1 Affected Range Validation

To validate that degree-based and degree-free GNNs exhibit different ranges of affected neighbors in GU, we conduct experiments on the Cora dataset. Specifically, we evaluate five common GNN architectures (GCN, SGC, GAT, GIN, SAGE) with two layers. Each model's parameters are randomly initialized and remain fixed throughout the experiment.

We simulate various GU tasks by randomly deleting graph elements at different levels (nodes, edges, or features). The original graph $\mathcal{G}$ and the remaining graph $\mathcal{G}\backslash\Delta\mathcal{G}$ are then put into the GNN to obtain node representations $\mathbf{H}$ (before deletion) and $\mathbf{H}'$ (after deletion), respectively. By comparing $\mathbf{H}$ and $\mathbf{H}'$, we can infer the range of nodes whose message-passing process is affected by the deleted elements.

From the results in Tables 1–3, we can find that: (1) In node and edge unlearning, the affected range of degree-based GNNs extends one hop further than that of degree-free GNNs. (2) In feature unlearning, since the graph structure remains unchanged, both types of GNNs exhibit the same affected range. (3) The farther a node is from the deleted elements, the smaller the influence it experiences. Specifically, marginal neighbors (e.g.,3-hop neighbors in Table 1, 2-hop neighbors in Table 2) are less affected than other affected neighbors. These findings align with our main text analysis, validating the rationality of our neighbor selection strategy.

2 Dataset Descriptions

We evaluate AGU on seven graph datasets: Cora, Citeseer, Pubmed, Coauthor-CS, Amazon-Photo, Amazon-Computers, and Flickr. Table 4 summarizes the statistics of these datasets, and detailed descriptions of each dataset are provided below.

- **Cora**, **CiteSeer**, and **PubMed** [Yang *et al.*, 2016] are three citation network datasets representing undirected graphs, where nodes correspond to research papers and edges denote citation relationships. Node features are represented as binary word vectors, with each element indicating the presence (1) or absence (0) of a specific word in the paper. The node labels correspond to the research topics of the papers.
- **Amazon Photo** and **Amazon Computers** [Shchur *et al.*, 2018] are segments of the Amazon co-purchase graph, where nodes represent products and edges indicate frequent co-purchases. Each product is characterized by bag-of-words features extracted from customer reviews. The task associated with these graphs is to classify products into their respective categories.
- **Coauthor CS** [Shchur *et al.*, 2018] is a co-authorship graph derived from the Microsoft Academic Graph. In this graph, nodes represent authors, and an edge connects two authors if they have co-authored a paper. Node features correspond to paper keywords extracted from each author's publications, while class labels denote the author's primary field of study.
- **Flickr** [Zeng *et al.*, 2019] is a dataset derived from SNAP, where Flickr data is used to construct an undirected graph. Nodes represent images, and edges connect images that share common properties, such as geographic location, gallery, or comments. Each node is represented by a 500-dimensional bag-of-words feature vector extracted from the images. The labels are obtained by manually merging 81 tags into 7 classes.

Table 1: Average cosine similarity (%) of node representations before and after randomly deleting 10% nodes. '**k-hop**' refers to nodes that are *k* hops away from the deleted nodes, while 'same' indicates identical representations.

	Bone	**1-hop**	**2-hop**	**3-hop**	**Other**
Degree-based	GCN	92.46	98.71	99.94	same
	SGC	95.79	99.64	99.99	same
Degree-free	GAT	96.49	99.62	same	same
	GIN	95.15	99.19	same	same
	SAGE	99.99	≈100	same	same

Table 2: Average cosine similarity (%) of node representations before and after randomly deleting 10% edges. '**k-hop**' refers to nodes that are k hops away from the ego nodes, while 'same' indicates identical representations.

	Bone	**Ego nodes**	**1-hop**	**2-hop**	**Rest**
Degree-based	GCN	97.56	99.65	99.99	same
	SGC	96.66	99.60	99.99	same
Degree-free	GAT	97.21	99.63	same	same
	GIN	96.74	99.27	same	same
	SAGE	≈100	≈100	same	same

Table 3: Average cosine similarity (%) of node representations before and after deleting features of 10% randomly selected nodes. '**k-hop**' refers to nodes that are k hops away from those with deleted features, while 'same' indicates identical representations.

	Bone	**1-hop**	**2-hop**	**3-hop**	**Rest**
Degree-based	GCN	97.44	99.51	same	same
	SGC	97.48	99.56	same	same
Degree-free	GAT	97.61	99.62	same	same
	GIN	98.34	99.24	same	same
	SAGE	≈100	≈100	same	same

3 Backbone GNNs

We select five widely used GNNs as backbone models to simulate scenarios where unlearning requests arise during training. Specifically, we consider two **degree-based GNNs:**

Table 4: Statistics of Datasets.

Dataset	**#Nodes**	**#Edges**	**#Features**	**#Classes**
Cora	2,708	5,429	1,433	7
Citeseer	3,327	4,732	3,703	6
PubMed	19,717	44,338	500	3
Photo	7,487	119,043	745	8
Computers	13,381	245,778	767	10
CS	18,333	81,894	6,805	15
Flickr	89,250	899,756	500	7

SGC and GCN, along with three **degree-free GNNs:** GAT, GIN, and SAGE. A brief overview of each backbone GNN is provided below.

- **GCN** [Kipf and Welling, 2016] applies a layer-wise propagation rule based on a first-order approximation of spectral graph convolutions, enabling efficient learning on graph-structured data.
- **SGC** [Wu *et al.*, 2019] simplifies GCN by removing nonlinear transformations between layers, collapsing them into a single linear operation to enhance computational efficiency.
- **GAT** [Velickovic *et al.*, 2017] employs a self-attention mechanism to adaptively weigh the influence of neighbors when aggregating features.
- **GIN** [Xu *et al.*, 2018] introduces a multi-layer perceptron (MLP)-based architecture that matches the representational power of the Weisfeiler-Lehman graph isomorphism test.
- **SAGE** [Hamilton *et al.*, 2017] is an inductive framework that efficiently generates node representations by aggregating attribute information from neighboring nodes.

4 Graph Unlearning Baselines

We provide descriptions of GU methods that have been proposed in recent years:

- **GNNDelete** [Cheng *et al.*, 2023] introduces a model-agnostic layer-wise operator to optimize the influence of topology in graph unlearning requests.
- **GIF** [Wu *et al.*, 2023a] leverages influence functions to approximate parameter changes induced by an unlearning request. It then updates the model parameters accordingly to facilitate graph unlearning.
- **IDEA** [Dong *et al.*, 2024] analyzes objective differences before and after removing specific information from the graph and provides a theoretical guarantee to certify the effectiveness of unlearning.
- **MEGU** [Li *et al.*, 2024b] unifies predictive capability and forgetting ability within a single optimization framework, enabling mutual reinforcement between these objectives.
- **UTU** [Tan *et al.*, 2024] removes the influence of forgotten edges by unlinking them, thereby obstructing corresponding message-passing paths in the GNN during inference.
- **ETR** [Yang *et al.*, 2024] proposes a neighborhood-aware parameter editing strategy to mitigate the impact of unlearned samples and introduces a subgraph-based approximation method to estimate gradients on the remaining data.
- **Cognac** [Koli. *et al.*, 2024] refines the representations of affected neighbors through contrastive fine-tuning and mitigates potential errors in deletion set labels via gradient ascent.

Table 5: Performance improvement of baselines. Unlearn ratio=3%.

Bone	Method	Cora F1	Cora Time	Citeseer F1	Citeseer Time
SGC	MEGU	82.5±.3	0.153	73.9±.2	0.186
	+MNF	82.7±.3	0.131	73.9±.2	0.175
	+ANS	83.0±.2	0.129	74.0±.2	0.177
	+Both	**83.3±.5**	**0.124**	**74.1±.2**	**0.166**
	Delete	81.8±.6	0.886	72.4±.3	0.917
	+MNF	82.1±.4	0.757	72.3±.2	0.811
	+ANS	82.3±.2	0.745	72.6±.4	0.806
	+Both	**82.4±.5**	**0.753**	**72.9±.1**	**0.774**
	ETR	82.4±.4	0.015	74.0±.2	0.018
	+MNF	82.5±.3	0.014	73.9±.1	0.016
	+ANS	82.9±.1	0.013	73.9±.2	0.016
	+Both	**83.2±.6**	**0.012**	**74.1±.1**	**0.014**
GIN	MEGU	85.5±.3	0.179	73.3±.6	0.158
	+ANN	85.6±.4	0.157	73.4±.5	0.147
	+ANS	**85.9±.4**	**0.151**	**73.6±.5**	**0.132**
	Delete	81.2±.5	1.075	71.5±.4	1.039
	+ANN	81.4±.4	0.938	71.3±.6	0.934
	+ANS	**82.3±.1**	**0.842**	**71.7±.1**	**0.792**
	ETR	85.9±.3	0.021	73.1±.5	0.027
	+ANN	86.0±.9	0.019	73.4±.6	0.024
	+ANS	**86.1±.2**	**0.017**	**73.5±.8**	**0.022**

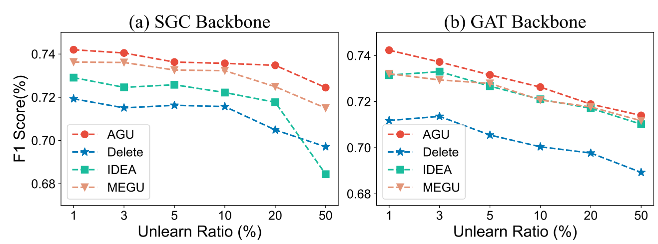

Figure 1: Node unlearning with different unlearn ratios on Citeseer.

5 Experimental Environment

The experiments are conducted on a machine with an Intel(R) Xeon(R) 8352V CPU (2.10GHz) and a single NVIDIA GeForce RTX 4090 GPU with 24GB memory. The system runs Ubuntu 20.04.5, with software versions including Python 3.10, PyTorch 2.1.2, and CUDA 11.8.

6 Additional Experiments

Strategy Generalizability Investigation: In addition to the experiments presented in the main text, we conduct additional experiments using different GNN backbones and unlearn ratios to evaluate the generalizability of our neighbor selection strategies: marginal neighbor filtering (MNS) and affected neighbor selection (ANS). As shown in Table 5, when employing SGC and GIN backbones with a 3% unlearn ratio, incorporating MNF and ANS consistently improves both the effectiveness and efficiency of existing GU methods.

Unlearn Ratio Investigation: In our experimental setup of the main text, we use a default configuration where 5% of graph elements are chosen as unlearning elements. To comprehensively evaluate AGU's performance across different unlearning scales, we present additional experimental results

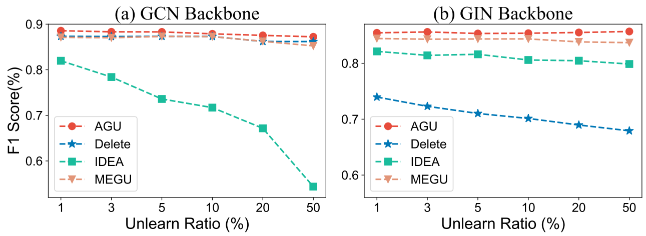

Figure 2: Edge unlearning with different unlearn ratios on PubMed.

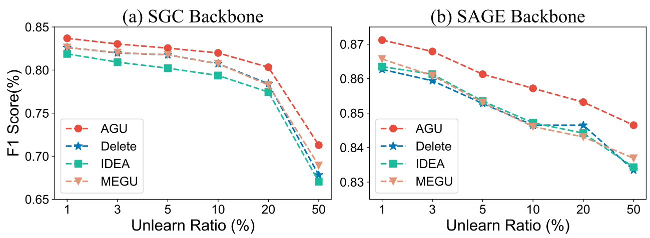

Figure 3: Feature unlearning with different unlearn ratios on Cora.

in Figures 1-3. The results show that AGU consistently outperforms other baselines across different unlearn ratios. Notably, as the unlearn ratio increases, all four GU methods experience a performance decline due to the higher proportion of forgotten data, which negatively impacts model predictions. Despite this inevitable deterioration, AGU exhibits superior robustness compared to other methods under the same unlearning scenarios. These findings highlight AGU's effectiveness in addressing GU tasks across various graph levels and unlearn ratios.