

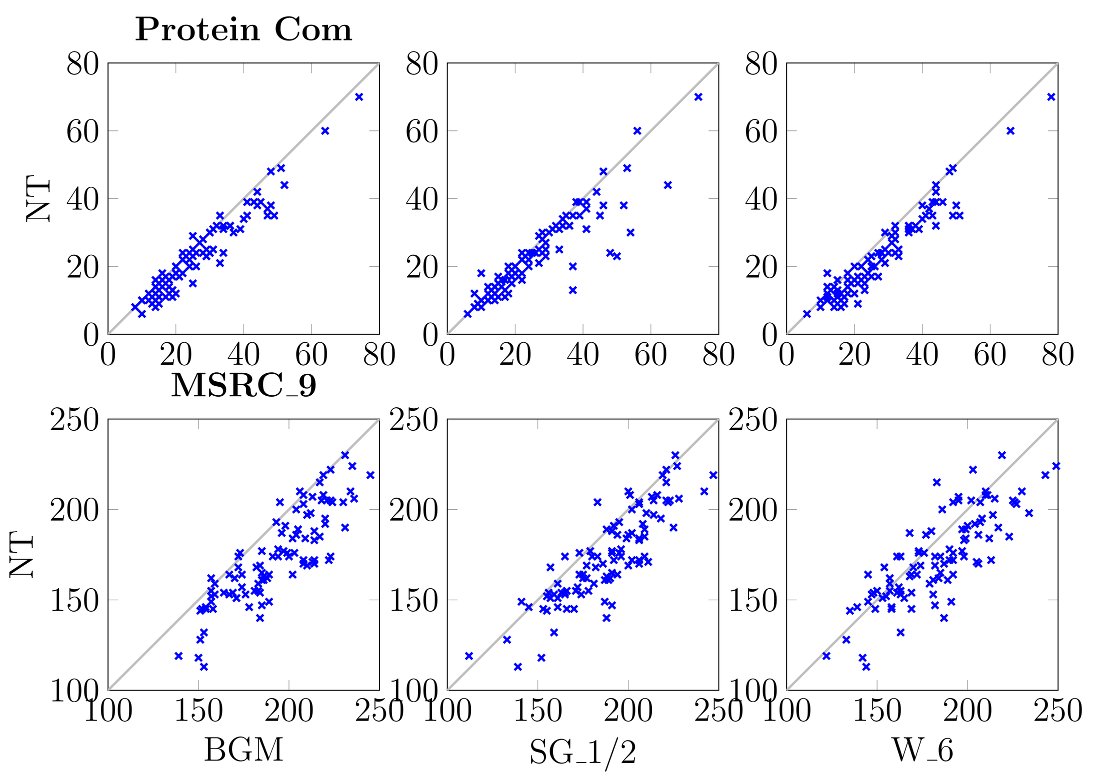
Method	General graphs	Bounded-degree graphs
BGM	$O(\lvert V\rvert^2\Delta^3)$	$O(\lvert V\rvert^2)$
Walk	$O(\log(h)\lvert V\rvert^{2\omega})$	$O(\log(h)\lvert V\rvert^{2\omega})$
Subgraph	$O(\lvert V\rvert^2 \exp(\Delta^h))$	$O(\lvert V\rvert^2 \exp(h))$
Ours	$O(\lvert V\rvert^2\lvert E\rvert^2\Delta)$	$O(\lvert V\rvert^4)$

Quality of Approximation

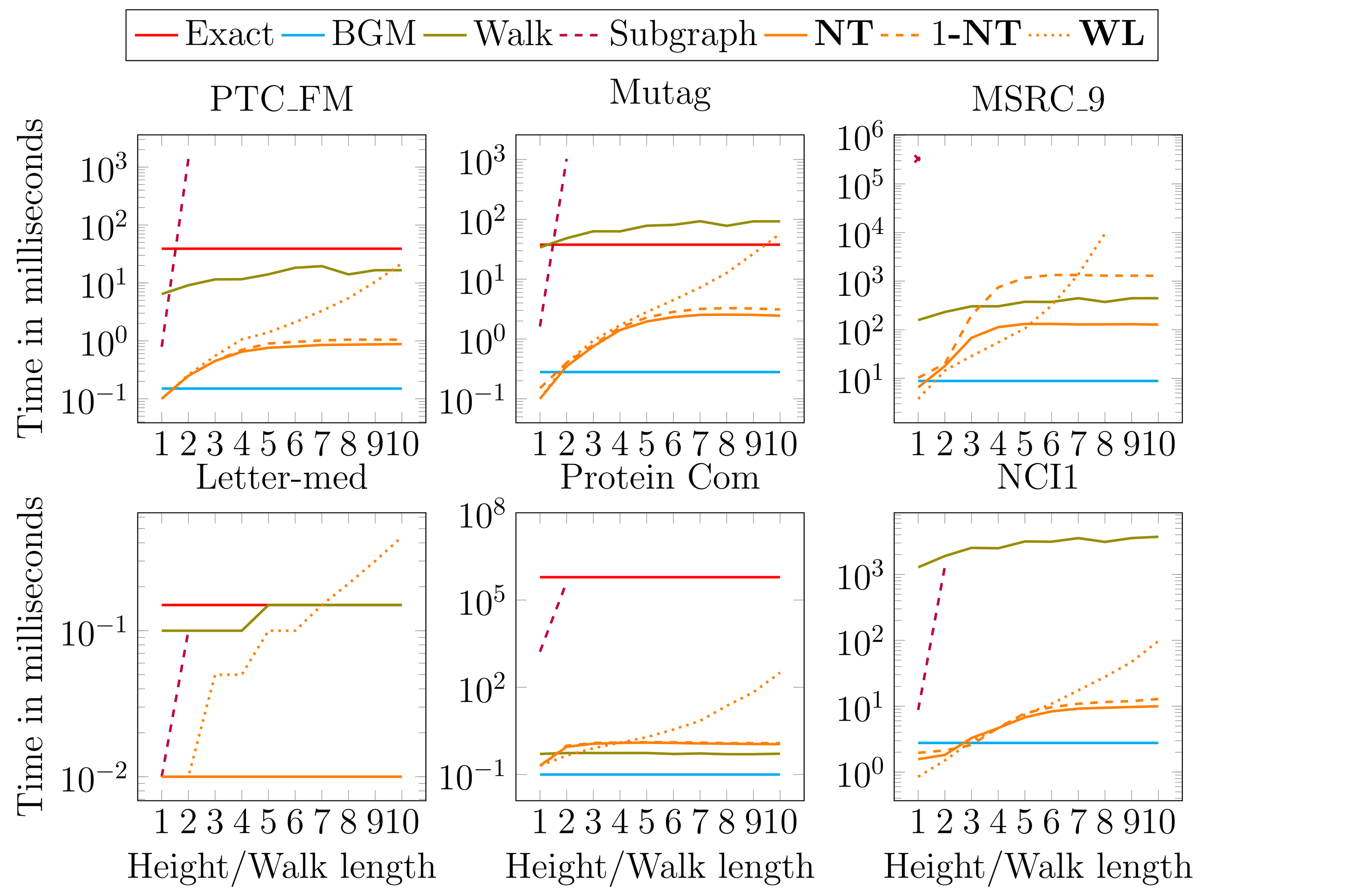
Average Relative Approximation Error



Comparison of Different Bounds



Runtime Comparison



Conclusion

- **Approximation quality better** than computationally more expensive methods
- **Efficient** computation
- Limiting height of trees: **trade-off** between **runtime** and **accuracy**