Accumulated Local Effects and Graph Neural Networks

Paulina Kaczyńska

Faculty of MIM, University of Warsaw
IPPT PAN, Polish Academy of Sciences

Accumulated Local Effects for GNNs in Link Prediction

Objective:
Investigate the use of Accumulated Local Effects (ALE) (Apley and Zhou, 2020) for explaining Graph Neural Networks (GNNs) trained for link prediction tasks.

Why is it important? ALE visualizes the impact of the feature's value on the prediction. Most GNN explainability methods can be classified as "feature importance" methods: they highlight how important is the given feature, but not what is the impact of the given value.

Why is ALE Different for GNNs?

During ALE calculation, we modify certain nodes to check how prediction changes.

Tabular Data:

- Data points are independent
- We can modify multiple points simultaneously

Graph Neural Networks:

- GNN layers update node embeddings with information from neighbors passed through edges (**message passing**)
- **Key Challenge:** Simultaneous node modifications during ALE can interfere with GNN predictions

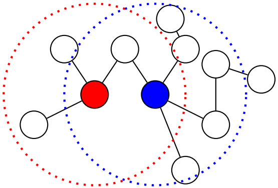
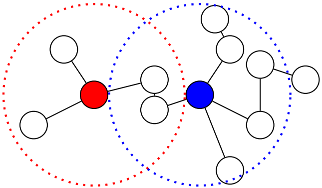


Figure: Interaction between modified nodes in two-layer GNN



Research Question

Two Approaches to ALE for GNNs:

- **Exact (Safe):** Modify one node at a time
  - $\rightarrow$ Accurate but computationally expensive
- **Approximate (Fast):** Modify multiple nodes simultaneously
  - $\rightarrow$ Faster but may interfere with message passing

Research Question:

Can we treat GNNs like tabular data without significant loss of explanation quality?

Accumulated Local Effects

$$g_{S,ALE}(x_S) \equiv \int_{x_{\min,S}}^{x_S} \mathbb{E}\left[\frac{\partial f(x_S, X_C)}{\partial x_S}\middle| x_S = z_S\right] dz_S \qquad (1)$$

Estimation

$$\hat{g}_S(x_S) \equiv \sum_h \frac{1}{n_S(h)} \sum_{\{i: x_{i,S} \in N_S(h)\}} [f(z_{h,S}, x_C) - f(z_{h-1,S}, x_C)] \qquad (2)$$

Additional parameters used while using only a subset of nodes in link prediction tasks:

- parameter *max_bin_size* - how many nodes are modified at once?
- parameter k - how many nodes do we check for a probability of a link with a modified node?

Two algorithms

Input: Model M, Dataset $\mathcal{D}$, Feature index f, Number of bins N
Output: Accumulated Local Effects (ALE) values

Algorithm 1: ALE Exact Version

Initialize empty list ALE;
Divide feature values into N bins;
for *each bin* b_i **do**
    Get nodes in bin b_i;
    for *each node* n_j *in* b_i **do**
        Set feature f of n_j to lower bin edge;
        Compute prediction P_{low};
        Set feature f of n_j to upper bin edge;
        Compute prediction P_{high};
        Compute difference $D = P_{high} - P_{low}$;
        Store D;
    Compute average difference for bin b_i and update ALE;
Return ALE;

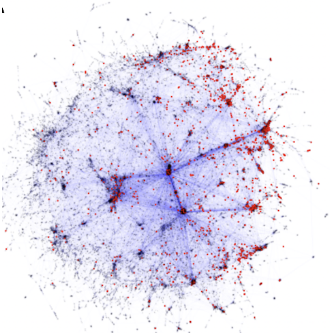
Algorithm 2: ALE Approximate Version

Initialize empty list ALE;
Divide feature values into N bins;
for *each bin* b_i **do**
    Get nodes in bin b_i;
    Set feature f of all nodes in b_i to lower bin edge;
    Compute prediction P_{low};
    Set feature f of all nodes in b_i to upper bin edge;
    Compute prediction P_{high};
    Compute average difference $D = P_{high} - P_{low}$;
    Update ALE with D;
Return ALE;

Dataset 1: AI Research Citations

Network Structure

- **Nodes:** 159,734 AI research papers from the S2ORC corpus
- **Edges:** 227,565 citations
- **Node features:** Number of authors, affiliation details, words from the abstract



Citation patterns between industry and academia

Research Focus

Question: Do Big Tech affiliations influence citation patterns?

- **Explained Variable**: Fraction of authors affiliated with Big Techs.
- **Target:** Likelihood of citation

Dataset 2: Mouse Brain Vasculature

3D Vessel Network

- **Nodes:** 1.66M vessel points
- **Edges:** 2.15M connections
- **Node Features:** Cartesian coordinates of nodes, border flag and brain region classification.

Analysis Goal

Question: How does vessel connectivity change with brain height?

- **Key Feature:** Z-coordinate (height)
- **Target:** Connection probability

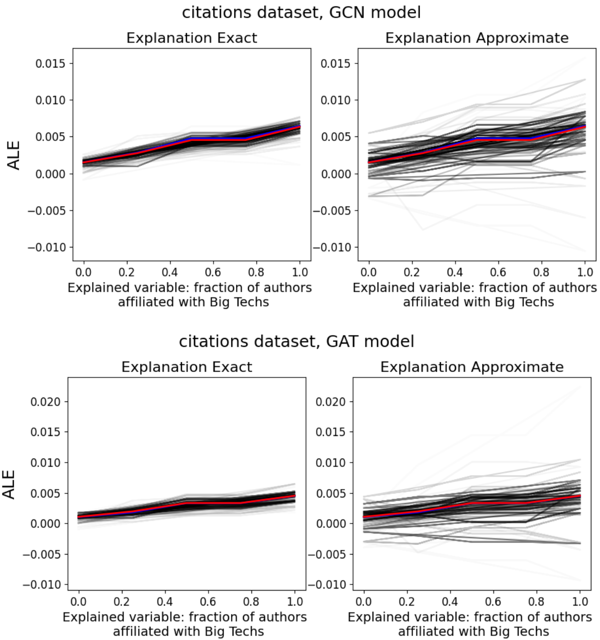
Models

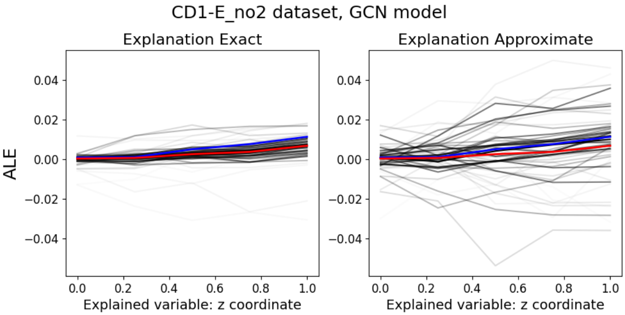
- Link prediction task
- Two 256-dimensional layers
  - Graph Convolutional Network
  - Graph Attention Network
- Citations dataset trained on CPU, CD1-E_no2 on GPU

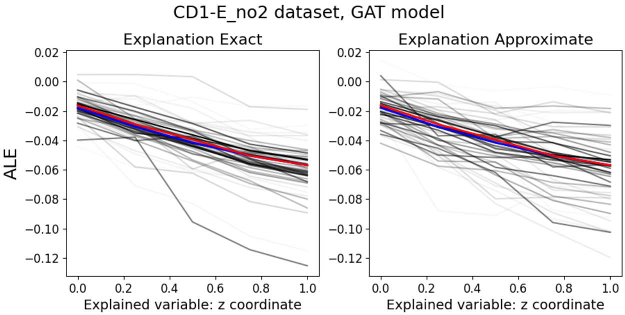
Model		Layers	Layer Dim.	Epochs	F1 Score	AUC ROC
Type	**Dataset**	**(count)**	**(units)**	**(count)**	**(test/val)**	**(test)**
GAT	Citations	2	256	15	0.683	0.635
GCN	Citations	2	256	15	0.703	0.759
GAT	CD1-E_no2	2	256	50	0.741	-
GCN	CD1-E_no2	2	256	50	0.833	-

Table: **Architectural details and metrics of the models for Citations and CD1-E_no2 datasets.**

ALE profiles




CD1-E_no2 dataset, GCN model
Explanation Exact
Explanation Approximate
ALE
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0
Explained variable: z coordinate


CD1-E_no2 dataset, GAT model
Explanation Exact
Explanation Approximate
ALE
0.02
0.00
−0.02
−0.04
−0.06
−0.08
−0.10
−0.12
0.0
0.2
0.4
0.6
0.8
1.0
Explained variable: z coordinate

Statistical Comparison of ALE Curves

χ^2 Test:
Null Hypothesis: Approximate and Exact ALE profiles are sampled from the same distribution.
Rejection threshold: $\chi^2 > 11.07$

Dataset	GCN	GAT
Citations	7.165	5.413
CD1-E_no2	**17.439**	1.296

Conclusion:
Significant difference for GCN on CD1-E_no2 dataset.

Permutation Test:
Null Hypothesis: Approximate and Exact ALE profiles are sampled from the same distribution.
$n = 10,000$ random splits.
Significance level: $\alpha = 0.05$

Dataset	GCN	GAT
Citations	0.407	0.898
CD1-E_no2	0.195	0.155

Conclusion:
No significant differences found.

Permutation test results

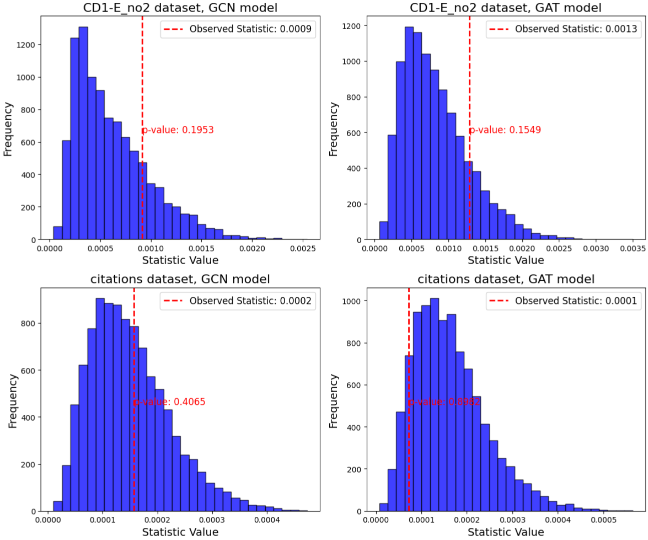
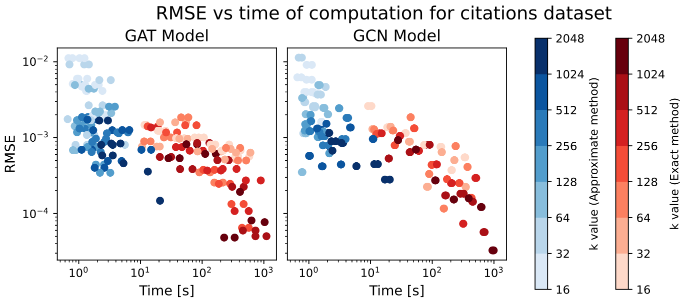
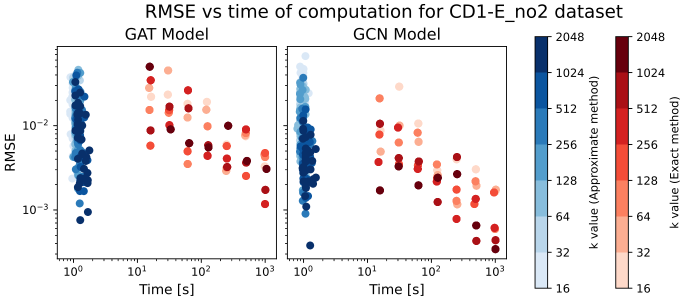


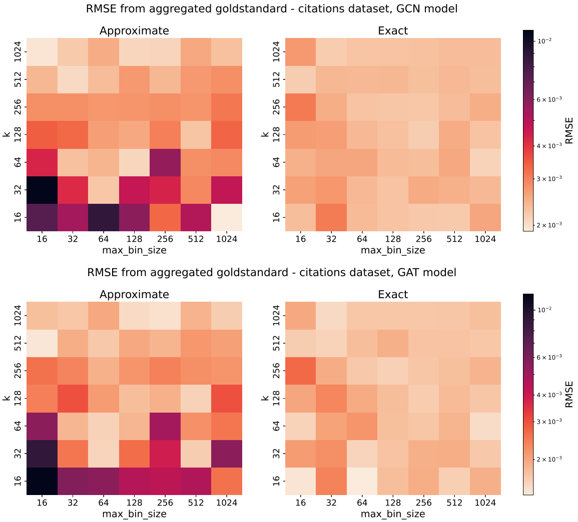
Figure: Histogram of different permutation's test statistic's value

Accuracy and time





Comparison of RMSE for GCN and GAT Models



Key Findings: ALE for Graph Neural Networks

Accuracy vs. Efficiency Trade-off

- Approximate ALE closely matches exact method, while being considerably faster
- But shows higher variance across runs

Performance Factors

Exact Method:

- More modified nodes $\rightarrow$ Better estimation
- Significantly slower computation

Approximate Method:

- More analyzed neighbors $\rightarrow$ Better results
- Balances speed and accuracy

Conclusions

Key Takeaway

Approximate ALE offers a practical approach for explaining GNNs, with manageable accuracy trade-offs for significant speed gains

Future directions

- Averaging multiple ALE profiles obtained with the approximate method seems to produce a profile that is very close to the one obtained with the exact method. Maybe a method combining the two would produce the best results?
- How does it generalize to deeper networks, other tasks like node classification, denser datasets?