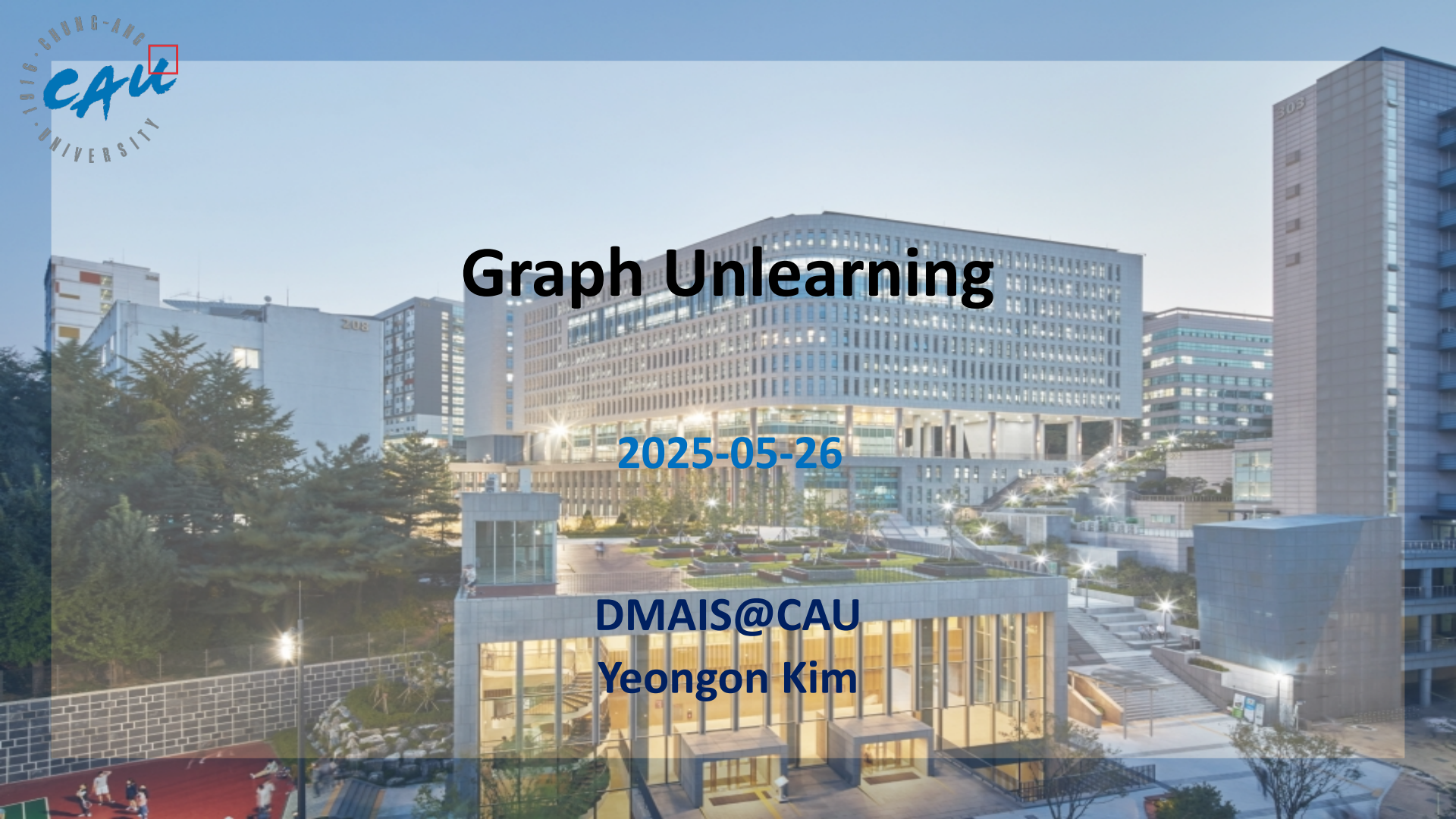


Graph Unlearning

2025-05-26

DMAIS@CAU
Yeongon Kim

INDEX

- Introduction

- Graph Unlearning
  - Graph Eraser
  - Graph Influence Function
  - Gradient Transformation

- Conclusion

- Appendix

Introduction

❖ Machine Unlearning

- Selectively removing the influence of specific data from a trained model without requiring full retraining

❖ Need for Unlearning

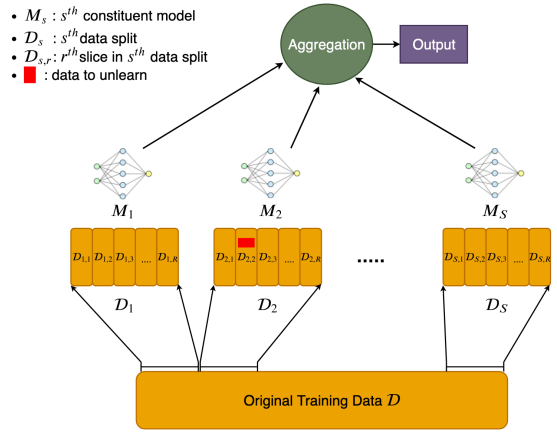
- Simply deleting the raw data is not enough
  - Models internalize and propagate knowledge across layers and parameters
- Retraining from scratch is impractical
  - Large models and datas require excessive time and resources to retrain
- Right to be forgotten
  - Legally mandated right under the European GDPR

Introduction

❖ Machine Unlearning Methods

- **SISA(Sharded, Isolated, Sliced, and Aggregated)**
  - **Structuring training data into independently trained, sequentially updated partitions**

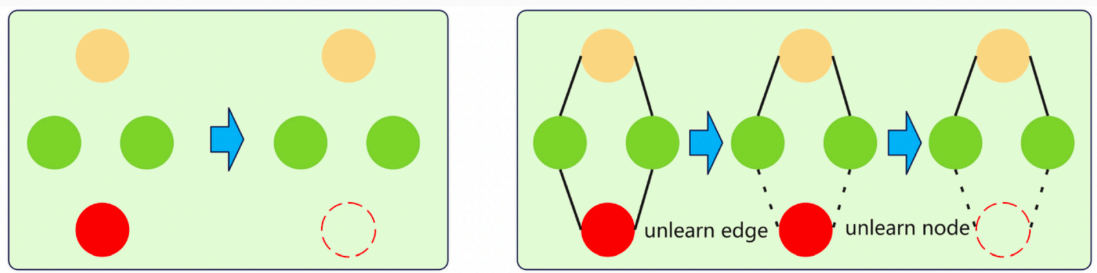
- **Influence Function based Unlearning**
  - **Estimate and remove the impact of a data sample on model parameters**



Introduction

❖ Graph Unlearning

- In graphs, data points are dependent to each other
- Removing a node affects its direct and multi-hop neighbors
- Graph unlearning must consider both the direct effects on model parameters and the indirect effects propagated through the graph structure

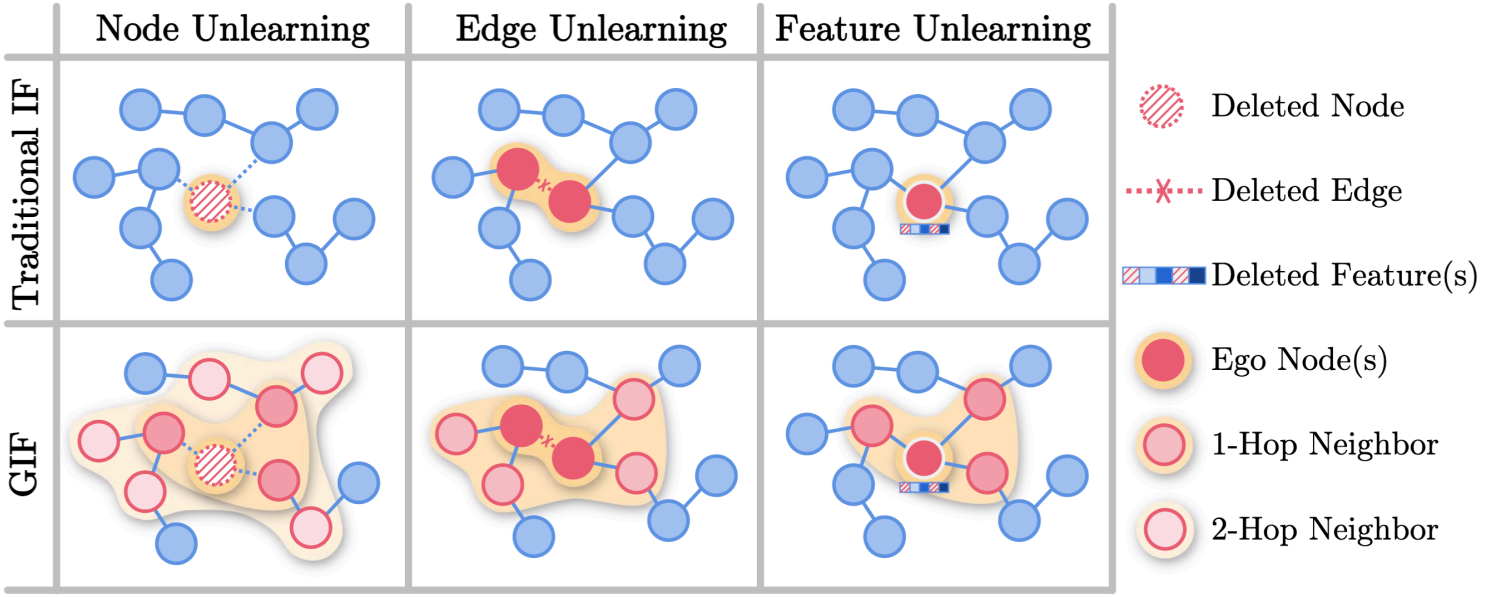


(a) Regular data unlearning

(b) Graph unlearning

Introduction

❖ Graph Unlearning Tasks



Introduction

Graph Unlearning Methods

- Retraining from Scratch
- SISA based Method (e.g. Graph Eraser)
- Influence function based method (e.g. Graph Influence Function)
- Fine tuning based method
- Learning-based method (e.g. Gradient Transformation)

Graph Unlearning

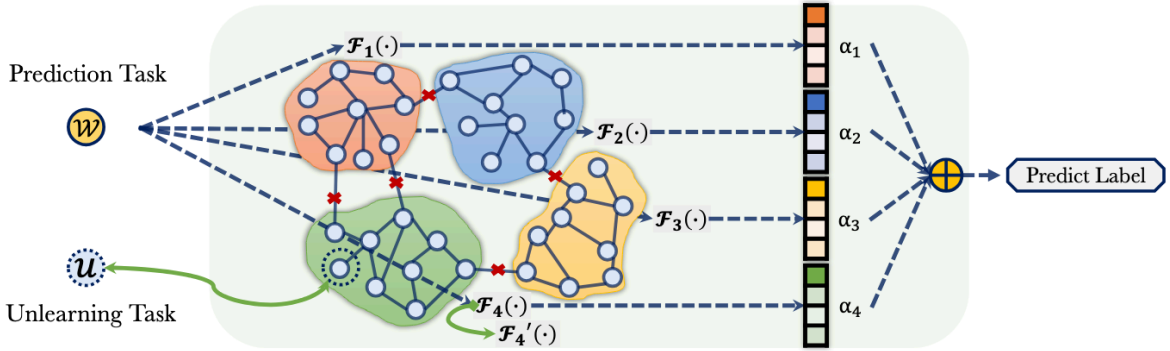
Min Chen, Zhikun Zhang, Tianhao Wang,
Michael Backes, Mathias Humbert, Yang Zhang

CCS 2022

Graph Eraser

❖ Graph Eraser

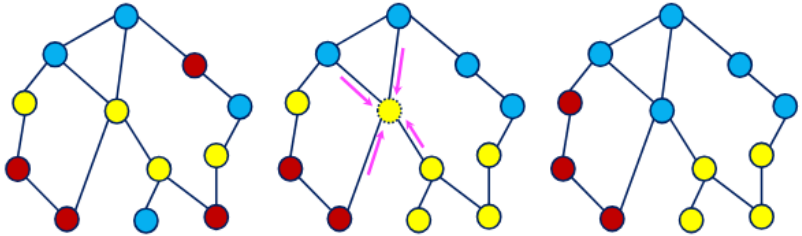
- Partition a graph and trained on separate models
- Results are aggregated to produce the final output
- Introduces two balanced graph partitioning methods and one aggregation strategy
- Balanced graph partitioning is used to prevent a shard becoming too large



Graph Eraser

❖ Balanced Graph Partition

- BLPA(Balanced Label Propagation Algorithm)
  - Assigned node to the shard that contains the majority of its neighbors
  - Each shard has a maximum node capacity(δ)
  - Considers only the structural information



- BEKM(Balanced Embedding K-means)
  - Assigns node embedding to clusters(shards) similar to K-Means
  - Each cluster(shard) is constrained to hold only a limited number of nodes
  - Considers both the structural information and the node features

Graph Eraser

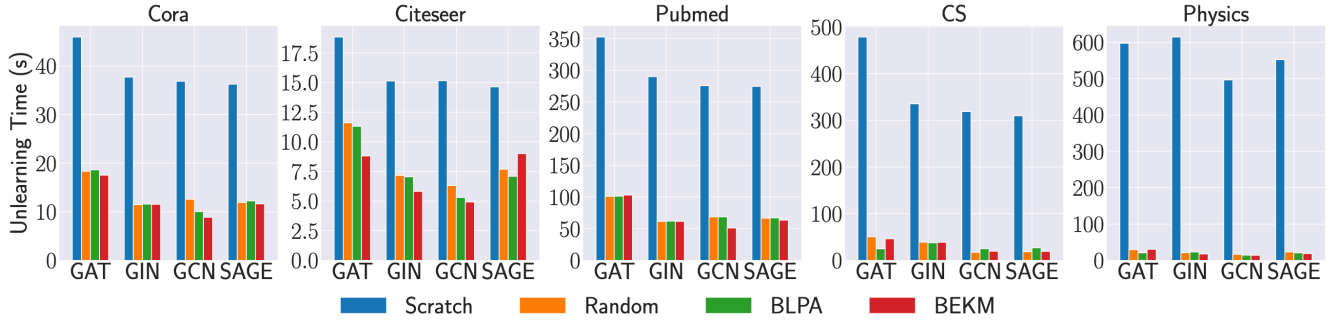
❖ Learning-Based Aggregation(LBAggr)

- Assigns a learnable weight to each shard model
- Improves on the limitation of treating all shards equally in MajAggr and MeanAggr
- α is updated when the unlearned node was part of the sampled training set

$$\min_{\alpha} \mathop{\mathbb{E}}_{w \in \mathcal{G}_o} \left[\mathcal{L} \left(\sum_{i=0}^{m} \alpha_i \cdot \mathcal{F}_i(X_w, \mathcal{N}_w), y \right) \right] + \lambda \sum_{i=0}^{m} ||\alpha_i||$$

Graph Eraser

❖ Experiments



Dataset	Graph Partition Cost			Prediction Cost		Learn Cost of
	Random	BLPA	BEKM	Scratch	Shard	LBAggr
Cora	0.8s	3s	26s	0.002s	0.003s	1.3s
Citeseer	0.5s	2s	20s	0.003s	0.004s	1.5s
Pubmed	1s	20s	240s	0.004s	0.008s	19s
CS	1s	13s	220s	0.004s	0.009s	25s
Physics	1s	40s	480s	0.005s	0.021s	33s

Graph Eraser

❖ Experiments

	Dataset/ Model	Scratch	Random			GraphEraser-BLPA			GraphEraser-BEKM		
			MeanAggr	MajAggr	LBAggr	MeanAggr	MajAggr	LBAggr	MeanAggr	MajAggr	LBAggr
Cora	GAT	0.823 ± 0.006	0.649 ± 0.006	0.638 ± 0.010	**0.706 ± 0.004**	0.356 ± 0.005	0.492 ± 0.009	**0.727 ± 0.009**	0.672 ± 0.004	0.669 ± 0.012	**0.754 ± 0.009**
	GCN	0.739 ± 0.003	0.337 ± 0.006	0.188 ± 0.004	**0.509 ± 0.009**	0.590 ± 0.008	0.319 ± 0.007	**0.676 ± 0.004**	0.390 ± 0.011	0.247 ± 0.012	**0.493 ± 0.006**
	GIN	0.787 ± 0.013	0.760 ± 0.030	0.702 ± 0.033	**0.736 ± 0.021**	0.681 ± 0.039	0.594 ± 0.028	**0.753 ± 0.015**	0.758 ± 0.016	0.742 ± 0.031	**0.801 ± 0.018**
	SAGE	0.824 ± 0.004	0.583 ± 0.009	0.572 ± 0.012	**0.682 ± 0.013**	0.354 ± 0.008	0.486 ± 0.012	**0.684 ± 0.014**	0.673 ± 0.008	0.646 ± 0.010	**0.740 ± 0.013**
Citeseer	GAT	0.691 ± 0.015	0.502 ± 0.012	0.507 ± 0.016	**0.631 ± 0.015**	0.504 ± 0.010	0.486 ± 0.009	**0.676 ± 0.004**	0.744 ± 0.007	0.712 ± 0.010	**0.746 ± 0.006**
	GCN	0.493 ± 0.006	0.263 ± 0.014	0.157 ± 0.011	**0.277 ± 0.009**	0.372 ± 0.006	0.192 ± 0.006	**0.450 ± 0.006**	0.298 ± 0.005	0.129 ± 0.007	**0.332 ± 0.006**
	GIN	0.587 ± 0.031	0.611 ± 0.028	0.540 ± 0.056	**0.626 ± 0.022**	0.451 ± 0.062	0.447 ± 0.032	**0.612 ± 0.026**	0.725 ± 0.016	0.696 ± 0.014	**0.739 ± 0.020**
	SAGE	0.668 ± 0.013	0.519 ± 0.024	0.536 ± 0.026	**0.623 ± 0.014**	0.447 ± 0.007	0.472 ± 0.024	**0.657 ± 0.012**	0.708 ± 0.003	0.710 ± 0.007	**0.716 ± 0.007**
Pubmed	GAT	0.851 ± 0.004	0.852 ± 0.001	0.851 ± 0.002	**0.857 ± 0.002**	0.843 ± 0.002	0.840 ± 0.002	**0.858 ± 0.003**	0.853 ± 0.001	0.852 ± 0.001	**0.860 ± 0.003**
	GCN	0.748 ± 0.017	0.484 ± 0.004	0.207 ± 0.000	**0.551 ± 0.005**	0.644 ± 0.004	0.423 ± 0.011	**0.718 ± 0.010**	0.353 ± 0.003	0.207 ± 0.000	**0.482 ± 0.003**
	GIN	0.837 ± 0.015	0.854 ± 0.003	0.852 ± 0.003	**0.856 ± 0.003**	0.849 ± 0.002	0.843 ± 0.002	**0.855 ± 0.004**	0.859 ± 0.002	0.851 ± 0.010	**0.859 ± 0.003**
	SAGE	0.874 ± 0.003	0.854 ± 0.002	0.852 ± 0.003	**0.857 ± 0.002**	0.841 ± 0.003	0.836 ± 0.003	**0.863 ± 0.002**	0.854 ± 0.002	0.852 ± 0.002	**0.862 ± 0.002**
CS	GAT	0.919 ± 0.004	0.880 ± 0.001	0.877 ± 0.001	**0.882 ± 0.000**	0.664 ± 0.015	0.662 ± 0.009	**0.858 ± 0.004**	0.885 ± 0.001	0.882 ± 0.003	**0.906 ± 0.002**
	GCN	0.903 ± 0.006	0.644 ± 0.002	0.528 ± 0.001	**0.706 ± 0.008**	0.658 ± 0.004	0.440 ± 0.003	**0.750 ± 0.023**	0.620 ± 0.003	0.502 ± 0.003	**0.812 ± 0.012**
	GIN	0.867 ± 0.005	0.856 ± 0.006	0.839 ± 0.004	**0.858 ± 0.005**	0.655 ± 0.024	0.691 ± 0.011	**0.789 ± 0.013**	0.857 ± 0.005	0.844 ± 0.005	**0.891 ± 0.002**
	SAGE	0.932 ± 0.004	0.896 ± 0.005	0.896 ± 0.003	**0.905 ± 0.004**	0.745 ± 0.009	0.679 ± 0.003	**0.886 ± 0.010**	0.904 ± 0.007	0.903 ± 0.001	**0.927 ± 0.002**
Physics	GAT	0.955 ± 0.005	0.917 ± 0.001	0.915 ± 0.001	**0.920 ± 0.002**	0.871 ± 0.032	0.858 ± 0.044	**0.921 ± 0.004**	0.920 ± 0.001	0.917 ± 0.000	**0.925 ± 0.001**
	GCN	0.947 ± 0.002	0.597 ± 0.001	0.533 ± 0.001	**0.747 ± 0.010**	0.817 ± 0.003	0.770 ± 0.001	**0.858 ± 0.008**	0.575 ± 0.003	0.506 ± 0.001	**0.815 ± 0.001**
	GIN	0.934 ± 0.003	0.903 ± 0.002	0.916 ± 0.001	**0.921 ± 0.002**	0.842 ± 0.009	0.840 ± 0.006	**0.907 ± 0.003**	0.924 ± 0.002	0.919 ± 0.001	**0.926 ± 0.001**
	SAGE	0.956 ± 0.005	0.712 ± 0.003	0.717 ± 0.002	**0.823 ± 0.011**	0.905 ± 0.003	0.894 ± 0.003	**0.922 ± 0.001**	0.926 ± 0.003	0.924 ± 0.002	**0.933 ± 0.001**

Dataset	Model	Scratch	Random	BLPA	BEKM
Cora	GAT	0.823 ± 0.005	0.723 ± 0.009	**0.774 ± 0.008**	0.756 ± 0.005
	GCN	0.742 ± 0.004	0.448 ± 0.005	**0.657 ± 0.005**	0.474 ± 0.002
	GIN	0.786 ± 0.011	0.755 ± 0.007	0.762 ± 0.009	**0.768 ± 0.027**
	SAGE	0.827 ± 0.007	0.669 ± 0.005	0.721 ± 0.003	**0.731 ± 0.002**
Citeseer	GAT	0.706 ± 0.003	0.620 ± 0.017	**0.674 ± 0.002**	0.670 ± 0.001
	GCN	0.470 ± 0.005	0.464 ± 0.004	0.532 ± 0.008	**0.571 ± 0.017**
	GIN	0.610 ± 0.019	0.592 ± 0.015	0.632 ± 0.026	**0.736 ± 0.020**
	SAGE	0.667 ± 0.002	0.670 ± 0.012	0.680 ± 0.062	**0.711 ± 0.006**
Pubmed	GAT	0.844 ± 0.003	0.827 ± 0.002	0.848 ± 0.002	**0.854 ± 0.007**
	GCN	0.740 ± 0.001	0.549 ± 0.005	**0.716 ± 0.010**	0.578 ± 0.002
	GIN	0.846 ± 0.015	0.857 ± 0.050	0.865 ± 0.004	**0.859 ± 0.003**
	SAGE	0.873 ± 0.001	0.837 ± 0.002	**0.868 ± 0.002**	0.855 ± 0.002
CS	GAT	0.930 ± 0.004	0.882 ± 0.010	0.847 ± 0.002	**0.896 ± 0.001**
	GCN	0.905 ± 0.006	0.706 ± 0.018	**0.790 ± 0.003**	0.732 ± 0.022
	GIN	0.887 ± 0.005	0.858 ± 0.005	0.789 ± 0.013	**0.861 ± 0.002**
	SAGE	0.953 ± 0.004	0.898 ± 0.009	0.896 ± 0.015	**0.923 ± 0.001**
Physics	GAT	0.956 ± 0.002	0.910 ± 0.003	0.925 ± 0.002	**0.928 ± 0.003**
	GCN	0.942 ± 0.005	0.729 ± 0.013	**0.853 ± 0.007**	0.773 ± 0.002
	GIN	0.939 ± 0.003	0.910 ± 0.005	0.917 ± 0.003	**0.929 ± 0.002**
	SAGE	0.950 ± 0.005	0.817 ± 0.021	0.924 ± 0.001	**0.936 ± 0.001**

- Node classification, Node unlearning(left), Edge unlearning(right)
- F1 scores for unlearning methods and different aggregation methods

Graph Eraser

❖ Conclusion & Future work

- ✓ **First machine unlearning framework GraphEraser in the context of GNNs**
- □ **Shard count enables efficient, balanced partitioning**
- □ **Limited by its reliance on preprocessing and graph partitioning itself**

GIF: A General Graph Unlearning Strategy via Influence Function

Jiancan Wu, Yi Yang, Yuchun Qian,
Yongduo Sui, Xiang Wang, Xiangnan He

WWW 2023

Graph Influence Function

❖ Limitations of GraphEraser

- Cannot handle feature unlearning
- Splitting data into shards will inevitably destroy the connections in samples, especially for graph data, hence hurting the model performance
- Reliance on pre-processing, Model-specific design
  - Each GNN requires a different clustering method

Graph Influence Function

❖ Influence Function

- Estimate how much a single sample influences the model's parameters
- Approximate the full effect of deleting the data by a tiny perturbation
- Also efficiently approximated the Hessian matrix to reduce computational overhead

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}, \quad \mathcal{L} = \sum_{z_i \in \mathcal{D}_0 \setminus \Delta\mathcal{D}} l\left(f_{\mathcal{G}\setminus\Delta\mathcal{G}}(z_i), y_i\right) \qquad \theta_\epsilon = \arg\min_{\theta}(\mathcal{L}_0 + \epsilon\mathcal{L}_{\Delta\mathcal{G}})$$

$$0 \approx \Delta\theta\left(\nabla^2_{\theta_0}\mathcal{L}_0 + \epsilon\nabla^2_{\theta_0}\mathcal{L}_{\Delta\mathcal{G}}\right) + \left(\epsilon\nabla_{\theta_0}\mathcal{L}_{\Delta\mathcal{G}} + \nabla_{\theta_0}\mathcal{L}_0\right)$$

$$\hat{\theta} - \theta_0 = \theta_{\epsilon=-1} - \theta_0 \approx H^{-1}_{\theta_0}\nabla_{\theta_0}\mathcal{L}_{\Delta\mathcal{G}}$$

✦ Assumes that a perturbation in one sample does not affect other samples

Graph Influence Function

❖ Graph Influence Function

- Considers the influence of a sample has on its multi-hop neighbors
- This paper derived a closed-form mathematical formulation of GIF using a one-layer GCN

$$\hat{l}(z_i, y_i) = \begin{cases} l(f_{\mathcal{G}}(z_i), y_i), & z_i \in \Delta\mathcal{G} \\ l(f_{\mathcal{G}}(z_i), y_i) - l(f_{\mathcal{G}\setminus\Delta\mathcal{G}}(z_i), y_i), & z_i \text{ is influenced by } \Delta\mathcal{G} \\ 0, & \text{other nodes} \end{cases}$$

- *For Node Unlearning request* $\Delta\mathcal{G} = \{\mathcal{V}^{(rm)}, \varnothing, \varnothing\}$, *the influenced region is* $\mathcal{N}_k(\mathcal{V}^{(rm)}) = \bigcup_{e_i \in \mathcal{V}^{(rm)}} \mathcal{N}_{k+1}(e_i)$;
- *For Edge Unlearning request* $\Delta\mathcal{G} = \{\varnothing, \mathcal{E}^{(rm)}, \varnothing\}$, *the influenced region is* $\mathcal{N}_k(\mathcal{E}^{(rm)}) = \bigcup_{z_i \in \mathcal{E}^{(rm)}} \mathcal{N}_k(z_i)$;
- *For Feature Unlearning request* $\Delta\mathcal{G} = \{\varnothing, \varnothing, \mathcal{X}^{(rm)}\}$, *the influenced region is* $\mathcal{N}_k(\mathcal{X}^{(rm)}) = \bigcup_{z_i \sim \mathcal{X}^{(rm)}} \mathcal{N}_k(z_i)$, *where* $z_i \sim \mathcal{X}^{(rm)}$ *indicates that the feature of node* z_i *is revoked.*

- *For Node Unlearning tasks,*

$$\Delta\theta = H_{\theta_0}^{-1} \sum_{z_i \in \mathcal{N}_k(\mathcal{V}^{(rm)}) \cup \mathcal{V}^{(rm)}} \nabla_{\theta_0} l\left(f_{\mathcal{G}}(z_i), y_i\right) - H_{\theta_0}^{-1} \sum_{z_i \in \mathcal{N}_k(\mathcal{V}^{(rm)})} \nabla_{\theta_0} l\left(f_{\mathcal{G}}(z_i; \mathcal{G} \setminus \Delta\mathcal{G}), y_i\right).$$

- *For Edge Unlearning tasks,*

$$\Delta\theta = H_{\theta_0}^{-1} \sum_{z_i \in \mathcal{N}_k(\mathcal{E}^{(rm)})} \nabla_{\theta_0} l\left(f_{\mathcal{G}}(z_i), y_i\right) - H_{\theta_0}^{-1} \sum_{z_i \in \mathcal{N}_k(\mathcal{E}^{(rm)})} \nabla_{\theta_0} l\left(f_{\mathcal{G}}(z_i; \mathcal{G} \setminus \Delta\mathcal{G}), y_i\right).$$

- *For Feature Unlearning tasks,*

$$\Delta\theta = H_{\theta_0}^{-1} \sum_{z_i \sim \mathcal{N}_k(\mathcal{X}^{(rm)}) \cup \mathcal{X}^{(rm)}} \nabla_{\theta_0} l\left(f_{\mathcal{G}}(z_i), y_i\right) - H_{\theta_0}^{-1} \sum_{z_i \sim \mathcal{N}_k(\mathcal{X}^{(rm)}) \cup \mathcal{X}^{(rm)}} \nabla_{\theta_0} l\left(f_{\mathcal{G}}(z_i; \mathcal{G} \setminus \Delta\mathcal{G}), y_i\right).$$

Graph Influence Function

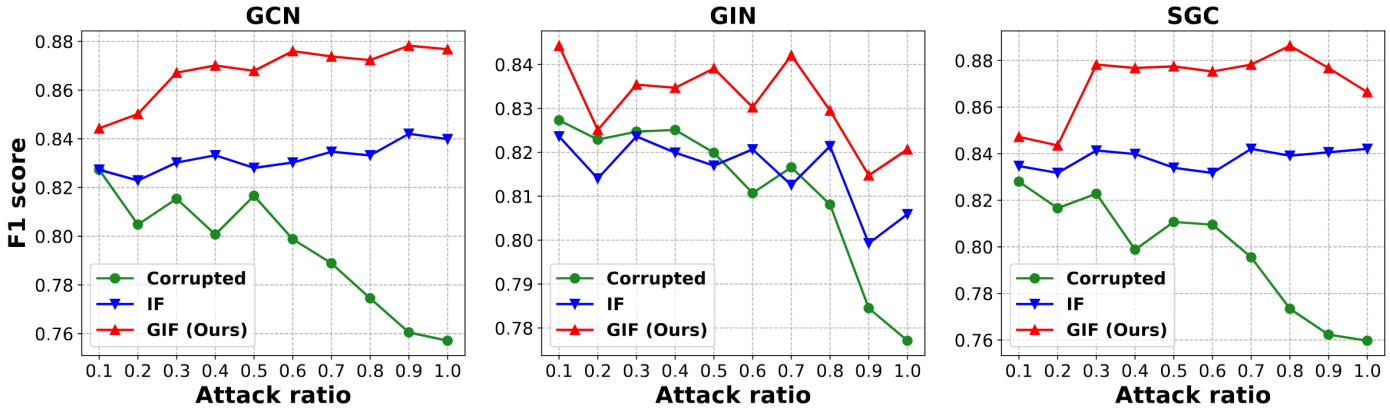
❖ Experiments

Model		Dataset					
		Cora		Citeseer		CS	
Backbone	Strategy	F1 score	RT (second)	F1 score	RT (second)	F1 score	RT (second)
GCN	Retrain	0.8210±0.0055	6.33	0.7318±0.0096	7.52	0.9126±0.0055	55.95
	LPA	0.6790±0.0001	1.35	0.5556±0.0001	1.69	0.7732±0.0001	3.04
	Kmeans	0.5535±0.0001	1.89	0.5045±0.0001	1.56	0.7754±0.0001	9.97
	GIF	**0.8218±0.0066**	**0.16**	**0.6925±0.0060**	**0.13**	**0.9137±0.0016**	**0.26**
GAT	Retrain	0.8804±0.0060	15.72	0.7643±0.0049	19.50	0.9305±0.0011	110.39
	LPA	0.3432±0.0001	3.04	0.6997±0.0001	3.92	0.7650±0.0001	6.43
	Kmeans	0.6900±0.0001	3.19	0.7628±0.0001	3.41	0.8794±0.0001	17.36
	GIF	**0.8649±0.0072**	**0.86**	**0.7663±0.0072**	**0.59**	**0.9325±0.0015**	**1.02**
SGC	Retrain	0.8236±0.0142	6.63	0.7132±0.0091	7.16	0.9165±0.0045	58.12
	LPA	0.3247±0.0001	2.03	0.3934±0.0001	1.70	0.5267±0.0001	3.08
	Kmeans	0.3690±0.0001	1.41	0.3874±0.0001	1.56	0.6532±0.0001	10.59
	GIF	**0.8129±0.0110**	**0.12**	**0.6892±0.0082**	**0.12**	**0.9164±0.0051**	**0.24**
GIN	Retrain	0.8051±0.0144	8.48	0.7294±0.0207	9.94	0.8822±0.0074	70.38
	LPA	0.6605±0.0001	2.65	0.6096±0.0001	2.21	0.6510±0.0001	3.82
	Kmeans	0.7491±0.0001	2.53	0.6517±0.0001	2.11	0.8336±0.0001	10.29
	GIF	**0.8059±0.0234**	**0.48**	**0.7315±0.0185**	**0.48**	**0.8884±0.0083**	**0.57**

- Edge unlearning with 5% edges deleted from the original graph.
- GIF achieve a better trade-off between unlearning efficiency and model utility

Graph Influence Function

❖ Experiments



- Model utility is not enough to measure unlearning
- Add adversarial edges to the graph, measure F1 source recovery after their removal

Graph Influence Function

❖ Experiments

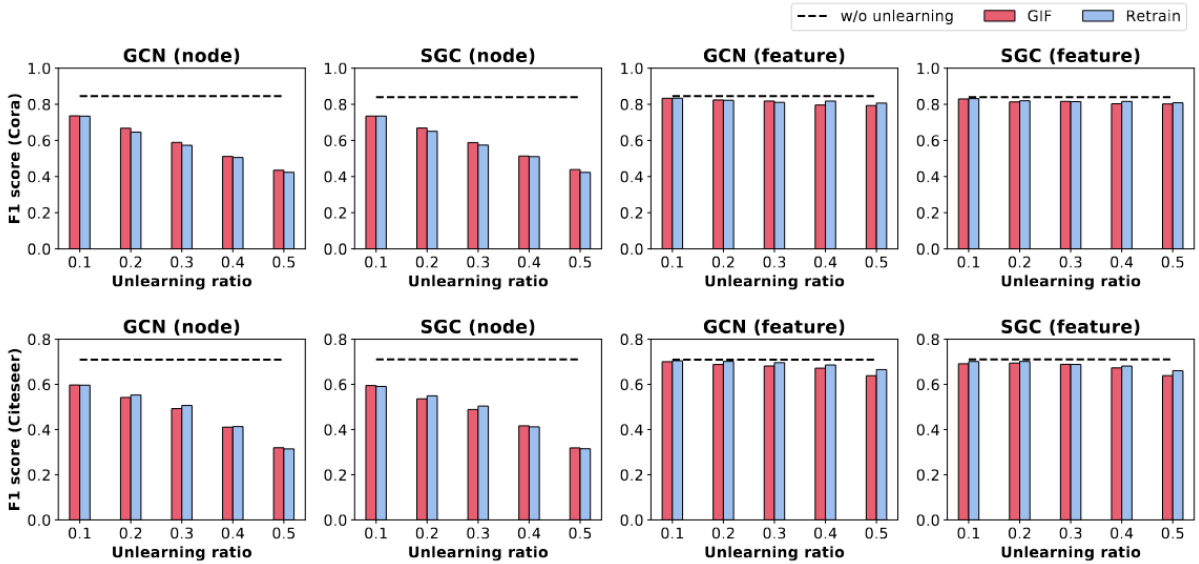


Figure 3: Impact of unlearning ratio ρ on node unlearning tasks and feature unlearning tasks.

- GIF performs well when unlearning a large amount of data

Graph Influence Function

❖ Conclusion & Future work

- ✓ **More general and powerful graph unlearning algorithm for GNN models**
- □ **First attempts to interpret the black box of graph unlearning with mathematical proof**
- □ **Propose more comprehensive evaluation criteria and major tasks for graph unlearning**
- □ **The computation of the Hessian matrix remains burdensome for large models**

Dynamic Graph Unlearning: A General and Efficient Post-Processing Method via Gradient Transformation

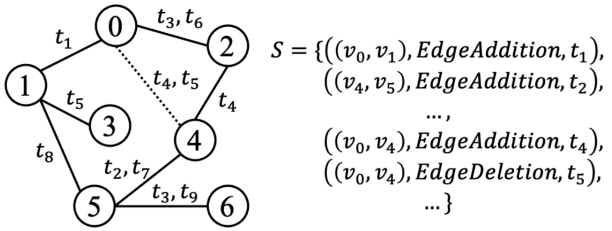
He Zhang, Bang Wu, Xiangwen Yang,
Xingliang Yuan, Xiaoning Liu, Xun Yi

WWW 2025

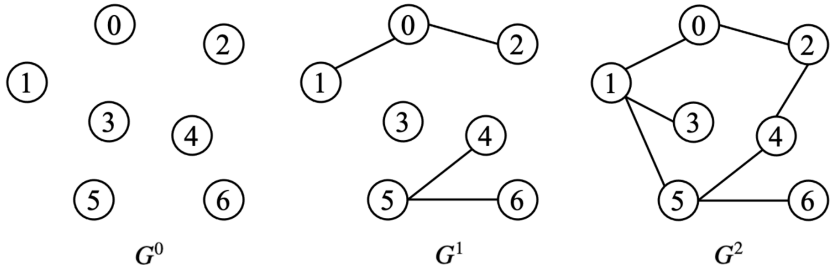
Gradient Transformation

❖ Dynamic Graph

- A graph continuously evolves with node, edge and feature changes over time
- Continuous-time Dynamic Graphs
  - Individual timestamped events occurring at arbitrary real-valued times
- Discrete-time Dynamic Graphs
  - Represented by snapshots at regular time intervals



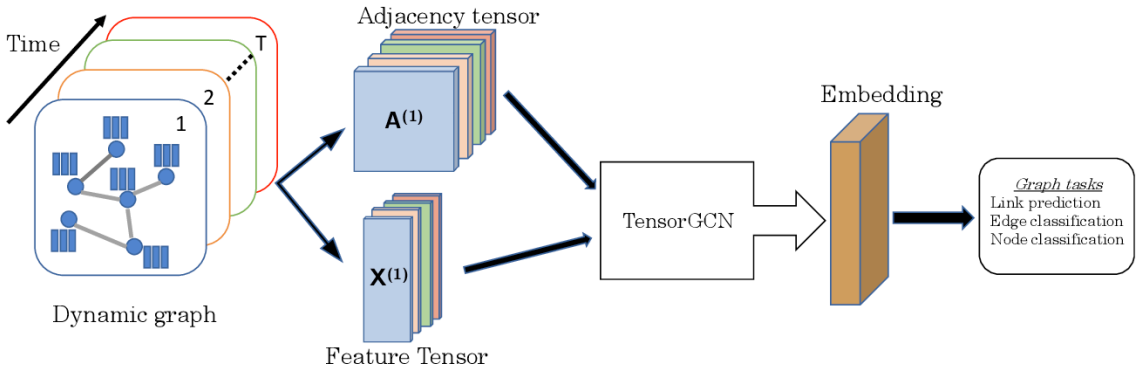
(b) Continuous-Time Dynamic Graph $\mathcal{G} = \{G, S\}$



(c) Discrete-Time Dynamic Graph $\mathcal{G} = \{G^0, G^1, G^2\}$

Gradient Transformation

- **Dynamic Graphs Neural Networks**
  - Designed to learn additional and complex temporal changes in dynamic graphs
  - Common tasks on dynamic graphs are node classification and link prediction
  - Used to make predictions in the future



Gradient Transformation

- **Fine-tuning methods (e.g. GraphGuard)**
  - Optimizes the parameter of the target model in a post-processing manner
  - Relatively light using standard gradient descent
  - G~r: retained nodes, G~p: unlearning target nodes

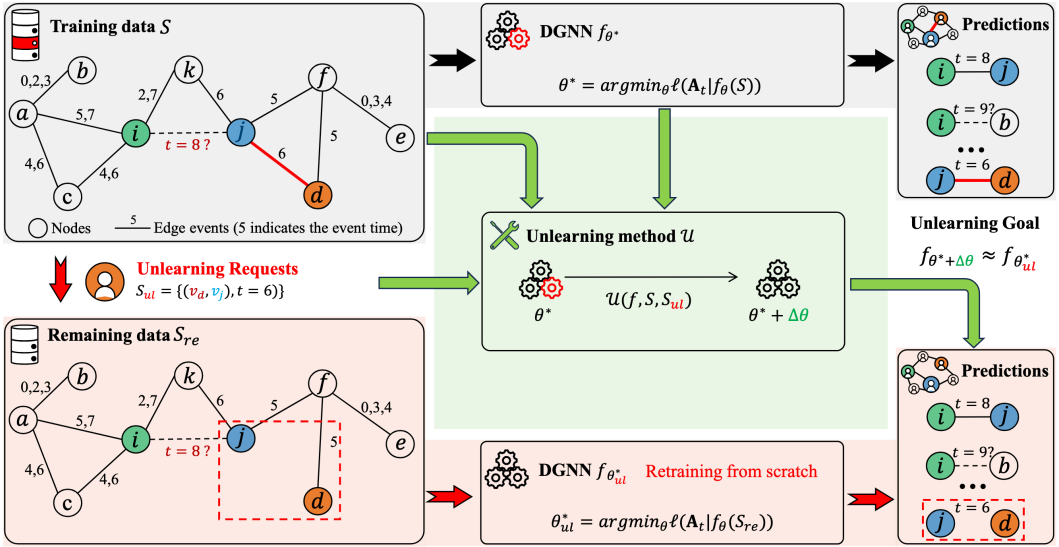
$$\min_{\theta} \; L\big(f_\theta(\tilde{G}_r)\big) \; - \; \alpha \, L\big(f_\theta(\tilde{G}_p)\big)$$

- **Limitation of previous works**
  - Reliance on pre-processing, Model-specific design
  - Impractical resource requirements, Limited to static graphs
    - DyGFomer(4.146MB fp parameters) with GIF needs 4.298 TB of storage space
  - Fine-tuning method is prone to overfitting unlearning samples

Gradient Transformation

❖ Learning-based Unlearning

- A separate two-layer MLP is trained with dedicated loss functions to perform unlearning
- Unlearning goal is to make minimum difference with retraining from scratch



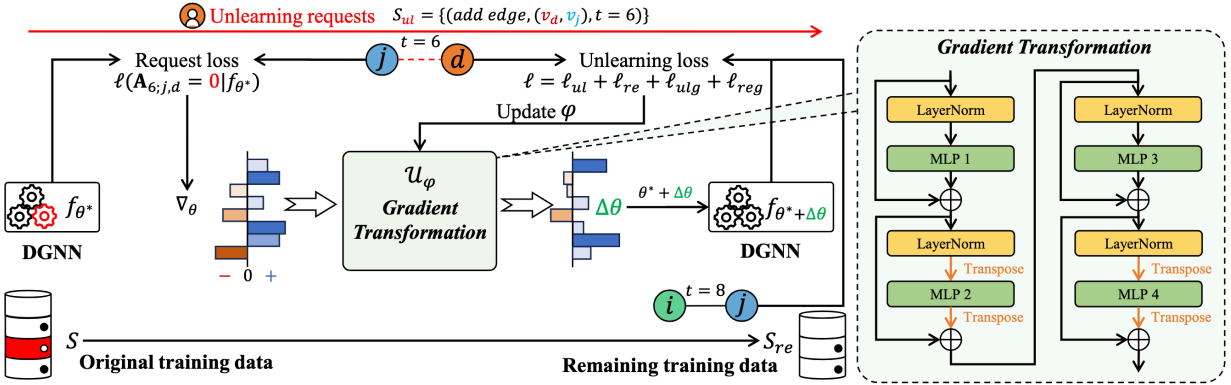
$$\min_{\phi} \text{dis}\left(f(\theta^* + \Delta\theta),\ f_{\theta^*_{ul}}\right)$$

$$\Delta\theta = U_\phi\left(A_f(S),\ S,\ S_{ul}\right)$$

Gradient Transformation

❖ Unlearning Loss Function

- ℓ_ul : Unlearning Loss, ℓ_re : Remaining Data Loss
- ℓ_reg : To avoid the performance drop caused by unlearning
- ℓ_ulg : To avoid the overfitting caused by unlearning



$$\nabla\theta = \frac{d}{d\theta}\ell(\widehat{Y_{ul}} \mid f_{\theta^*}, S_{ul})$$

$$\ell_{ul} = \ell(\widehat{Y_{ul}} \mid f_{\theta^*+\Delta\theta}, S_{ul})$$

$$\ell_{re} = \ell(Y_{re} \mid f_{\theta^*+\Delta\theta}, S_{re})$$

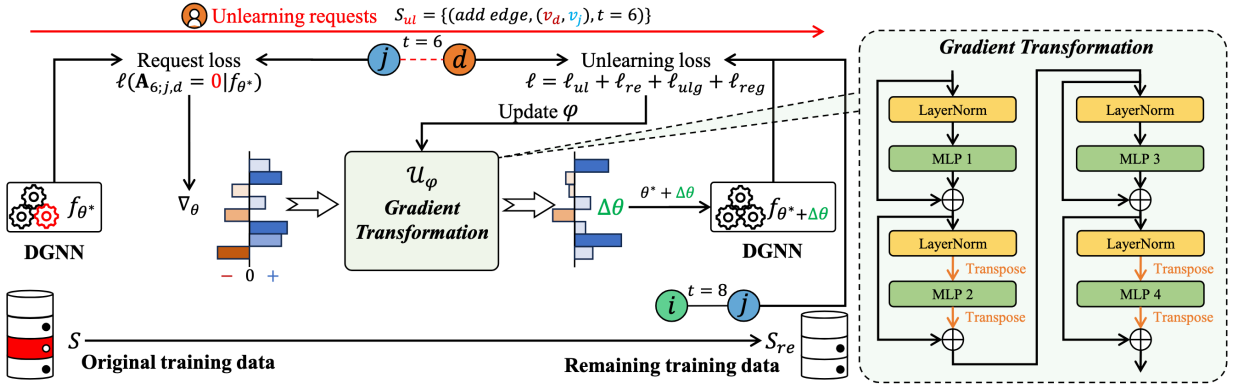
$$\ell_{ulg} = d(\widehat{Y_{ul}}, Y_{ul}^c)$$

$$\ell_{reg} = d(Y_{re}, Y_{val})$$

Gradient Transformation

❖ Gradient Transformation Model

- Gradient transformation function by a two-layer MLP trained just for unlearning tasks
- Instead of computing all four loss terms each time, gradient transformation function is used



$$\mathbf{H}^{(1)}_{\text{tok}} = \mathbf{H}_{\text{in}} + \mathbf{W}^{(2)}_{\text{tok}}\,\text{GeLU}\left(\mathbf{W}^{(1)}_{\text{tok}}\,\text{LN}(\mathbf{H}_{\text{in}})\right),$$

$$\mathbf{H}^{(1)}_{\text{cha}} = \mathbf{H}^{(1)}_{\text{tok}} + \text{GeLU}\left(\text{LN}(\mathbf{H}^{(1)}_{\text{tok}})\mathbf{W}^{(1)}_{\text{cha}}\right)\mathbf{W}^{(2)}_{\text{cha}},$$

$$\mathbf{H}^{(2)}_{\text{tok}} = \mathbf{H}^{(1)}_{\text{cha}} + \mathbf{W}^{(4)}_{\text{tok}}\,\text{GeLU}\left(\mathbf{W}^{(3)}_{\text{tok}}\,\text{LN}(\mathbf{H}^{(1)}_{\text{cha}})\right),$$

$$\mathbf{H}^{(2)}_{\text{cha}} = \mathbf{H}^{(2)}_{\text{tok}} + \text{GeLU}\left(\text{LN}(\mathbf{H}^{(2)}_{\text{tok}})\mathbf{W}^{(3)}_{\text{cha}}\right)\mathbf{W}^{(4)}_{\text{cha}},$$

Gradient Transformation

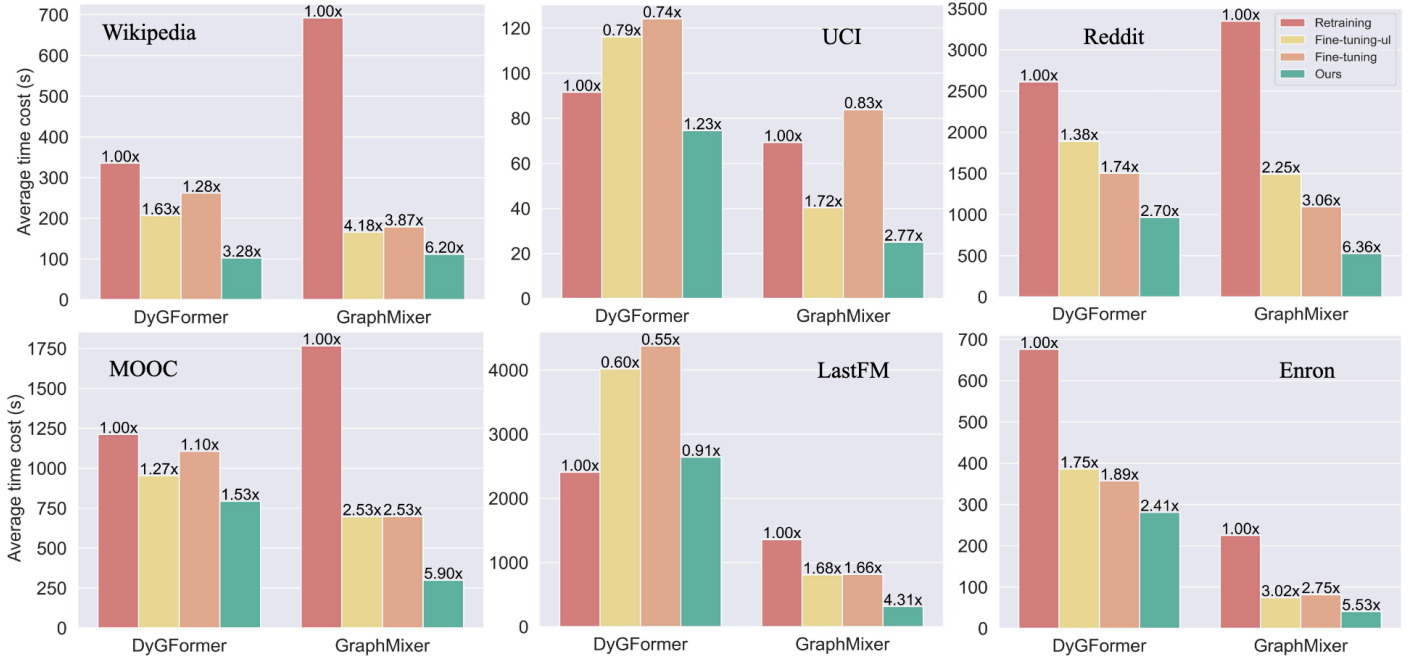
❖ Experiments

Table 3: Acc **comparison between our method and baseline methods.**

Datasets	Methods	DyGFormer		GraphMixer	
		Acc(S_{re})/ΔAcc ↑	Acc(S_{ul})/|ΔAcc| ↓	Acc(S_{re})/ΔAcc ↑	Acc(S_{ul})/|ΔAcc| ↓
Wikipedia	Retraining	0.9507 ± 0.0004	0.1470 ± 0.0435	0.9105 ± 0.0028	0.2050 ± 0.0078
	Fine-tuning-*ul*	0.7541 ± 0.1463	**0.1574 ± 0.1005**	0.9112 ± 0.0056	**0.1666 ± 0.0146**
	Fine-tuning	0.9062 ± 0.0545	0.3962 ± 0.0797	0.9117 ± 0.0056	0.1640 ± 0.0111
	Ours	**0.9529 ± 0.0008**	0.1773 ± 0.0330	**0.9307 ± 0.0016**	0.1339 ± 0.0145
UCI	Retraining	0.8458 ± 0.0008	0.2056 ± 0.0097	0.8685 ± 0.0003	0.1361 ± 0.0080
	Fine-tuning-*ul*	0.7552 ± 0.1427	0.3850 ± 0.3515	0.8705 ± 0.0036	0.1246 ± 0.0066
	Fine-tuning	0.7572 ± 0.1437	0.4006 ± 0.3408	0.8710 ± 0.0034	0.1235 ± 0.0080
	Ours	**0.8484 ± 0.0020**	**0.2057 ± 0.0028**	**0.8737 ± 0.0019**	**0.1258 ± 0.0055**
Reddit	Retraining	0.9460 ± 0.0012	0.0540 ± 0.0033	0.9009 ± 0.0003	0.0477 ± 0.0028
	Fine-tuning-*ul*	0.6665 ± 0.0817	0.3116 ± 0.2221	0.8874 ± 0.0058	0.3361 ± 0.0512
	Fine-tuning	0.8592 ± 0.0104	0.2433 ± 0.0727	0.8879 ± 0.0060	0.3342 ± 0.0516
	Ours	**0.9487 ± 0.0006**	**0.0476 ± 0.0014**	**0.9326 ± 0.0005**	**0.0212 ± 0.0021**
MOOC	Retraining	0.8176 ± 0.0124	0.0452 ± 0.0389	0.8171 ± 0.0033	0.1240 ± 0.0068
	Fine-tuning-*ul*	0.6090 ± 0.1116	0.4962 ± 0.3865	0.8161 ± 0.0053	0.0552 ± 0.0110
	Fine-tuning	0.7510 ± 0.0252	0.4804 ± 0.0595	0.8161 ± 0.0053	0.0552 ± 0.0110
	Ours	**0.7950 ± 0.0036**	**0.0472 ± 0.0164**	**0.8222 ± 0.0026**	**0.0630 ± 0.0130**
LastFM	Retraining	0.8738 ± 0.0049	0.3110 ± 0.1099	0.6583 ± 0.0006	0.2918 ± 0.0075
	Fine-tuning-*ul*	0.5578 ± 0.0660	0.5322 ± 0.2697	0.6593 ± 0.0024	0.3059 ± 0.0071
	Fine-tuning	0.7590 ± 0.0163	0.9103 ± 0.0232	0.6597 ± 0.0024	**0.2980 ± 0.0030**
	Ours	**0.8198 ± 0.0224**	**0.2834 ± 0.1714**	**0.6632 ± 0.0015**	0.3009 ± 0.0080
Enron	Retraining	0.9321 ± 0.0119	0.2101 ± 0.0399	0.8007 ± 0.0009	0.2866 ± 0.0215
	Fine-tuning-*ul*	0.5012 ± 0.0007	0.9953 ± 0.0059	0.8109 ± 0.0015	**0.2938 ± 0.0569**
	Fine-tuning	**0.9015 ± 0.0329**	0.2759 ± 0.1047	**0.8110 ± 0.0015**	0.2771 ± 0.0498
	Ours	0.8226 ± 0.1652	**0.2324 ± 0.2934**	0.8100 ± 0.0012	0.2944 ± 0.0341

Gradient Transformation

❖ Experiments



Gradient Transformation

❖ Conclusion & Future work

- ✓ **The first study of the unlearning problem in the context of dynamic graphs**
- □ **Effective, efficient, model-agnostic, and post-processing method**

Appendix

Dataset $\mathcal{D}$	Model $\mathcal{F}$	BLPA-based GraphEraser-BLPA	BLPA-based BLPA-LP	BEKM-based GraphEraser-BEKM	BEKM-based BEKM-Hungarian	Minimum Edge Cut METIS
Cora	**GAT**	0.727 ± 0.009	0.712 ± 0.006	**0.754 ± 0.009**	0.740 ± 0.006	0.683 ± 0.007
	GCN	**0.676 ± 0.004**	0.668 ± 0.020	0.531 ± 0.009	0.552 ± 0.005	0.458 ± 0.010
	GIN	0.753 ± 0.015	0.722 ± 0.029	**0.801 ± 0.018**	0.795 ± 0.016	0.703 ± 0.020
	SAGE	0.684 ± 0.014	0.708 ± 0.002	**0.740 ± 0.013**	0.739 ± 0.005	0.694 ± 0.008
Citeseer	**GAT**	0.688 ± 0.005	0.590 ± 0.009	**0.738 ± 0.006**	0.737 ± 0.003	0.615 ± 0.002
	GCN	**0.516 ± 0.004**	0.504 ± 0.022	0.417 ± 0.018	0.397 ± 0.023	0.457 ± 0.006
	GIN	0.597 ± 0.021	0.589 ± 0.041	**0.678 ± 0.072**	0.655 ± 0.059	0.574 ± 0.064
	SAGE	0.642 ± 0.005	0.682 ± 0.007	**0.743 ± 0.002**	0.734 ± 0.002	0.677 ± 0.004
Pubmed	**GAT**	0.858 ± 0.003	0.857 ± 0.001	**0.860 ± 0.003**	0.857 ± 0.003	0.841 ± 0.001
	GCN	**0.718 ± 0.010**	0.709 ± 0.004	0.659 ± 0.020	0.628 ± 0.034	0.650 ± 0.018
	GIN	0.855 ± 0.004	0.854 ± 0.001	**0.859 ± 0.003**	0.853 ± 0.001	0.836 ± 0.001
	SAGE	0.863 ± 0.002	0.857 ± 0.003	**0.862 ± 0.002**	0.858 ± 0.00	0.849 ± 0.003
CS	**GAT**	0.858 ± 0.004	0.862 ± 0.003	**0.906 ± 0.002**	0.901 ± 0.003	0.891 ± 0.013
	GCN	0.750 ± 0.023	0.745 ± 0.004	**0.812 ± 0.012**	0.806 ± 0.007	0.782 ± 0.021
	GIN	0.789 ± 0.013	0.786 ± 0.003	**0.891 ± 0.002**	0.883 ± 0.007	0.862 ± 0.002
	SAGE	0.886 ± 0.010	0.889 ± 0.023	**0.927 ± 0.002**	0.922 ± 0.002	0.906 ± 0.004
Physics	**GAT**	0.921 ± 0.004	0.918 ± 0.004	**0.925 ± 0.001**	0.923 ± 0.001	0.918 ± 0.002
	GCN	**0.858 ± 0.008**	0.856 ± 0.005	0.815 ± 0.001	0.808 ± 0.001	0.810 ± 0.001
	GIN	0.907 ± 0.003	0.897 ± 0.011	**0.926 ± 0.001**	0.923 ± 0.002	0.895 ± 0.003
	SAGE	0.922 ± 0.001	0.913 ± 0.002	**0.933 ± 0.001**	0.931 ± 0.001	0.911 ± 0.005

Dataset $\mathcal{D}$	BLPA-based GraphEraser	BLPA-based LP	BEKM-based GraphEraser	BEKM-based Hungarian	Minimum Edge Cut METIS
Cora	**3s**	179s	**26s**	817s	4s
Citeseer	**2s**	30s	**20s**	1,309s	3s
Pubmed	**20s**	301s	**240s**	174,684s	21s
CS	**13s**	705s	**220s**	174,498s	15s
Physics	**40s**	2,351s	**480s**	948,790s	58s