DiffIM: Differentiable Influence Minimization with Surrogate Modeling and Continuous Relaxation

Junghun Lee, Hyunju Kim, Fanchen Bu, Jihoon Ko, Kijung Shin
{junghun.lee, hyunju.kim, boqvezen97, jihoonko, kijungs} @ kaist.ac.kr

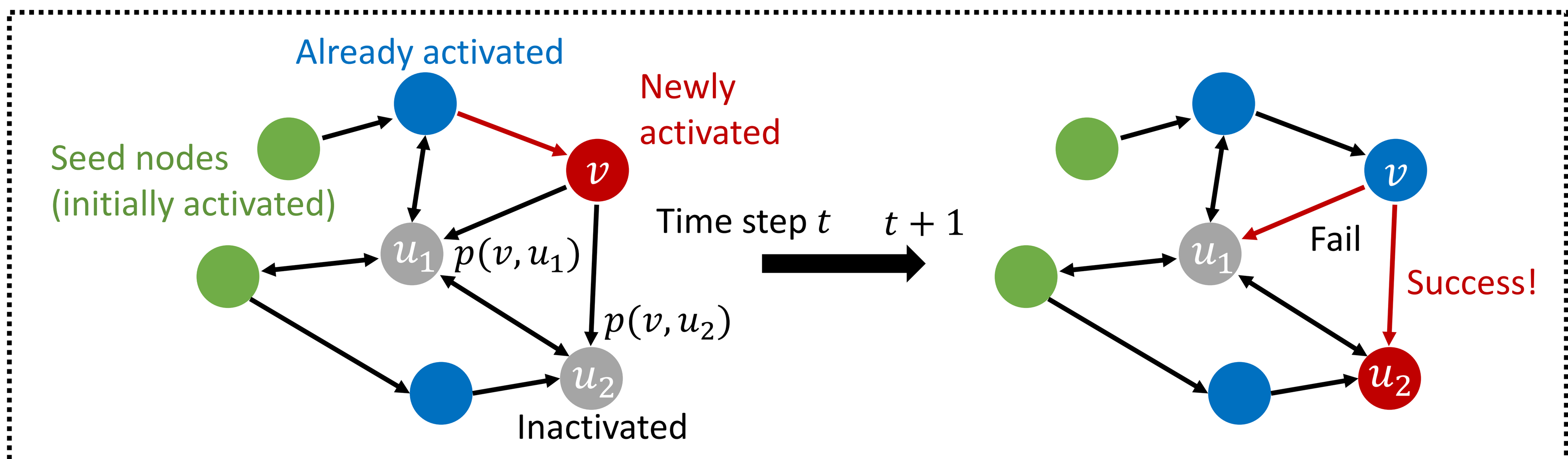
https://github.com/junghunl/DiffIM

Summary

- **Goal:** To minimize influence within social graphs by removing a given number of edges
- **Previous work:** Greedy algorithms using influence estimation based on sampling
- **Proposed methods:**
  - **DiffIM**: Uses a **GNN** as a surrogate model for efficient influence estimation
  - **DiffIM+**: Introduces **continuous relaxation** to edge removal decisions, enabling optimization though **differentiation**
  - **DiffIM++**: Selects edges for removal **instantly** based on **gradients**
- **Experiments:**
  - **Influence estimation**: GNN-based influence estimation shows **near-perfect correlation with the ground truth** and faster than MC simulations
  - **Performance**: DiffIMs reduce influence **faster and more effectively** than baselines
  - **inductivity**: DiffIMs are effective even on **graphs unseen during training**

Influence Minimization: Problem Definition

- **Motivation**: People influence others through social links (e.g. rumor, disease), and it is important to **quickly suppress the spread** before it explodes
- **Basic concepts**
  - **Independent cascade model**: A widely-used mathematical model of spread



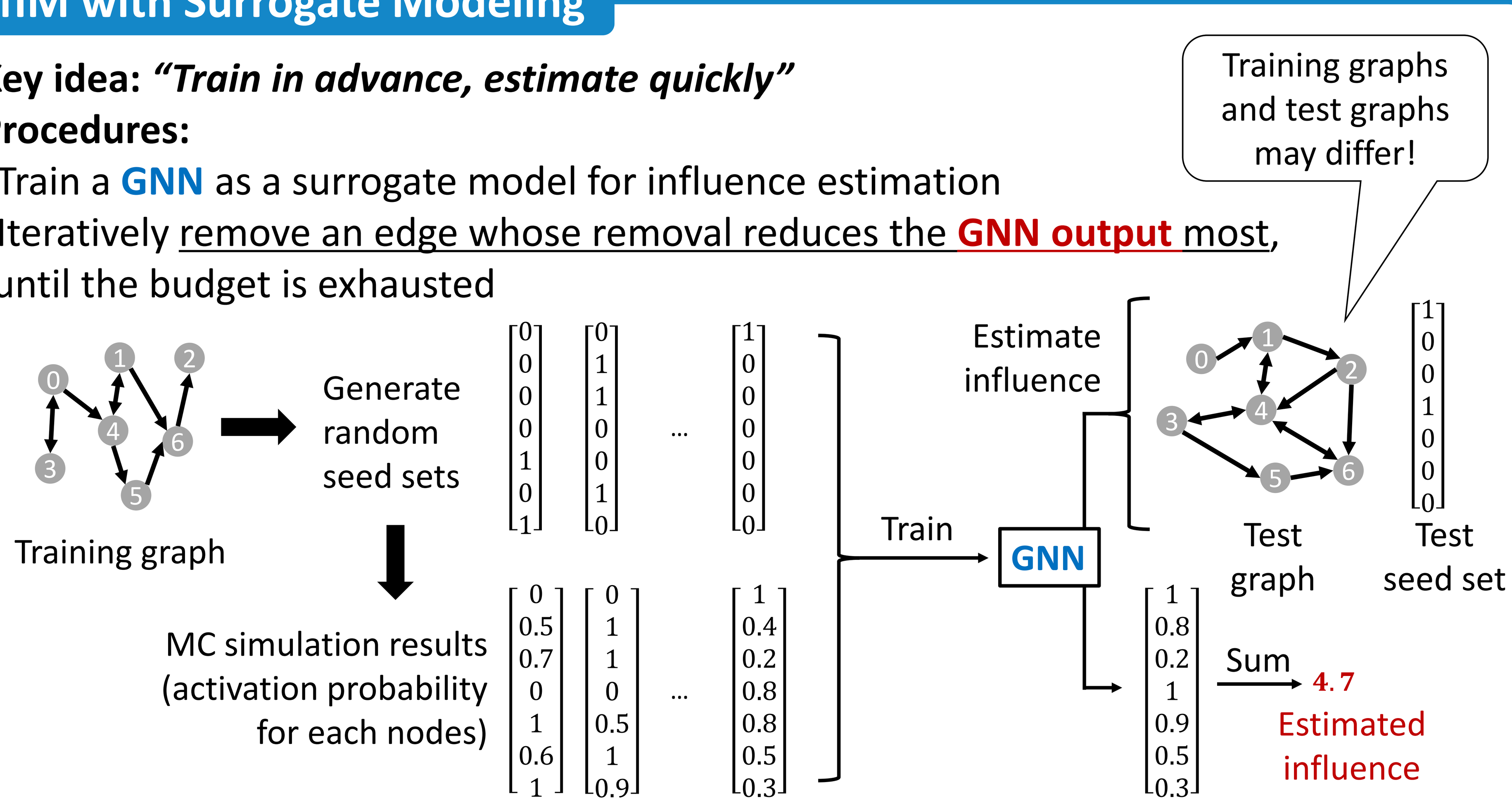
  - **Influence**: Expected number of finally activated nodes
- **Problem definition of influence minimization:**
  - **Input:** A graph with activation probabilities, seed nodes, and an edge-count budget
  - **Output:** A set of edges to remove within the budget.
  - **Goal:** To minimize influence after removing the output edges

Greedy Algorithm and Limitations

- **Procedure:** Iteratively remove an edge whose removal reduces influence most, until the budget is exhausted
- **Challenges:** Exact computation of influence is computationally prohibitive
- **Remedy:** Employ **sampling** to estimate influence (e.g., Monte Carlo, Bond percolation method, Reverse influence sampling)
- **Remaining limitations:**
  ① Requires **a large number of samples** for accurate estimation
  ② Must estimate influence **individually for each edge removal**
- **Time complexity:** $O(budget \cdot \underbrace{\# samples}_{①} \cdot \underbrace{|E|^2}_{②})$

DiffIM with Surrogate Modeling

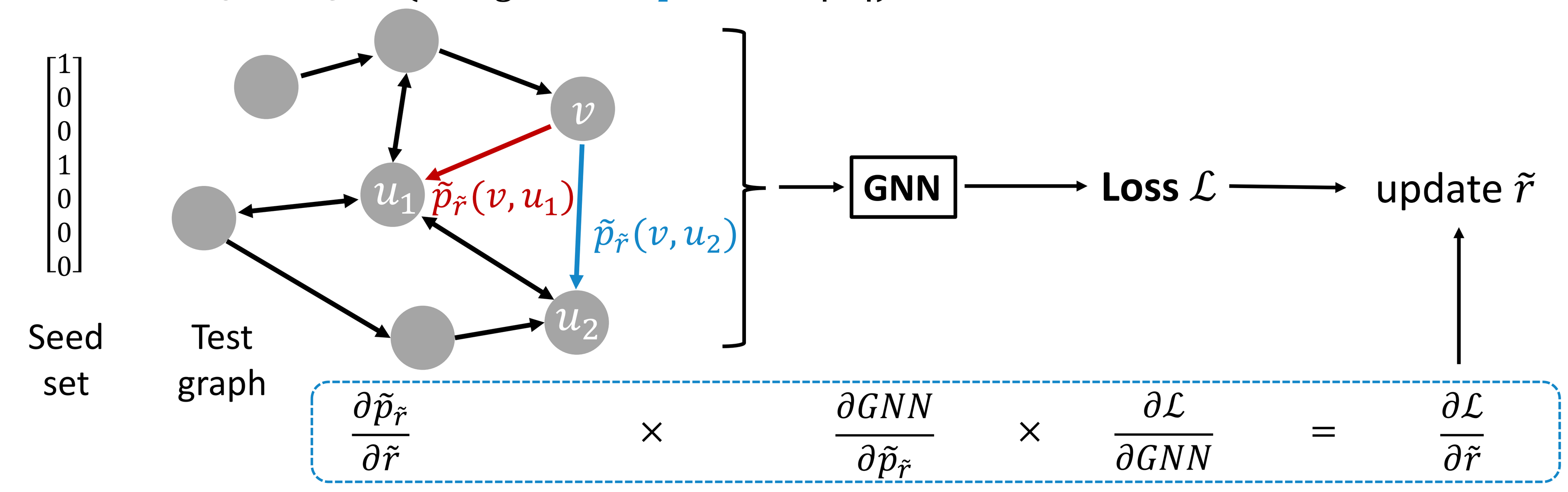
- **Key idea:** ***"Train in advance, estimate quickly"***
- **Procedures:**
  - Train a **GNN** as a surrogate model for influence estimation
  - Iteratively remove an edge whose removal reduces the **GNN output** most, until the budget is exhausted



- **Remaining limitations:** Must estimate influence **individually for each edge removal**
- **Time complexity:** $O(budget \cdot |E|^2)$

DiffIM+(+) with Continuous Relaxation

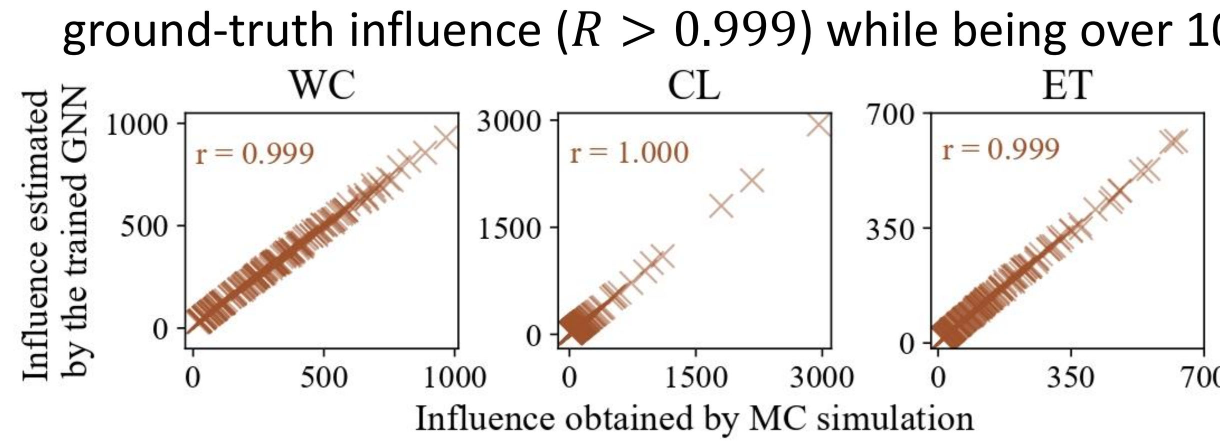
- **Key idea:** ***"Continuous relaxation on edge removal decision"***
  - Binary (whether to remove an edge or not) → Continuous (probability of edge removal)
  - $\Pr[v$ activates u in the output graph$]$ = $\Pr[v$ activates $u] \cdot \Pr[$edge (v,u) is not removed$]$ = $\boxed{p(v,u)\tilde{r}(v,u)}$ → **Modified activation probability $\tilde{p}_{\tilde{r}}$**
  - (Continuous relaxation + GNN) makes the problem ***differentiable***
- **DiffIM+**: Use **gradient descent** to optimize every continuous decision simultaneously
  - **Procedure:** ① Optimize all edge removal probabilities together by ***gradient descent***
    ② Remove an edge with the ***smallest optimized probability***,
    ③ Repeat ① and ② until the budget is exhausted
  - **Loss function:** (influence) + (penalty for the budget) + (penalty for the uncertainty)
  - **Remaining limitation:** Requires **multiple epochs of gradient descent** for each removal
  - **Time complexity:** $O(budget \cdot \# \boldsymbol{epochs} \cdot |E|)$



- **DiffIM++**: Selects edges for removal **instantly** based on **gradients**
  - **Procedure:** ① Compute the gradient of each edge removal decision $\frac{\partial GNN}{\partial \tilde{r}}$ as ***sensitivity***
    ② Remove an edge with the ***largest gradient*** *without further optimization*
    ③ Repeat ① and ② until the budget is exhausted
  - **Time complexity:** $O(budget \cdot |E|)$

Experimental Results

- **Q1. Influence estimation:** GNN estimation achieves **near-perfect correlation** with the ground-truth influence ($R > 0.999$) while being over 100× faster than MC simulations

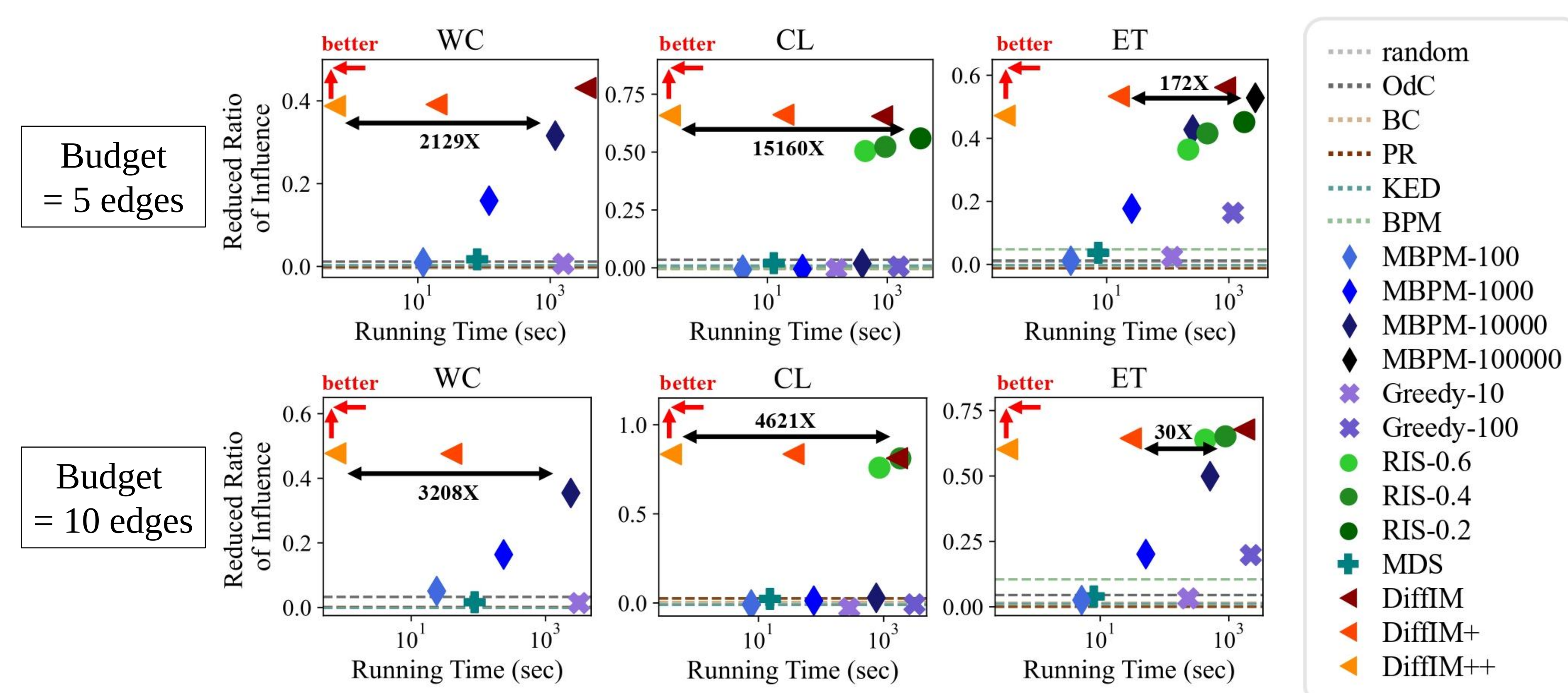


Three real-world social networks

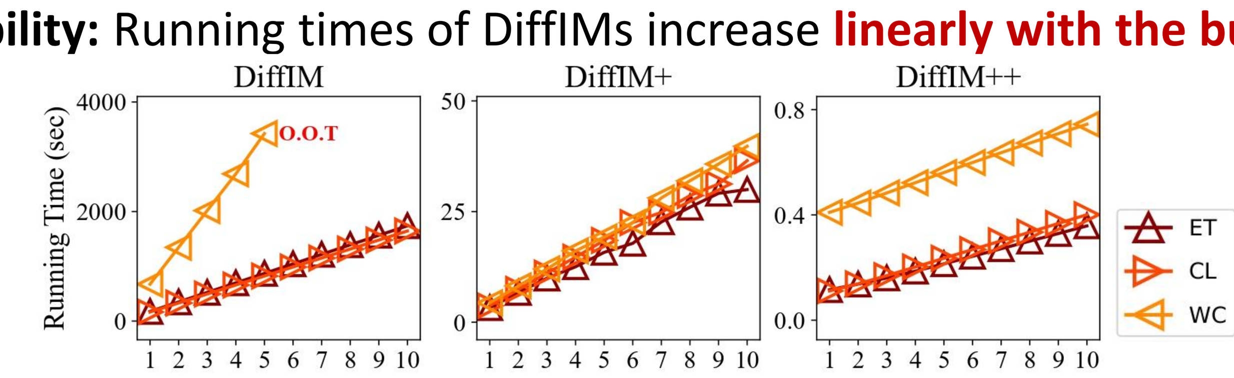
Estimation method	WC	CL	ET
GNN	0.0082	0.0056	0.0056
MC simulation	3.7757	0.8276	0.7817

Estimation time (in seconds)

- **Q2. Performance:** DiffIMs are **faster and more effective** than the best baseline



- **Q3. Scalability:** Running times of DiffIMs increase **linearly with the budget**



- **Q4. Inductivity:** DiffIMs are effective even on **graphs unseen during training**

Method	DiffIM			DiffIM+			DIffIM++		
Dataset	WC	CL	ET	WC	CL	ET	WC	CL	ET
Transductive Inductive (Difference)	0.4311 0.4256 (-1.3%)	0.6547 0.6394 (-2.3%)	0.5613 0.5429 (-3.3%)	0.3914 0.3910 (-0.1%)	0.6614 0.6534 (-1.2%)	0.5332 0.5045 (-5.4%)	0.3876 0.3692 (-4.7%)	0.6583 0.6230 (-5.4%)	0.4718 0.5050 (+7.0%)
Strongest baseline (transductive)	0.3160	0.5591	0.5284	0.3160	0.5591	0.5284	0.3160	0.5591	0.5284