GRAPH UNLEARNING: A REVIEW

Anwar Said
Vanderbilt University,
TN, USA
anwar.said@vanderbilt.edu

Tyler Derr
Vanderbilt University,
TN, USA
tyler.derr@vanderbilt.edu

Mudassir Shabbir
Vanderbilt University,
TN, USA
mudassir.shabbir@vanderbilt.edu

Waseem Abbas
University of Texas at Dallas,
TX, USA
waseem.abbas@utdalls.edu

Xenofon Koutsoukos
Vanderbilt University
TN, USA
xenofon.koutsoukos@vanderbilt.edu

ABSTRACT

Graph unlearning emerges as a crucial advancement in the pursuit of responsible AI, providing the means to remove sensitive data traces from trained models, thereby upholding the right to be forgotten. It is evident that graph machine learning exhibits sensitivity to data privacy and adversarial attacks, necessitating the application of graph unlearning techniques to address these concerns effectively. In this comprehensive survey paper, we present the first systematic review of graph unlearning approaches, encompassing a diverse array of methodologies and offering a detailed taxonomy and up-to-date literature overview to facilitate the understanding of researchers new to this field. Additionally, we establish the vital connections between graph unlearning and differential privacy, augmenting our understanding of the relevance of privacy-preserving techniques in this context. To ensure clarity, we provide lucid explanations of the fundamental concepts and evaluation measures used in graph unlearning, catering to a broader audience with varying levels of expertise. Delving into potential applications, we explore the versatility of graph unlearning across various domains, including but not limited to social networks, adversarial settings, and resource-constrained environments like the Internet of Things (IoT), illustrating its potential impact in safeguarding data privacy and enhancing AI systems' robustness. Finally, we shed light on promising research directions, encouraging further progress and innovation within the domain of graph unlearning. By laying a solid foundation and fostering continued progress, this survey seeks to inspire researchers to further advance the field of graph unlearning, thereby instilling confidence in the ethical growth of AI systems and reinforcing the responsible application of machine learning techniques in various domains.

***K*eywords** Graph Unlearning · Graph Neural Networks · Differential Privacy · Graph Embeddings · Graph Machine Learning · Machine Unlearning · Data Privacy · Adversarial Attacks

1 Introduction

In the ever-evolving landscape of artificial intelligence (AI) and data-driven decision-making, machine learning (ML) has emerged as a transformative force, empowering systems to learn from vast amounts of information and make intelligent predictions. However, as our digital world evolves, so do the ethical and privacy challenges it presents [1]. Enter the groundbreaking concept of machine unlearning—an awe-inspiring technological frontier designed to harmonize the power of machine learning with individual data rights and data privacy regulations [2, 3]. Machine

unlearning represents an extraordinary leap forward in the pursuit of responsible AI, allowing us to gracefully erase the footprints of sensitive data from trained models, empowering individuals to exercise their *right to be forgotten*, and instilling confidence in AI's ethical evolution [3]. Unlearning is indispensable in various applications. For instance, consider a company has trained a graph machine learning model on a social network dataset to make personalized friend recommendations for its users. The model is designed to analyze the network's structure, user interactions, and interests to suggest potential new friends. Let's say a user, Alice, decides to exercise her right to be forgotten and requests the company to remove her data from the model's training dataset. The company, being privacy-conscious, acknowledges Alice's request and decides to perform unlearning on the trained model to ensure her data is no longer used for recommendations. The unlearning process involves updating the model to eliminate any impact of Alice's data on the recommendations [2]. In the past few years, significant efforts have been taken in machine unlearning, resulting in several comprehensive reviews [4, 5]. Despite these efforts, the field of Graph Machine Learning (GML) has remained relatively unexplored in terms of such reviews. Consequently, this comprehensive review is dedicated to shedding light on the latest advancements exclusively within the domain of GML.

Graphs are powerful mathematical structures used to represent and analyze complex relationships and interactions between various entities [6]. They provide a compact and intuitive representation of data, making them invaluable in disciplines such as social networks, in which nodes represent individuals and edges represent friendships, biological networks, including protein-protein interactions, gene regulatory networks, and brain networks [7]. Moreover, transportation systems, recommendation engines, and supply chains, as well as innumerable other systems, are suitable candidates for graph-based modeling because they contain multiple interconnected components [8, 9]. By leveraging the power of graphs, we obtain valuable insights into the structures and behaviors of these complex systems, which leads to improved decision-making, analysis, and forecasting [10].

Learning on graphs, also known as GML or Graph Representation Learning (GRL), has emerged as a flourishing area of research that focuses on developing algorithms capable of extracting meaningful patterns and knowledge from graph-structured data [10, 11]. GML presents unique challenges as conventional machine learning methods are often not directly applicable to graph-structured data due to its inherent complexity and irregularity. GML models, however, have shown remarkable versatility in addressing a wide range of problems [12]. They excel in tasks such as node classification, where the goal is to predict the class or category of nodes in a graph, link prediction, which involves forecasting potential connections between nodes and graph classification, which involves predicting the class or category of the entire graph [13]. Additionally, graph-based models are adept at community detection [14], identifying densely connected groups of nodes, and anomaly detection, where they can pinpoint unusual patterns or outliers in the network. By leveraging the rich structural information within graphs, these models offer novel solutions to real-world challenges and have the potential to revolutionize fields like drug discovery, fraud detection, and social network analysis, smart city policing among others [15, 16].

Machine unlearning is a novel concept for ensuring *right to be forgotten* that refers to the process of selectively forgetting, or deleting specific data points or instances from the model [4]. Unlearning permits the exclusion of specific data from the training set of a model, making it more adaptable to evolving privacy regulations. Graph unlearning is an extension of machine unlearning designed specifically for graph-structured data [17]. It involves the complete removal of all traces of removed data such as node and edges from a graph, including all of its connections. In other terms, the objective of graph unlearning is to remove specific nodes and edges among others settings (see section 2) from the graph while preserving its overall structure. This level of data removal is significantly more complicated than traditional machine unlearning because it requires disentangling entities that are interconnected [18].

The complexity of graph-structured data poses a challenge for unlearning in graphs. Relationships between nodes and edges are intricate in graphs, forming a dense web of interconnected information [19]. To unravel the impact of forgotten data while conserving the graph's structure, sophisticated algorithms are required. These algorithms must be able to untangle these intricate connections without compromising the graph's functionality [20]. In terms of data privacy and moral machine learning practices, graph unlearning is of the utmost relevance. As data privacy regulations and the rights of individuals to control their data become increasingly important, the ability to comprehensively remove sensitive information from graph-structured data becomes a crucial ethical safeguard. Ensuring the effective and complete eradication of forgotten data from graphs gives individuals more control over their information, instills confidence in machine learning systems, and upholds ethical standards for the responsible management of data [17].

In the past few years, significant efforts have been made in the design of effective graph unlearning strategies, indicating a growing awareness of the significance of addressing data privacy and individual rights in the domain of GRL [21, 22, 23]. These works have yielded optimistic advancements, demonstrating the researchers' and practitioners' commitment to tackling the complexities of graph unlearning. Despite these advancements, it is essential to recognize that the discipline is still relatively young and continues to evolve rapidly. As graph unlearning encounters the unique obstacles posed by interconnected networks, continued effort and creative solutions are required to overcome these obstacles. In light of the

rapidly changing landscape of graph unlearning and the urgent need to resolve data privacy concerns in interconnected networks, we present this comprehensive survey as a crucial step in gaining a better understanding of the field. This survey seeks to summarize, analyze, and classify existing graph unlearning techniques, shedding light on their merits, limitations, and practical implications. By collecting and analyzing these strategies, we hope to provide researchers and practitioners with a clear map for navigating this new domain. In addition, we envision this survey as a catalyst for future progress, outlining potential research and development avenues that will influence the future of graph unlearning.

1.1 Contributions

The field of graph unlearning has recently captivated researchers, witnessing a number of innovative contributions focused on confronting its challenges head-on. Despite this growing interest, a critical void remains in the existing literature – the absence of a comprehensive review that sheds light on these cutting-edge approaches and provides a detailed and systematic analysis along with potential open research directions. Recognizing this critical opportunity, we present the inaugural review paper on graph unlearning. We offer the following contributions.

- We propose a comprehensive taxonomy that categorizes graph unlearning approaches into three distinct groups: exact unlearning, approximate unlearning, and differential privacy, each exhibiting unique characteristics and diverse applications.
- We establish a vital connection between differential privacy and unlearning in the graph domain, presenting an in-depth analysis of all relevant approaches.
- We explore potential applications of graph unlearning across various fields and delve into intricate details for each case.
- We provide a summary of existing open source models and data to help practitioners and inspired researchers.
- Lastly, we emphasize multiple open research directions that demand further investigation, driving the pursuit of novel frontiers in this captivating domain.

1.2 Motivation

In the rapidly evolving landscape of machine learning, security and privacy have emerged as crucial factors requiring the utmost consideration. As machine learning models are increasingly used to process and analyze sensitive data, ensuring user privacy and protecting against potential security vulnerabilities has become a pressing necessity [24]. The potential dangers posed by malicious attacks, data breaches, and unauthorized access to personal data have spurred a growing interest in the development of machine learning techniques that can help protect privacy. addressing these challenges is crucial for nurturing public trust, promoting responsible data usage, and harnessing the full potential of machine learning for the betterment of society [4, 25].

In recognition of the significance of privacy rights in the digital era, several countries have taken legislative measures to enforce the *right to be forgotten*. This legal principle affords individuals the right to request the removal or erasure of their personal data from any online platforms, databases and ML models, ensuring their data is no longer publicly accessible or retained beyond the necessary purpose. Prominent examples of legislation focused on safeguarding individuals' digital privacy and empowering them with greater control over their personal data in the online realm include the European Union's General Data Protection Regulation (GDPR) [26], enacted in May 2018. The GDPR stands as a pioneering framework for data protection, setting high standards for data privacy, and promoting responsible data handling practices. Additionally, other regions and countries have also implemented similar regulations to address the pressing concerns of data privacy. For instance, the California Consumer Privacy Act (CCPA) [27] in California, the Personal Information Protection and Electronic Documents Act (PIPEDA) [28] in Canada, and the Brazilian General Data Protection Law (LGPD) [29] in Brazil exemplify significant legislative efforts to ensure privacy rights and data sovereignty. These legislative initiatives underscore the growing global awareness of privacy rights and emphasize the importance of secure and privacy-conscious approaches in machine learning practices in today's interconnected society [17, 18].

The concept of *the right to be forgotten* has emerged as a central ethical consideration in the context of machine learning. In this context, a model provider's duty takes on a new dimension: the obligation to defend this fundamental privilege. In recent years, a significant amount of effort has been devoted to the implementation of machine unlearning [2]. In the context of machine learning, this emergent field of study seeks to address the critical issue of protecting privacy and adhering to the *right to be forgotten*. Several machine unlearning approaches have been developed in recognition of the importance of working with independent data samples where no relationships exist [4, 30]. These methodologies use a variety of approaches to address the complex problem of removing the impact of data belonging to individuals who have asked to be forgotten.

Table 1: Notation descriptions.

Notation	Description	Notation	Description
G	A simple undirected graph	$\mathcal{G}$	A set of graphs
V	A set of nodes in G	E	The set of edges in G
v	A node $v \in V$	n, m	The number of nodes, edges in G
x	A node feature	y	The node label
$\mathcal{Y}$	A set of node labels	L	A set of graph labels
$X \in \mathbb{R}^{n \times d}$	The node feature matrix	$X_e \in \mathbb{R}^{m \times d'}$	Edge feature matrix
d	The size of node feature	d'	The size of edge feature
$\mathcal{D}$	A dataset	$\mathcal{D}'$	The dataset with removed data points
$\mathcal{M}$	A model trained on $\mathcal{D}$	$\mathcal{M}'$	An updated model for $\mathcal{D}'$
A	A randomized mechanism for getting M'	$\hat{\mathcal{M}}$	A model trained from scratch on $\mathcal{D}'$
h_v	A node-level embedding	h_g	The graph-level embedding
W	$\mathcal{M}$'s weight matrix	W'	The updated weight matrix
$\mathcal{N}_v$	Node v's neighbors	$X_{v'}$	The updated X after v's removal
V'	A set of nodes requested for removal	H	Hessian matrix

2 Definitions and Background

In this section, we delve into the background of GML, laying the foundation for our review on graph unlearning. We begin by introducing essential concepts and definitions that underpin the field of GML, offering readers an understanding of the subject. Building upon this knowledge, we then present our taxonomy for graph unlearning—a structured framework that categorizes and organizes various approaches and techniques used in the process of unlearning information from graph-based models. This taxonomy serves as a valuable guide for navigating the diverse landscape of graph unlearning, enabling a deeper exploration of the field's methods and applications. Through this review, we aim to provide readers with the necessary context and tools to comprehend, analyze, and advance the exciting field of graph unlearning.

2.1 Definitions

In this section, we introduce several key concepts and definitions that will be used in the rest of the paper.

Generally, a graph is defined as $G = (V, E, X, X_e)$, where V represents a set of nodes denoted by $V = \{v_1, v_2, ..., v_n\}$, and E represents a set of edges denoted by $E = \{(v_i, v_j)\}$, signifying pairwise associations between nodes. The node feature matrix, $X \in \mathbb{R}^{n \times d}$, captures the attributes or characteristics of each node, where n is the number of nodes in the graph, and d represents the dimensionality of node features. The edge feature matrix, $X_e \in \mathbb{R}^{|E| \times d'}$, is optional and can be empty ($X_e = \emptyset$) in unweighted graphs. Here d' is the dimensionality of edge features [31]. We also define $\mathcal{Y} = \{y_1, y_2, .., y_n\}$ consists of the labels associated with each node in the graph. The length of $|\mathcal{Y}|$ might be less than n in some cases where fewer node labels are available. Consider the set of graphs or a graph dataset denoted as $\mathcal{G} = \{G_1, G_2, ..., G_N\}$ and its corresponding label set $L = \{l_1, l_2, \ldots, l_N\}$. Let $\mathcal{D}$ represent the corresponding embeddings of $\mathcal{G}$, given as $\mathcal{D} \in \mathbb{R}^{N \times d}$ and obtained through a graph embedding function denoted as $f(.)$. We introduce $\mathcal{D}'$ as the embeddings obtained from $\mathcal{D}$ following a data removal request. Let's also define $\mathcal{M}$, indicates the model trained on $\mathcal{D}$, $\mathcal{M}'$ denotes the unlearned model updated through some unlearning method A and $\hat{\mathcal{M}}$ indicates the model retrained from scratch after the removal request. We provide description of the all the notations used in this study in Table 1. In the following, we present a brief overview of each of these learning tasks:

Definition 2.1 (Node Classification) *Given a training graph G and a set of labels $\mathcal{Y}$, node classification task aims to learn a representation vector h_v for v using a learning model $\mathcal{M}$ such that v's label can be predicted as $y_v = \mathcal{M}(h_v)$.*

Definition 2.2 (Link Prediction) *Given a training graph G, the link prediction task aims to learn a representation vector h for each node of the graph using a learning model $\mathcal{M}$ such that the existence of an edge between any two nodes can be predicted as $e_{u,v} = \mathcal{M}(h_u, h_v)$.*

Definition 2.3 (Graph Classification) *Given a set of graphs $\mathcal{G}$ and a set of labels L, the objective of graph classification task is to learn a representation vector h_G for G using a learning model $\mathcal{M}$ such that G's label can be predicted as $y_G = \mathcal{M}(h_G)$.*

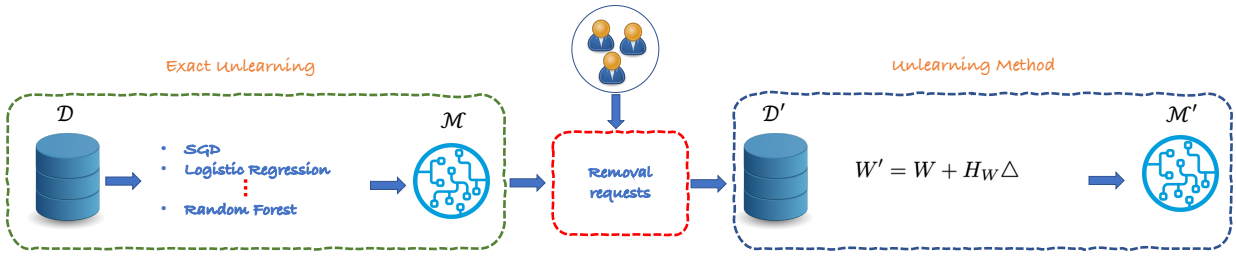

Figure 1: Illustration of machine unlearning framework. Here, W' represents the parameters of the unlearning model, derived from W, which in turn denotes the parameters of the initially learned model.

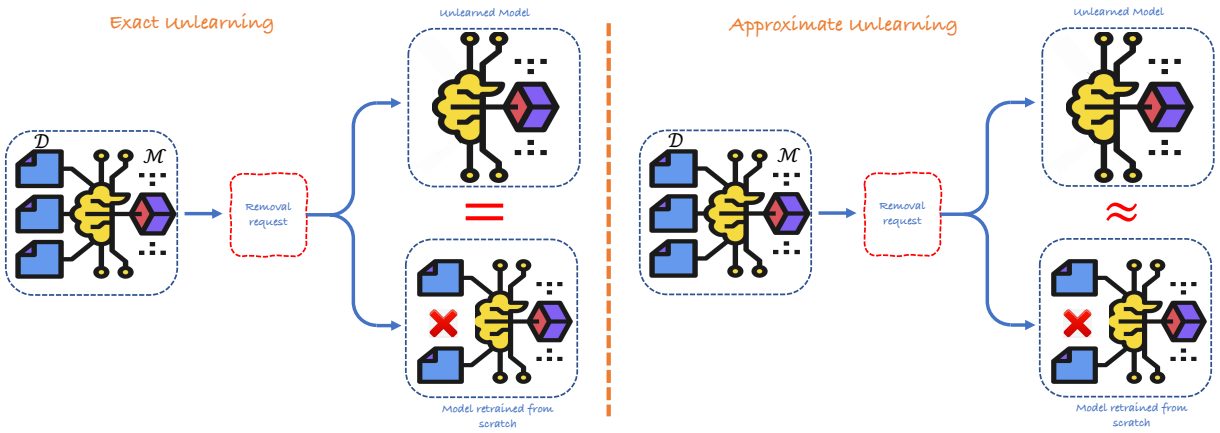

Figure 2: Illustration of exact and approximate machine unlearning approaches. The key distinction lies in the level of equality achieved during the unlearning process. Exact unlearning results in both models being identical, whereas approximate unlearning yields models that are approximately equal but not precisely identical.

The graph regression problem shares similarities with the graph classification task, but its primary focus lies in predicting continuous labels for each graph, rather than discrete classes. These machine learning tasks on graph-structured data, have garnered significant attention and found diverse applications in recent years. With these definitions in mind, we now turn our attention to defining the unlearning tasks on graphs.

Definition 2.4 (Machine Unlearning) *Machine unlearning is a process that aims to update a trained machine learning model $\mathcal{M}$ in the presence of data removal, while mitigating the influence of the removed data point $\mathbf{x}_i$. Formally, let $\mathcal{D}$ be the original dataset with N data points, where $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)\}$, and $\mathcal{D}'$ represents the dataset with the i-th data point removed. Machine unlearning aims to obtain an updated model $\mathcal{M}'$ with minimal divergence from the original model $\mathcal{M}$ when trained on $\mathcal{D}'$, while ensuring that the impact of $\mathbf{x}_i$ on $\mathcal{M}'$ is significantly reduced. Mathematically, the unlearning process seeks to minimize the discrepancy between $\mathcal{M}$ and $\mathcal{M}'$ while maximizing the privacy guarantees for the removed data point $\mathbf{x}_i$ and preserving the performance of $\mathcal{M}'$ on the remaining data.*

We provide an illustration of machine unlearning in Figure 1. Initially, a learning method $\mathcal{M}$ is trained on a dataset $\mathcal{D}$. Once a removal request or a set of requests is received, the unlearning process is initiated. The unlearning method removes the impact of the requested data points and computes the updated model from W to W', correspond to the parameters of the initially learned and unlearned models.

Machine unlearning can be categorized into two main categories: exact unlearning and approximate unlearning. Figure 2 provides a visual representation of both categories. Let's now proceed to formally define each of these approaches.

Definition 2.5 (Exact Machine Unlearning) *Given a dataset $\mathcal{D}$ and a trained model $\mathcal{M}$, exact machine unlearning refers to the process of updating the model $\mathcal{M}$ after a removal request, such that the updated model $\mathcal{M}'$ is identical to the model $\hat{\mathcal{M}}$, which is trained from scratch using the modified dataset $\mathcal{D}'$. In other words, exact unlearning ensures that the retrained model $\mathcal{M}'$ is indistinguishable from a model freshly trained on the altered dataset $\mathcal{D}'$, effectively eliminating the influence of the removed data points.*

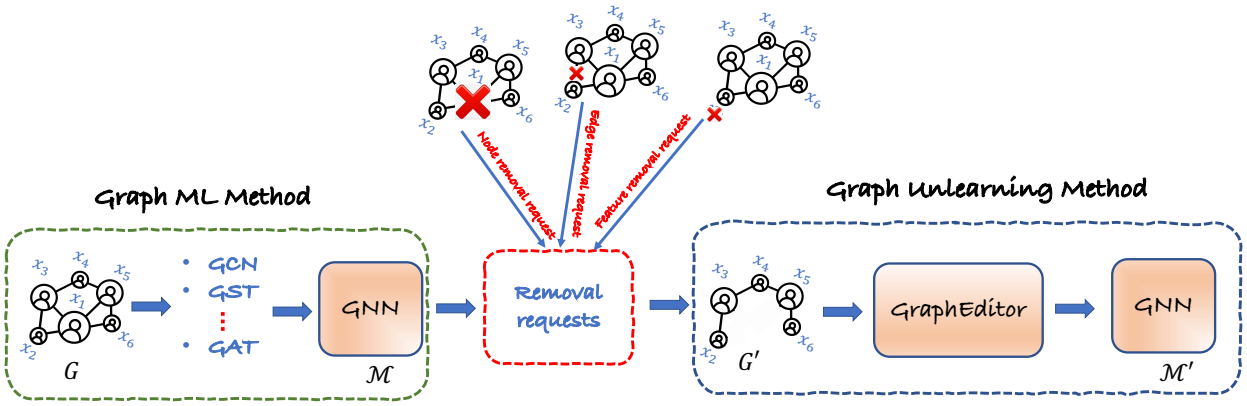

Figure 3: The graph unlearning framework is illustrated. It depicts three types of removal requests, namely node removal, edge removal, and node feature removal. Each of these requests represents different scenarios where specific elements are to be removed from the trained model.

Definition 2.6 (Approximate Machine Unlearning) *Given a dataset $\mathcal{D}$ and a trained model $\mathcal{M}$, approximate machine unlearning refers to the process of updating the model $\mathcal{M}$ after a removal request in such a way that the updated model $\mathcal{M}'$ is approximately equal to the model $\hat{\mathcal{M}}$, which is trained from scratch using the modified dataset $\mathcal{D}'$. Unlike exact unlearning, where the retrained model is identical to the one trained from scratch on $\mathcal{D}'$, approximate unlearning allows for small differences between the updated model $\mathcal{M}'$ and the model $\hat{\mathcal{M}}$. The objective is to minimize the impact of the removed data points while accepting a certain level of discrepancy between the two models.*

In approximate machine unlearning, the focus is on achieving a close resemblance between the retrained model and the one trained from scratch, rather than insisting on exact identity as in the case of exact unlearning. This flexibility allows for more practical scenarios where achieving perfect equality may be challenging or unnecessary, as long as the differences between the models remain within acceptable bounds. Now, let's proceed to define machine unlearning on graphs below.

Definition 2.7 (Graph Unlearning) *Graph unlearning is an essential extension of the unlearning process to the domain of graph machine learning, where the input is an attributed graph $G = (V, E, X, X_e)$. Formally, let $\mathcal{M}$ be the original graph machine learning model, trained on the graph G or a set of graphs $\mathcal{G} = \{G_1, G_2, .., G_N\}$ to perform a specific downstream task. The graph unlearning process aims to update $\mathcal{M}$ in the presence of data removal in the form of node or node feature vector, or edge or edge feature vector, etc. An illustration of graph unlearning has been shown in Figure 3.*

Definition 2.8 (Node Unlearning) *Node unlearning aims to effectively remove the data associated with an individual node, denoted as v, from the trained model $\mathcal{M}$ and its corresponding training graph G. When a data subject v exercises the right to revoke their data, node unlearning ensures the erasure of v's node features, represented by $X_{v'}$, as well as the links connecting v with other nodes, denoted by $e_{u,v}$ for all u in the neighborhood $\mathcal{N}_v$. Formally, node unlearning involves deriving an unlearned model $\mathcal{M}'$ that is trained on the updated graph $G_v = G \setminus \{v, X_{v'}, e_{u,v} | \forall u \in \mathcal{N}_v\}$.*

Definition 2.9 (Node Feature Unlearning) *Node feature unlearning aims to obtain an unlearned model $\mathcal{M}'$ that is trained on the modified graph $G_v = (V, E, X_{v'}, W)$, where $X_{v'}$ represents the updated node feature matrix after removing the feature vector of node v. The removal from X can be done by replacing the feature vector with zeros or ones. The model $\mathcal{M}'$ is optimized to adapt to this updated graph G_v while ensuring that the influence of X_v on the model's predictions is significantly reduced.*

Definition 2.10 (Edge Unlearning) *Edge unlearning refers to the removal of a specific edge from the trained model $\mathcal{M}$ and its corresponding training graph G, while retaining all nodes in the graph. In edge unlearning, a particular edge, denoted as $e_{u,v}$ connecting nodes u and v, is requested to be revoked by the data subject or individual to whom it pertains. This process involves removing the edge along with its corresponding attributes, represented by $X_{e(u,v)}$, from the model's training graph G, without affecting other connections and node features. Formally, edge unlearning seeks to obtain an unlearned model $\mathcal{M}'$ that is trained on the updated graph $G' = (V, E \setminus (u, v), X, X_{-e(u,v)})$, where $E_{-e_{(u,v)}}$ represents the graph with the $e_{(u,v)}$ edge removed, and $X_{-e_{(u,v)}}$ corresponds to the updated edge feature matrix.*

Unlearning in Graph-level Tasks

Unlearning in graph-level tasks where a set of graphs $\mathcal{G} = \{G_1, G_2, ..., G_N\}$ is available, unlearning involves the removal of specific data points at different granularities. The graph-level tasks could be considered in a setting such as graph classification and graph regression. Mainly, the following types of unlearning requests can be received in this scenario:

1. **Node unlearning request:** A removal request for a specific node v from a particular graph G_i in the set can be received. This request requires updating the model parameters to accommodate the removal of node v and its associated information from the graph G_i.
2. **Set of nodes unlearning request:** A removal request for a set of nodes $V' = \{v_i, v_j, v_k\}$ from a subset of graphs $\{G_i, G_j, G_k\}$ can be received. In this scenario, the model parameters need to be updated to handle the removal of the specified nodes from the respective graphs without affecting the remaining graph data.
3. **Edge or a set of edge unlearning request:** A removal request for a specific edge $e_{(u,v)}$ from one graph or a set of edges that corresponds to multiple graphs from the set can be received.
4. **Graph-level unlearning request:** A request for the entire graph G to be removed from $\mathcal{G}$ can also be received. This requires updating the model parameters to account for the removal of the entire graph G and its associated information from the set of graphs $\mathcal{G}$.

When dealing with graph-level unlearning requests, the task becomes similar to conventional machine unlearning, as each graph is treated as an independent data point in such scenarios. Consequently, well-established machine unlearning approaches can be effectively applied in these cases [20]. However, it is important to note that this approach has certain limitations when it comes to unlearning on graphs, as it may restrict the possibility of accommodating other types of unlearning requests simultaneously. For instance, node and edge unlearning requests can be received in the graph-level unlearning tasks which necessitates dedicated efforts beyond the simple unlearning. Moreover, unlearning in the domain of graphs introduces a unique and intricate challenge, setting it apart from traditional machine unlearning in other domains. The fundamental distinction lies in the underlying message passing mechanism, where information is iteratively propagated through multiple hops across the graph's interconnected nodes. Once a node or edge is removed from the graph, unlearning extends beyond simple data point removal, as it necessitates updating the effects of the deleted nodes throughout their respective neighborhoods. This complexity arises from the convolution operation inherent in graph representation learning, where information spreads across the graph's structure. Consequently, the unlearning process demands careful handling of the model's parameters to ensure the accurate recalibration of neighboring nodes and edges after data removal.

Differential Privacy

Differential privacy (DP) is a statistical property that ensures that the inclusion or exclusion of a single item from a database will not significantly alter the results of any analysis. It follows that, individuals face no increased risk when contributing their data to the database [32]. DP can be formally defined as follows:

Definition 2.11 (Differential Privacy) *[33] A randomized mechanism $A : \mathcal{D} \rightarrow \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ϵ, δ)-differential privacy if for any two adjacent inputs $d_0, d_1 \in \mathcal{D}$ and for any subset of outputs $S \subseteq \mathcal{R}$, it holds that*

$$Pr[A(d_0) \in S] \leq e^{\epsilon} Pr[A(d_1) \in S] + \delta$$

Where ϵ represents an estimation of the level of privacy protection, while δ signifies the likelihood of a privacy breach occurring. DP is achieved by introducing controlled random noise to the queries or computations performed on the data. This noise masks the exact contributions of individual data points, making it difficult to determine the specific information of any single individual in the dataset. However, the aggregate statistical properties and trends of the data can still be accurately captured. In machine unlearning context, DP serves as a powerful sufficient condition. For instance, if a model adheres to DP, adversaries are unable to distinguish whether the model was trained on the original dataset or one with a single data point removed. Therefore, the differentially private model inherently unlearns the omitted data point without necessitating explicit model updating. Recently, a number of DP-based unlearning approaches have been proposed, however, achieving sufficient accuracy with these methods often necessitates a high privacy cost [34].

When a model is differentially private, it ensures that an adversary cannot distinguish between the model trained on the original dataset and the one trained on a dataset with one data point removed [30, 35]. Thus, even without explicit model updating, a differentially private model automatically unlearns the removed data point. Although differential

privacy is a sufficient condition for unlearning, it is not necessary, and many differentially private models experience significant performance degradation [22, 33]. As a result, machine unlearning offers a trade-off between performance and computational cost, with complete retraining and differential privacy lying at two opposite ends of the spectrum [36].

2.2 Background

Given an attributed graphs, several learning tasks can be defined to leverage the graph's rich information. For instance, given an attributed graph G, one fundamental learning task involves predicting a label $f(G)$ associated with the entire graph. Likewise, the learning process can be applied at the node-level or edge-level. For instance, given G and a specific node $v \in V$, the task becomes predicting a label $f(v)$ associated with that particular node. Alternatively, for a pair of nodes (u, v) within the graph, the learning task shifts towards predicting a link $g(u, v)$ between those two nodes, reflecting their relationship or connection within the graph. These learning tasks allow for the extraction of valuable insights and predictive power from the attributed graph, fostering a wide array of applications across domains such as social networks, recommendation systems, and biological networks [12, 37, 10, 38, 39, 40, 41, 42].

At the core of GML domain, graph kernels are fundamental algorithms used to measure the similarity between graphs [43]. These kernels effectively compare the structural properties of different graphs, enabling us to classify and cluster them based on their shared features. Complementing graph kernels are graph descriptors [42, 41], which provide compact representations of graphs, capturing their essential characteristics. These descriptors serve as informative fingerprints of the underlying data, facilitating efficient and scalable analyses of large graph datasets [44]. In recent years, the emergence of graph neural networks (GNNs) has revolutionized the field of GML [12, 13]. GNNs are a class of deep learning models specifically designed to handle graph-structured data. These innovative architectures enable us to learn powerful representations of nodes and edges within a graph, and the graph-level representations, leveraging local and global information to perform tasks such as node classification, link prediction, and graph-level predictions [37, 45]. By iteratively aggregating information from neighboring nodes through message passing mechanism, GNNs excel at capturing intricate patterns and dependencies present in graphs, making them indispensable for a wide range of applications, including social network analysis, drug discovery, and recommendation systems. The flexibility and adaptability of GNNs have sparked immense interest in the scientific community, propelling further advancements in this exciting field of research [16].

In order to address the diverse learning tasks defined on graphs, the generation of rich embeddings that encapsulate the graph's information becomes paramount. GML has offered several techniques for creating such embeddings, including traditional methods like graph kernels [43], and graph descriptors [44, 42, 41, 46]. However, the emergence of GNNs has proven transformative, as they provide an innovative solution for generating expressive embeddings that preserve the intricate semantics of the graphs [13, 31, 10]. By effectively employing message passing and graph convolution techniques, GNNs adeptly capture the dependencies and interactions among nodes and their features, resulting in powerful representations that can be utilized for a wide range of learning tasks. As we delve into graph unlearning in subsequent sections, we will provide a concise overview of these GML approaches and their interconnections, laying the groundwork for further exploration of unlearning techniques.

Graph Neural Networks:

Here we focus on Message Passing Neural Networks (MPNNs) that operate through an iterative scheme, where they maintain a representation (embedding) $h_v^{(t)} \in \mathbb{R}^d$ for each node $v \in V$. At each iteration t, MPNNs update each node's embedding $h_v^{(t)}$ by aggregating information from its neighboring nodes and possible edge attributes. The message passing process can be formally described using three equations: initialization, aggregation, and update.

1. Initialization:

$$h_v^{(0)} = x_v \quad \forall v \in V \tag{1}$$

where x_v represents the initial feature representation of node v.

2. Aggregate:

$$a_v^{(t)} = f_{Agg}^{(t)}(a_v^{(t-1)}, \{\{a_u^{(t-1)}, w(u,v)\ u \in \mathcal{N}(v)\}\}), \quad 1 \leq t \leq T \tag{2}$$

The aggregate function f_{Agg} represents a bijective function, which can take various forms such as the sum, product, or any other suitable transformation, while $\mathcal{N}(v)$ indiciates the neighbors of node v. T indicates the depth of the network.

3. Update:

$$h_v^{(t)} = f_{Up}(h_v^{(t-1)}, a_v^{(t)}) \tag{3}$$

The update function f_{Up} is typically expressed as a weighted combination using learnable weight matrices W:

$$f_{Up}(h_v^{(t-1)}, a_v^{(t)}) = \sigma(W_1^{(t)} h_v^{(t-1)} + W_2^{(t)} a_v^{(t)}) \tag{4}$$

The final node representations $h_v^{(T)}, \forall v \in V$, are obtained as the last iteration's outputs, potentially concatenated with a linear classifier. In this process, the node embeddings undergo multiple iterations, each refining the representation using information from neighboring nodes. The result is a set of enriched node representations, which can further be utilized for downstream tasks, and in some cases, combined with a linear classifier to make predictions or perform other computations. For the graph-level task such as graph classification or graph regression, generally a pooling or readout function is used as defined below.

$$f(G) = f_{Read}(h_v^{(T)} | \forall v \in V). \tag{5}$$

In our assumption, f_{Read} is considered a straightforward operation such as a simple sum or average. Nevertheless, it can be substituted with more sophisticated graph pooling layers designed to learn and mitigate potential information loss [47]. Following the f_{Read} function, the GNN architecture proceeds with using MLP or 1D convolutions. These subsequent layers serve to further enhance the representations, capturing higher-order dependencies and complex patterns in the graph [48].

3 Categorization and Frameworks

We acknowledge that the domain of graph unlearning is still in its early stages, with limited exploration in this direction. Drawing from the existing literature, we classify graph unlearning into three distinct categories: exact unlearning, approximate unlearning, and differential privacy. In Table 2, we offer a concise overview of these methods, providing a glimpse into their essence. Moving forward, in the forthcoming paragraphs, we will delve into each of these categories, providing explanations of the methodologies employed in graph unlearning. Our aim is to shed light on the intricacies of these methods, uncovering their unique contributions to the fascinating domain of graph unlearning.

3.1 Exact Graph Unlearning

The authors in [51] introduce GraphEditor, a graph representation learning and unlearning approach that supports both node and edge deletion and addition. GraphEditor offers an exact unlearning solution without necessitating extensive retraining. The core principle behind GraphEditor lies in transforming the conventional GNN training problem into an alternative problem with a closed-form solution. Once the initial training is completed and the weight matrix W is obtained, GraphEditor computes the closed-form solution, returning W', the updated weight matrix, and S^*, the inverse correlation matrix for unlearning. In the event of a deletion request, GraphEditor updates the model parameters W' by first removing the effect of the requested node/edge on the remaining nodes and then updating the model. This targeted approach ensures efficient and precise unlearning, making GraphEditor a useful tool for managing graph data and maintaining model integrity. However, a key limitation of the paper lies in its applicability, as the majority of GNNs employed are non-linear. Consequently, the current results may have limited relevance in such scenarios. Furthermore, envisioning the generalization of the current approach to more intricate network structures presents challenges, as it heavily relies on the linear characteristics of the problem. [62] propose an unlearning approach for training the model using mini-batch stochastic gradient descent (SGD). Throughout the training process, the model parameters are saved at each iteration. When deletion requests are received, retraining commences solely from the iteration when the deleted data first appeared. This strategy optimizes the unlearning process, allowing for targeted updates and minimizing redundant retraining efforts.

3.2 Approximate Unlearning

GraphEraser [17] stands as one of the pioneering works in the emerging domain of graph unlearning, providing a novel framework for addressing unlearning requests in graph machine learning models. This approach begins by partitioning the original training graph into separate and disjoint shards. Each shard is then used to train an individual model, denoted as $\mathcal{M}_i$, in parallel. During this process, GraphEraser also learns an optimal importance score, denoted as α_i, for each shard model, which reflects its significance in the overall prediction process. When a node v requires a prediction, GraphEraser forwards v to all shard models and obtains their corresponding posteriors. These posteriors are then aggregated using the optimal importance score α_i for each shard model, enabling GraphEraser to make a robust prediction for v. Furthermore, when a node u makes an unlearning request, GraphEraser promptly removes u from the corresponding shard and subsequently retrains the shard model. This process ensures that the influence of node u is mitigated, and the model remains updated and accurate without requiring a complete retraining of the entire graph machine learning model. GraphEraser draws inspiration from the widely recognized SISA (Sharded, Isolated, Sliced,

Table 2: Categorization of graph unlearning approaches. $V\mathcal{M}'$ denotes node unlearning, $E\mathcal{M}'$ represents edge unlearning, $X\mathcal{M}'$ refers to node feature unlearning, $G\mathcal{M}'$ represents the graph unlearning approach, and graph uni. indicates graph universal methods. DP stands for Differential Privacy, approx. unl. for approximate unlearning, FL for federated learning, and ACM Tran. IST refers to ACM Transaction on Intelligent Systems and Technology. The **Code** column indicates the availability of the implementation for the proposed method.

Method	Year&Venue	Code	$V\mathcal{M}'$	$E\mathcal{M}'$	$X\mathcal{M}'$	$G\mathcal{M}'$	G Type	$\mathcal{M}'$ Type	Method Summary
GraphEraser[17]	SIGSAC2022	✓	✓	✓	✗	✗	undirected	approx. unl.	Cluster the graph, train a separate model on each cluster, and then aggregates the results
EraEdge[49]	OpenRev.2022	✗	✗	✓	✗	✗	undirected	approx. unl.	Estimate the edge influence and propose EraEdge method to update the model
CertifiedDR[36]	ICML2022	✓	NA	NA	NA	✓	graph uni.	exact unl.	Use influence function to update model and provide bounds on the security parameters
Certified Unl.[50]	NeurIPS2022	✓	✓	✓	✓	✗	undirected	approx. unl.	Use Hessian for updating the model and provides theoretical foundation
GraphEditor[51]	OpenRev.2022	✗	✓	✓	✓	✗	undirected	exact unl.	Consider non-convex setting, compute closed-from solution of GNN output and $\mathcal{Y}$ and update the model
GUIDE [52]	USENIX2023	✓	✓	✗	✗	✗	undirected	approx. unl.	Compute k shards, apply repair function to recover edges, train k models and aggregate
GST Unlearn[20]	WWW2023	✓	✗	✗	✗	✓	undirected	approx. unl.	Use the traditional influence functions to compute the change and update the model
GIF[53]	WWW2023	✓	✓	✓	✓	✗	undirected	approx. unl.	Use the traditional influence functions and incorporate addition term to the loss
Projector[54]	AISTATS2023	✓	✗	✗	✓	✗	undirected	approx. unl.	Use orthogonal projection as a weighted combination of node features for unlearning
GNNDELETE[21]	ICLR2023	✓	✓	✓	✓	✗	undirected	approx. unl.	Introduce shared weight matrices across the nodes and use layer-wise deletion operator to udpate the model
SGC[18]	ICLR2023	✓	✓	✓	✓	✗	undirected	approx. unl.	Use influence function for model updates and derive robust theoretical guarantees in convex setting
SAFE[19]	arXiv2023	✗	✓	✓	✗	✗	undirected	approx. unl.	Using graph sharding mechansim to train secure GNN models
FedLU[55]	WWW2023	✓	✓	✓	✓	✗	KG	approx. unl.	Federated Learning based learning and unlearning approach
DP[33]	SIGSAC2016	✓	✗	✗	✗	✓	graph uni.	DP	Clipping the l_2 norm of each gradient, computing the average, adding Guassian noise, and then taking a step in the opposite direction
SGNN[56]	BigData2019	✗	✓	✓	✓	✗	undirected	FL	Consider node feature similarity matrix and use FL to train a GNN
DP-GCN[34]	arXiv2021	✓	✓	✗	✗	✗	directed	DP	Create nodes training dataset with subgraph sampling and train the model in DP-setting
LPGNN[57]	SIGSAC2021	✓	✓	✓	✓	✗	undirected	DP	Adding noise to node features to ensure DP and train model in distributed environment
LINKTELLER[58]	IEEESP2022	✗	✗	✓	✗	✗	undirected	DP	Perturbing the input graph using randomized response and train the model
PRIVGNN[59]	ACMSNGT2022	✗	✓	✓	✓	✗	undirected	DP	Considering a private data setting, neighbors are obtained through $k-$nearest neighbors and trained public GNN model
Feddy[60]	ACM TIST2022	✗	✗	✗	✗	✓	dynamic	FL	Perform secure aggregation and train GNN in FL setting
GDP[61]	arXiv2023	✗	✓	✓	✗	✗	directed	DP	Introduce the notion of relaxed node-level data adjacency in DP setting for GNNs

and Aggregated) [2] framework, known for its versatility in terms of model architecture. The fundamental concept behind SISA involves randomly partitioning the training dataset into distinct and disjoint shards, each independently trained to form a set of shard models. This decentralized training approach ensures that each shard model focuses on specific subsets of the data, enhancing efficiency and parallelizability. While the SISA framework provides quite a versatile approach for training and unlearning in graph machine learning, it also comes with certain limitations, when it comes to graphs. Specifically, it may lack the flexibility to handle complex unlearning requests that span multiple shards or require more extensive model updates. Moreover, SISA might not be well-suited for graphs with highly heterogeneous data or intricate interconnections between nodes, potentially impacting the performance and generalization capabilities of the shard models. The aggregation of predictions from shard models may also introduces challenges in determining the optimal importance scores, especially when faced with dynamic or evolving graph data. Following the GraphEraser, which is a transductive unlearning approach, the authors in [52] present a method called GUIDE (GUided InDuctivE Graph Unlearning). GUIDE consists of three essential components: guided graph partitioning, graph repairing, and aggregation. GUIDE begins by partitioning the graph into shards using newly proposed clustering techniques. Subsequently, GUIDE employs subgraph repair methods to restore lost connections and integrates them into their respective shards. This step is crucial to reconstruct the graph after partitioning where some of the connections are lost. Finally, GUIDE applies an aggregation method to learn a similarity score for each shard. This similarity score is then utilized for making inferences during the unlearning process.

The paper [46] introduces a graph embedding technique called Graph Scattering Transform (GST). GST relies on a collection of multiresolution graph wavelets, a pointwise nonlinear activation function, and a low-pass operator. Building upon GST, [20] propose an interesting graph unlearning approach, aptly named Unlearning GST. By utilizing

GST embeddings, Unlearning GST defines an unlearning model denoted as $\mathcal{M}'$, which updates the trained model from W' to W'. The latter represents an approximation of the unique optimizer of $(L(W, \mathcal{D}'))$. The derivation of W' is as follows:

To begin, H_W is defined as the Hessian of $(L(., \mathcal{D}'))$ at W and is computed as:

$$H_W = \nabla^2 L(W, \mathcal{D}')$$

Next, the gradient difference is evaluated by computing the difference between $\nabla^2 L(W, \mathcal{D})$ and $\nabla^2 L(W, \mathcal{D}')$:

$$\triangle = \nabla^2 L(W, \mathcal{D}) - \nabla^2 L(W, \mathcal{D}') \tag{6}$$

Using the obtained gradient difference, W' is derived as follows:

$$W' = W + H_W \triangle$$

Thus, W' represents the updated model obtained through this unlearning process. However, the proposed approach has a limitation related to the computation of the wavelet transform, which is cubic for node removal and quadratic for node feature removal.

A notable addition to the research landscape, akin to the aforementioned [46] study, is the certified graph unlearning approach presented in [50]. Inspired by the principles laid out in [36], this study extends the unlearning mechanism to accommodate graphs within a linear GNN setting. Generalizing the bounds, the authors obtain certified unlearning guarantees for node features, nodes, and edges. Despite its significance, this work's main limitation lies in its focus on linearity in GNNs, which limits its applicability to a select set of applications.

In the graph-level unlearning context, each graph effectively becomes an independent data point, and the task of unlearning translates to removing or updating individual graphs from the model. This interesting transformation aligns the graph-level unlearning problem with the established principles of machine unlearning. As a result, the conventional methodologies and techniques developed for machine unlearning hold relevance and can be effectively applied to tackle the graph-level unlearning problem [20]. A notable contribution in this domain can be found in [36]. This approach offers a method for *certified removal* of deleted data from a trained model. Operating within the non-convex setting, the method applies the Newton update, also known as the influence function, to the gradient influence of the removed data point. The paper goes on to establish robust theoretical bounds on the security parameters, providing rigorous proof of the certified removal's effectiveness.

In the pursuit of approximate unlearning within the non-convex setting, an approach called EraEdge is introduced in [49]. EraEdge specifically targets edge unlearning in graphs. To achieve this, EraEdge focuses on reversing the influence of the edge that needs to be removed from the new model W', leveraging the concept of influence functions [63]. The authors propose an estimation of the influence function by upweighting the set of all affected nodes and subsequently calculate the reverse of the Hessian matrix multiplied by the gradient vector to obtain the influence. To address computational complexity, they employ conjugate gradient optimization, effectively reducing the computational burden. In the same research domain, the authors in [53] introduce another unlearning method, Graph Influence Function (GIF). GIF aims to model the influence of each training data point on the model with respect to various performance criteria and subsequently eliminates the negative impact. The method incorporates an additional loss term that considers the influence of neighbors and estimates parameter changes in response to a $\epsilon-$mass perturbation in deleted data. Another similar approach is presented by Cong et al. in their work [54], where they introduce "Projector". This method aims to remove the impact of deleted data from the learned model by projecting the weight parameters to a different subspace that is unrelated to the deleted data. Specifically, Projector is designed to work with linear GNNs, and it demonstrates that all gradients lie within the linear span of all node features. Leveraging this insight, it performs unlearning using an orthogonal projection represented as a weighted combination of the remaining node features. By applying this technique, the learned model can effectively adapt to the removal request and update itself while minimizing the impact of the deleted data.

The unlearning process, as explored in [64], can potentially harm the performance of the underlying predictive model. In an effort to mitigate this issue and avoid updating the trained model, [21] introduces GNNDELETE. This approach formalizes two essential properties for the GNN deletion method: *the deleted edge consistency* and *neighborhood influence*. The deleted edge consistency ensures that the predicted probabilities for deleted edges in the unlearned model stay similar to those for nonexistent edges. Similarly, the neighborhood influence guarantees that predictions in the local vicinity of the deletion retain their original performance and remain unaffected by the removal. In pursuit of efficiency and scalability, GNNDELETE employs a layer-wise deletion operator to modify a pre-trained GNN model. Upon receiving deletion requests, GNNDELETE freezes the existing model weights and introduces small, shared weight matrices across the nodes in the graph. GNNDELETE guarantees strong performance by ensuring that the difference

between node representations obtained from the trained model $\mathcal{M}$ and those revised by GNNDELETE $\mathcal{M}'$ remains theoretically bounded. Building upon the foundation laid by the certified data removal approach in [36], an approximate unlearning approach has been proposed more recently in [18]. This study introduces methods for unlearning node features, edges, and nodes, presenting robust theoretical guarantees within the context of limited GNN models operating in a convex setting. The crux of their work lies in the utilization of influence function, combined with the incorporation of graph information through an additional term in the gradient update function. Moreover, the paper establishes robust bounds on the gradient residual norms, elevating the efficacy of graph unlearning in three distinct scenarios.

In the knowledge graph setting, the authors in [55] introduce FedLU, a federated learning framework for knowledge graphs (KG). The knowledge graph setting involves complex and diverse data, and FedLU addresses the challenges arising from data heterogeneity. One of the key contributions of FedLU is the introduction of a mutual knowledge distillation method. This method plays a crucial role in coping with the drift that can occur between local optimization and global convergence due to the heterogeneity present in the KG data. By utilizing mutual knowledge distillation, FedLU aims to harmonize the information learned across different local models and ensure more consistent global convergence. This not only helps in maintaining model performance but also enables better collaboration and sharing of knowledge across the participating entities in the federated learning setting. Another significant aspect of FedLU is the proposal of a KG embedding unlearning method. Knowledge forgetting is an essential requirement in dynamic knowledge graphs where information needs to be updated or removed over time. FedLU achieves this by combining two techniques: retroactive interference and passive decay. Retroactive interference allows the model to selectively unlearn specific knowledge embeddings related to outdated or irrelevant information, while passive decay gradually reduces the influence of such knowledge over time. This approach ensures that the KG remains up-to-date and relevant, adapting to changes in the underlying data distribution and facilitating the extraction of accurate and meaningful insights from the evolving knowledge graph.

3.3 Deferentially Private (DP) Approaches

Standard ML models are already known to be highly susceptible to leakage of sensitive information about the training data [65]. This is due to the fact that they are usually trained with Stochastic Gradient Descent (SGD) like optimization which may learns sensitive information about the training samples [33]. Let's consider an example of training a ML model to predict a person's income based on features like age, education level, and occupation. The dataset used for training contains information about various individuals, including their income, which is considered sensitive information. During the training process, SGD updates the model's parameters by calculating gradients based on a randomly selected mini-batch of individuals from the dataset. Suppose a particular mini-batch includes individuals with high incomes. As SGD updates the model based on this mini-batch, it may starts learning patterns that are specific to individuals with high incomes, even if those patterns are not explicitly intended to be captured. This learned information can unintentionally reveal sensitive details about specific individuals in the training data, leading to potential privacy breaches. For example, if the model has learned that a certain combination of age, education level, and occupation indicates a high income, an adversary may be able to infer the income of an individual with those specific attributes from the trained model's predictions.

The potential leakage of sensitive information in scenarios where the trained model is shared or used in sensitive applications raises concerns about individual privacy [34]. DP offers a solution to this issue by introducing controlled noise into the model updates during SGD, ensuring that individual data points do not unduly influence the model's parameters. Well-known works in this line of research include [33, 32], which introduces a novel algorithmic technique for training models under the umbrella of differential privacy with tight privacy bounds. [33] approach involves clipping the l_2 norm of each gradient, computing the average, adding noise to protect privacy, and then taking a step in the opposite direction of this average noisy gradient. Additionally, it incorporates a privacy loss function to further enhance privacy protection during model training.

The risk of sensitive information leakage is particularly pronounced in GNNs, as each node's prediction is based on its own features as well as aggregated data from its neighboring nodes. This aggregated information may potentially include sensitive node features, node labels, and connectivity details. Preserving privacy in GNNs presents significant challenges, as the learning process involves considering a set of neighboring nodes, making existing DP-based approaches not directly applicable. To address this, the authors of [34] introduce the Differential Privacy Graph Convolutional Network (DP-GCN) specifically tailored for GNNs. DP-GCN proposes a unique subgraph sampling technique with occurrence constraints to control the frequency of each node's appearance in the training dataset. Moreover, DP-GCN adopts a variant of the DP-SGD [66] approach to train the model while ensuring privacy for each update. This is achieved by adding noise according to the sensitivity of the aggregated gradient with respect to any individual node, which is bounded through careful subsampling of the input graph. The subsampling strategy employed by DP-GCN allows it to provide strong guarantees on node-level differential privacy, making it an effective solution for preserving individual

node privacy within GNN-based applications. Nevertheless, this method lacks inference privacy and is limited to 1-layer GNNs, thereby hindering its ability to leverage higher-order aggregations.

In recent times, there have been several efforts to incorporate DP to provide formal privacy guarantees in various GNN learning scenarios. One notable proposal by [58] introduces an edge-level DP learning algorithm for GNNs. They achieve this by directly perturbing the input graph using randomized response (EdgeRand) or the Laplace mechanism (LapGraph) and subsequently train the GNN on the perturbed graph. Despite its effectiveness, this method does not readily extend to the node-level privacy setting. In a different approach, [57] present a locally private GNN model in a distributed learning setup, where node features and labels remain private while the GNN training is federated by a central server with access to graph edges. However, this method is not suitable for scenarios where graph edges are also private. In a similar vein, another approach [67], denoted as GAP, employs aggregation perturbation to achieve edge-level privacy. The GAP framework is composed of three key modules: the encoder module, which learns solely from node features, the aggregation module, responsible for learning node representations from the graph topology, and the classification module. Addressing node-level privacy in a centralized learning setting, [59] adapt the PATE framework [68]. They train the student GNN model using publicly labeled graph data and teacher GNN models trained exclusively for each query node. Nevertheless, their reliance on public graph data restricts the applicability of their approach.

A wave of recent approaches has emerged, aiming to tackle the issue of privacy through techniques like federated and split learning. [56] propose a GNN that leverages structural similarity and federated learning, striving to cloak content and structure information, thereby enhancing privacy protection. On a similar front, [60] crafted a distributed and secure framework, embarking on the journey to unravel object representations in video data through graph sequences, expertly blending GNN and federated learning. They wove in secure aggregation primitives to fortify privacy during the federated learning process. However, their focus lies in a different territory, as they assume each party possesses a series of graphs extracted from video data, and the server orchestrates federated learning to construct an inductive GNN over this distributed dataset of graphs. Meanwhile, [69] directed their efforts towards privacy-preserving node classification. They splintered the computation graph of a GNN across multiple data holders, enlisting the expertise of a trusted server to amalgamate information from diverse parties and achieve comprehensive training. Nonetheless, these approaches depend on a trusted third party for model aggregation, leaving their privacy guarantees without formal validation. Recognizing that privacy requirements may differ for node attributes and graph structure in graph unlearning, [61] introduce a graph deferential privacy (GDP)framework. This paper also introduces the concept of relaxed node-level data adjacency. This relaxation allows to establish varying degrees of privacy for the graph topology while still ensuring the privacy of individual node attributes. With this relaxation, GDP framework provides a flexible solution to address diverse privacy needs in graph-based learning scenarios. In a related study by [19] , an alternative method involving shard graphs within the context of differential privacy has been put forward. This proposal leverages lightweight cross-attention adapters, which operate on the representations of a transformer encoder that has been pretrained. Specifically, each node in the graph is linked to an InCA adapter, as introduced in [70], which is trained on the combined data of the node and all its outward edges.

4 Graph Unlearning Evaluation

The evaluation of graph unlearning encompasses both approximate and exact unlearning approaches, each serving distinct purposes. In approximate unlearning, the primary objective is to ensure that the unlearned model $\mathcal{M}'$, maintains probabilistic similarity (ϵ, δ) with the model $\mathcal{M}$, which is retrained from scratch, ϵ, $\delta > 0$. The aim is to minimize these differences while maintaining a level of practicality and efficiency in the unlearning process. On the other hand, exact unlearning pursues a more stringent goal, where ϵ and δ are reduced to zero. In this scenario, the $\mathcal{M}'$ is intended to be a perfect replica of the $\mathcal{M}$, leaving no traces of past data or learned information. In the following, we dicsuss the commonly used evaluation metrics use for graph unlearning models.

Unlearning Efficiency

Unlearning efficiency refers to the speed with which a machine unlearning model can modify or remove specific data points or knowledge without compromising its overall performance. Unlearning efficiency is an essential aspect in scenarios where models need to adapt quickly to changing data distributions, new insights, or emerging requirements. Although there is no standardized formula for measuring efficiency, typically, models are assessed by comparing the time required for unlearning and retraining the model from scratch [17].

Model Effectiveness

Model effectiveness refers to the assessment of how efficient and reliable an unlearning model is in achieving its intended purpose. When evaluating the performance of unlearning models, various metrics are commonly employed,

such as $F1 - score$ and others. In addition to $F1$, other metrics like precision, recall, and area under the curve (AUC) can also offer valuable insights into the model's performance, especially in specific use cases. To evaluate the effectiveness of an unlearning model, a common approach is to compare its performance with that of a model trained from scratch on the entire dataset. This provides a better sense of how well the unlearning model performs in comparison to a the model trained from scratch [17].

Delete Data Replay Test (DDRT): DDRT is a new test proposed in [51] to test if an unlearning method can accurately unlearn the deleted data. For the test, an extra label category is added to all nodes, and the label of each deleted node is changed to this new category. Then, a comparison between the deleted nodes predicted as the extra-label category before and after the unlearning process is performed. The idea is that a successful unlearning method should never predict a node as the extra-label category after unlearning. This test helps understand how well the unlearning method works.

Membership Inference (MI): Membership inference is a privacy attack that aims to determine whether a specific data point was part of the training dataset used to train a machine learning model. This becomes particularly relevant in machine unlearning, posing a significant threat to privacy, and the success of MI can serve as a suitable measure for the quality of unlearning [21, 71]. However, evading MI proves to be a challenging task, which becomes of interest when building machine unlearning models. In [72], it was demonstrated that every GNN model is susceptible to MI attacks. To counter this vulnerability, they introduced two defense strategies utilizing output perturbation and query neighborhood perturbation. On the other hand, [73] approached the issue by considering the data to be unlearned as backdoored data. It's essential to recognize that while defense methods against MI attacks offer valuable perspectives for assessing unlearning, they serve distinct purposes from the unlearning process itself.

5 Graph Unlearning Applications

Graph unlearning stands as an important topic in the realm of machine learning and data privacy. Due to the exponential development of graph-structured data across multiple domains, ensuring the right to be forgotten and protecting individual privacy have become of the utmost importance. As we observe the remarkable success of graph machine learning applications, it is important that we devise effective graph unlearning strategies. By enabling the removal of all forgotten data while preserving the integrity of interconnected networks, graph unlearning gives individuals greater control over their data and nurtures confidence in the responsible management of data. In the forthcoming sections, we discuss potential applications of graph unlearning in a variety of domains. Figure 4 provides an illustration of a few potential graph unlearning applications.

5.1 Social Networks

The application of graph unlearning in social networks is essential for mitigating the risk that a trained model may inadvertently learn sensitive information about users' social relationships. On the basis of the underlying graph structure, machine learning models are frequently employed in social network analysis to infer user behavior, preferences, and social interactions. However, these models may inadvertently capture and encode sensitive information within their learned representations, such as private friendships, personal interests, and social circles [74]. Graph unlearning has the potential to provide means of mitigating this privacy concern. By employing graph unlearning techniques to the trained model, it is possible to delete or anonymize sensitive user data from the model's learned parameters. This promotes a more secure and ethical environment for social network analysis by ensuring that the model's predictions and insights are obtained without compromising the privacy and confidentiality of individual users. Through graph unlearning, researchers and organizations can strike a delicate equilibrium between leveraging the power of machine learning in understanding social dynamics and respecting and preserving users' privacy rights in the ever-changing social network landscape [75, 76].

5.2 Confidentiality in Financial Networks

In financial transactions and fraud detection, learning models are frequently used to analyze transaction patterns and identify potential anomalies. However, these models may inadvertently acquire sensitive information, such as the financial histories, transaction details, and account balances of individuals, thereby jeopardizing the confidentiality and privacy of users' financial data. By applying graph unlearning techniques to these models, it is possible to remove or obfuscate sensitive data from their learned representations. This ensures that the models can continue to provide accurate insights and detect fraudulent activities without compromising the privacy/confidentiality of individual users financial information [77]. Adopting graph unlearning in financial networks establishes a robust and ethical framework for data privacy and security, fostering user confidence and protecting sensitive financial data in this essential domain.

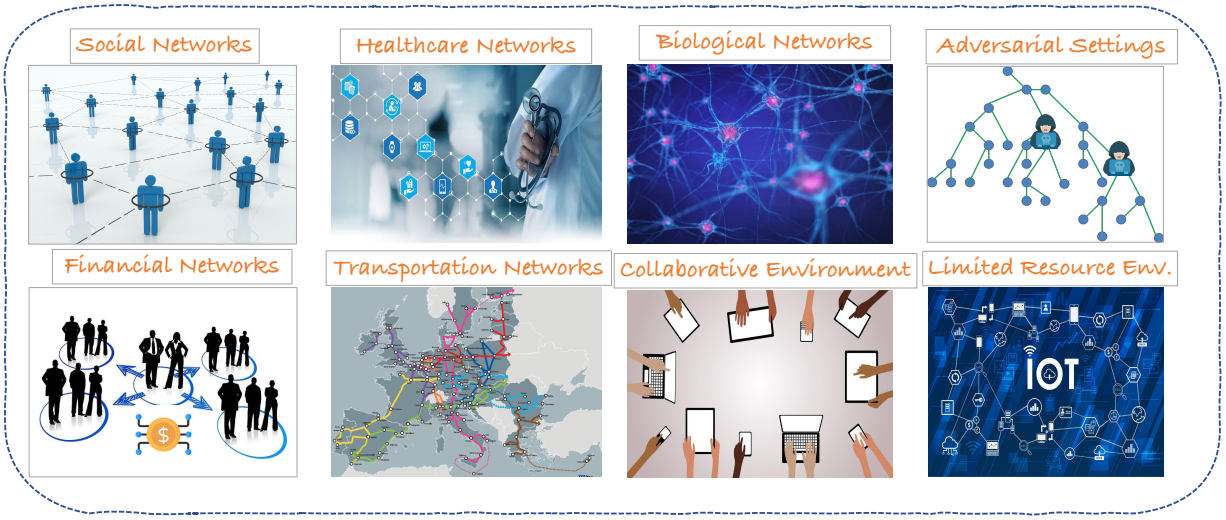

Figure 4: Some potential graph unlearning applications.

5.3 Ethical Consideration in Healthcare Networks

In healthcare networks, ethical considerations are of the uttermost importance. GML models can be used to analyze patient data, identify disease patterns, and support clinical decision-making in the healthcare industry. However, these models may inadvertently acquire confidential patient information, such as medical histories, diagnoses, and treatment plans, raising grave privacy concerns. These models may incorporate confidential health information within their learned representations, posing a substantial threat to patient confidentiality [78]. Integrating graph unlearning into healthcare networks may enables the effective protection of patient privacy.

5.4 Ethical Consideration in Transportation Networks

Commonly, machine learning models are employed in traffic analysis to predict traffic patterns, optimize routes, and enhance transportation systems [79]. However, these models may accidentally acquire sensitive information about an individual's mobility patterns, daily commute routes, and specific location histories, thereby compromising privacy. The risk of data leakage and identification of individuals through these models raises ethical concerns regarding the protection of the anonymity and privacy of travelers' data. By incorporating graph unlearning into traffic analysis, it is possible to effectively protect individual privacy. Using graph unlearning techniques permits the removal of sensitive data from trained models while preserving the models' capacity to provide accurate traffic predictions and enhance transportation efficiency.

5.5 Ethical Consideration in Biological Networks

In the fields of bioinformatics and biological network analysis, machine learning models play a crucial role in the analysis of complex biomolecular interactions and functions. However, these models may accidentally discover intricate details about individual genetic sequences, gene expressions, behavioral information and metabolic pathways, which raises privacy concerns [7]. The risk of data exposure and potential individual identification posed by these models can compromise the confidentiality and privacy of genomics data. By incorporating graph unlearning into biological networks, these ethical concerns can be effectively addressed.

5.6 Individuals' Privacy in Data Sharing in Collaborative Environment

In collaborative research projects, for instance, researchers may be required to share learning models trained on sensitive data in order to reproduce and validate results; however, they must also safeguard the privacy of individual contributors. In this scenario, graph unlearning can enables researchers to share the model's insights without disclosing the underlying raw data, thereby protecting the privacy of data contributors. Similarly, in cross-organizational initiatives, businesses frequently collaborate to optimize operations and acquire a holistic understanding of their respective fields. However, they must protect the confidentiality of sensitive business information. In the context of smart city projects, in which

various stakeholders seek to enhance urban services through data-driven strategies, there is an increasing demand to share data while protecting the privacy of citizens. Graph unlearning may enables city authorities to collaborate with private companies, researchers, and service providers without disclosing personal data, achieving a balance between urban development and data privacy. Moreover, in academic partnerships, researchers from various institutions may collaborate to resolve complex scientific problems, necessitating the exchange of data and models [80]. Graph unlearning ensures that valuable research insights can be shared without violating data protection laws, thereby fostering productive academic collaborations.

5.7 Adversarial Setting

In sensitive domains like healthcare, finance, and personal recommendation systems, the consequences of a security breach or an attacker gaining access to a trained machine learning model can be severe. In these scenarios, an attacker may attempt to compromise user privacy by injecting malicious data into the model's training dataset or by stealing sensitive information from the model itself. This malicious activity can lead to various detrimental outcomes, such as unauthorized access to personal medical records, financial data, or private user preferences [59]. For instance, in the healthcare domain, if an attacker gains access to a medical diagnosis model, they could inject misleading or harmful data, potentially leading to inaccurate diagnoses and jeopardizing patient safety. In the finance sector, unauthorized access to a fraud detection model could enable attackers to manipulate the system, bypass security measures, and conduct fraudulent transactions. Similarly, in personal recommendation systems, adversaries may exploit information about users' preferences and behaviors obtained through malicious data injection or information stealing [58]. This could result in biased or manipulated recommendations, eroding user trust and leading to potential misuse of personal data. To safeguard against these risks, the application of machine unlearning becomes essential.

5.8 Limited Resource Environment

The application of graph unlearning may holds immense potential, especially in limited resource environments such as the Internet of Things (IoT). In such scenarios, devices at the edge may encounter situations where certain training examples become obsolete, out of distribution, or no longer valid [15, 81]. For instances, a network of smart traffic cameras placed strategically across the city to monitor traffic flow and detect any congestion or accidents. Each traffic camera acts as a node in the knowledge graph, and the connections between these nodes represent the spatial relationships between different camera locations. Assume that we have different devices at different locations of the cities to collect data from the nearby cameras. As vehicles move through the city, the smart traffic cameras continuously capture real-time data, such as vehicle count, speed, and license plate recognition. This data is then used to generate and update the knowledge graph, which dynamically represents the traffic patterns and conditions in the city. Over time, the traffic patterns may change due to various factors, such as construction work, events, or changes in city planning. Some of the historical data captured by the traffic cameras on some devices may become less relevant for predicting and managing the current traffic conditions. To optimize the use of resources and improve the accuracy of traffic predictions, graph unlearning can be applied to the knowledge graph. Here, the concept of fine-tuning from an unlearning perspective becomes valuable, particularly when the devices aim to retain their existing data without adding more. Instead, these devices may selectively flag certain types of data for unlearning, allowing them to adapt to dynamic changes in their operating environment. The idea of personalized or customized unlearning strategies tailored to each individual device's needs further accentuates the versatility of graph unlearning.

6 Discussion and Future Directions

Graph unlearning is a promising and crucial field with the potential to address data privacy and security concerns in the realm of graph-based machine learning models [82]. As graph-based models continue to gain popularity for their capacity to capture intricate relationships and dependencies among data points in diverse domains, such as social network analysis, recommendation systems, and bioinformatics [13], the need for robust unlearning approaches becomes increasingly evident. The inherent nature of these models raises privacy issues, as they may accidentally store sensitive information about individuals or entities, jeopardizing user data confidentiality [34]. In Section 3, we presented a variety of graph unlearning approaches, including exact, approximate, and differential privacy-based methods proposed in recent years. These advancements cater to different aspects of graph unlearning, addressing specific subtypes such as node unlearning and edge unlearning. However, there is still a pressing need to advance this field in several new directions. We provide a few potential research directions and discuss them in the following.

6.1 Approximate Graph Unlearning for Non-Convex Settings

While remarkable progress has been made in graph unlearning, a notable area that warrants further exploration is the development of approximate unlearning methods tailored to non-convex settings. Currently, many existing approaches primarily focus on convex settings, where optimization landscapes are well-behaved and lend themselves to more straightforward solutions [20, 54]. However, real-world graph-based machine learning applications often involve non-convex settings with complex and irregular optimization landscapes. As such, designing effective approximate unlearning techniques for these scenarios is challenging but crucial. The need for approximate unlearning in non-convex settings stems from the realization that many practical graph-based models operate in environments where convexity assumptions do not hold. Examples include MPNN and other highly expressive models, which are capable of capturing intricate patterns and dependencies within the data [40, 38]. In these complex settings, unlearning approaches need to cope with non-convex optimization landscapes, which introduce additional intricacies and make the task significantly more demanding [2].

Tackling approximate unlearning in non-convex settings is essential for the wider adoption of graph-based machine learning in various real-world applications [10]. The ability to efficiently remove sensitive information and adjust models in response to evolving privacy requirements will instill confidence among users and stakeholders in utilizing graph-based machine learning solutions. While this direction poses challenges due to the inherent difficulty of approximating models in non-convex settings, it presents an exciting opportunity to push the boundaries of graph unlearning research [21]. Solutions that can effectively handle non-convexity while maintaining a balance between privacy preservation and model performance will be invaluable in applications ranging from social network analysis and recommendation systems to bioinformatics and beyond.

6.2 Graph Unlearning Beyond Simple Graphs

While the existing approaches have primarily focused on undirected graphs, many real-world scenarios involve more complex graph structures, such as directed, weighted, temporal, and knowledge graphs [55]. Directed graphs, in particular, are prevalent in applications involving causality, flow of information, or directed relationships [34]. Temporal graphs, on the other hand, capture time-varying relationships between entities, which are commonly found in dynamic systems and evolving networks [7]. Extending unlearning techniques to address these diverse graph types opens up a plethora of new applications. For instance, knowledge graphs, which model relationships and semantic information between entities, play a critical role in knowledge representation and reasoning tasks. Unlearning in knowledge graphs can offer valuable insights into privacy-preserving knowledge-sharing environments, ensuring that sensitive information is suitably removed while retaining the utility of the underlying knowledge [60]. Furthermore, directed and temporal graphs are prevalent in domains like social media analysis, financial transactions, and epidemiological studies, where understanding the directionality and evolution of relationships is essential. In these contexts, the ability to unlearn and protect sensitive information without compromising the temporal dynamics of the graph is of paramount importance [13]. Addressing graph unlearning in directed and temporal graphs requires devising novel algorithms and strategies tailored to their unique characteristics. Considering that these graph types exhibit rich and intricate structures, the challenges lie in developing efficient unlearning techniques that take into account causality, directed information flow, and temporal dependencies.

This research direction has the potential to revolutionize privacy and security measures in various graph-based machine learning applications. By enabling the unlearning of sensitive information in directed, weighted, temporal, and knowledge graphs, data privacy can be preserved while facilitating the development of more privacy-aware and responsible AI systems. The impact of such advancements is far-reaching, with applications spanning social network analysis, recommendation systems, epidemiology, finance, and more [79].

6.3 Universal Graph Unlearning Approaches

Universal graph unlearning represents a transformative direction in the field of graph unlearning, encompassing the capability to perform any unlearning task, including node, edge, feature unlearning, and graph-level unlearning. While the current approaches have made significant progress in addressing specific unlearning tasks, they often remain limited, specializing in one or a few types of unlearning tasks. The need to focus on designing universal approaches that can execute all unlearning tasks is paramount, considering that a majority of GNNs are designed to handle a wide range of tasks [12]. The versatility of universal graph unlearning offers profound benefits, as it caters to the diverse requirements of various graph-based machine learning applications. Different unlearning tasks address distinct aspects of data privacy and security in graphs. For instance, node unlearning is essential for preserving individual-level information, edge unlearning addresses the removal of specific relationships, feature unlearning ensures the protection of sensitive node attributes, and graph-level unlearning deals with privacy at a holistic level.

Designing universal approaches is challenging, as it necessitates accommodating the intricacies of multiple unlearning tasks within a single framework. Moreover, ensuring the efficiency and scalability of universal graph unlearning is vital to enable its application on large-scale graph datasets and complex graph structures [82]. By embracing the concept of universal graph unlearning, researchers can create a unified framework that seamlessly integrates with graph-based machine learning models, including GNNs with all graph unlearning tasks. This will foster the development of more comprehensive and privacy-aware AI systems, where data privacy can be adequately preserved across various unlearning tasks. Universal graph unlearning holds the potential to become a fundamental tool in enhancing data privacy and security measures, offering a holistic approach to address privacy concerns in graph-based machine learning.

6.4 Leveraging Graph Descriptors for Graph-level Unlearning

Another promising research direction is the exploration of graph descriptors or graph embedding methods when focusing solely on graph-level unlearning. Graph descriptors, also known as graph embeddings, represent graphs as vectors or low-dimensional representations in a continuous space [83]. They capture the structural information and relationships within the graph, enabling various machine learning algorithms to operate on graphs effectively. A significant advantage of using graph descriptors for graph-level unlearning is that once a graph embedding is computed, it remains fixed and unaffected by subsequent unlearning operations, as no further actions are needed. Moreover, majority of graph descriptors are extremely efficient to compute [44, 42]. This characteristic makes graph descriptors particularly well-suited for scenarios where frequent model updates or unlearning procedures may or may not be desirable.

Several graph descriptor methods have emerged in recent years, each with its unique characteristics. For further reading, we refer the reader to recent works on graph descriptors [41, 42, 44, 84, 83]. By incorporating graph descriptors into graph-level unlearning, researchers can take advantage of the rich representations they offer, allowing for efficient unlearning without the need to retrain the model repeatedly. The fixed nature of graph embeddings ensures that graph-level privacy-preserving operations do not interfere with the overall performance and generalization capabilities of the machine learning model. Exploring novel graph descriptor techniques tailored to graph-level unlearning requirements is an exciting avenue for future research. Additionally, comparative studies that evaluate different graph embedding methods in the context of unlearning tasks will provide valuable insights into their strengths and limitations. Moreover, since these descriptors are quite efficient, new embeddings could be quickly computed if node or edge-level removal requests are received.

7 Conclusion

This paper offers a comprehensive review of the emerging field of graph unlearning, which plays a vital role in responsible AI development. It details various graph unlearning approaches and methodologies, along with their applications in diverse domains, highlights the wide-ranging impact of this research area, including social networks, financial networks, and transportation systems. We have emphasized the sensitivity of GML to data privacy and adversarial attacks, underscoring the importance of graph unlearning techniques in addressing these critical concerns. Additionally, the connections drawn between graph unlearning and differential privacy underscore the significance of privacy-preserving techniques in the responsible design and deployment of AI systems. The comprehensive taxonomy, up-to-date literature overview, and evaluation measures provided in this survey empower researchers to explore new approaches and contribute to this evolving field. The suggested research directions offer exciting opportunities for future investigations, encouraging the development of novel algorithms, fortified privacy guarantees, and exploration of applications in emerging domains.

References

[1] Tom Michael Mitchell. *The discipline of machine learning*, volume 9. Carnegie Mellon University, School of Computer Science, Machine Learning . . . , 2006.

[2] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

[3] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE, 2015.

[4] Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*, 2022.

[5] Heng Xu, Tianqing Zhu*, Lefeng Zhang, Wanlei Zhou, and Philip S Yu. Machine unlearning: A survey. *ACM Computing Surveys*, 2023.

[6] Mark Newman. *Networks*. Oxford university press, 2018.

[7] Anwar Said, Roza G Bayrak, Tyler Derr, Mudassir Shabbir, Daniel Moyer, Catie Chang, and Xenofon Koutsoukos. Neurograph: Benchmarks for graph machine learning in brain connectomics. *arXiv preprint arXiv:2306.06202*, 2023.

[8] Albert-László Barabási. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987):20120375, 2013.

[9] Anwar Said, Timothy D Bowman, Rabeeh Ayaz Abbasi, Naif Radi Aljohani, Saeed-Ul Hassan, and Raheel Nawaz. Mining network-level properties of twitter altmetrics data. *Scientometrics*, 120:217–235, 2019.

[10] William L Hamilton. *Graph representation learning*. Morgan & Claypool Publishers, 2020.

[11] Anwar Said, Mudassir Shabbir, Saeed-Ul Hassan, Zohair Raza Hassan, Ammar Ahmed, and Xenofon Koutsoukos. On augmenting topological graph representations for attributed graphs. *Applied Soft Computing*, 136:110104, 2023.

[12] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[13] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[14] Anwar Said, Rabeeh Ayaz Abbasi, Onaiza Maqbool, Ali Daud, and Naif Radi Aljohani. Cc-ga: A clustering coefficient based genetic algorithm for detecting communities in social networks. *Applied Soft Computing*, 63:59–70, 2018.

[15] Saeed-Ul Hassan, Mudassir Shabbir, Sehrish Iqbal, Anwar Said, Faisal Kamiran, Raheel Nawaz, and Umar Saif. Leveraging deep learning and sna approaches for smart city policing in the developing world. *International Journal of Information Management*, 56:102045, 2021.

[16] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

[17] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. Graph unlearning. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 499–513, 2022.

[18] Eli Chien, Chao Pan, and Olgica Milenkovic. Efficient model updates for approximate unlearning of graph-structured data. In *The Eleventh International Conference on Learning Representations*, 2022.

[19] Yonatan Dukler, Benjamin Bowman, Alessandro Achille, Aditya Golatkar, Ashwin Swaminathan, and Stefano Soatto. Safe: Machine unlearning with shard graphs. *arXiv preprint arXiv:2304.13169*, 2023.

[20] Chao Pan, Eli Chien, and Olgica Milenkovic. Unlearning graph classifiers with limited data resources. In *Proceedings of the ACM Web Conference 2023*, pages 716–726, 2023.

[21] Jiali Cheng, George Dasoulas, Huan He, Chirag Agarwal, and Marinka Zitnik. Gnndelete: A general strategy for unlearning in graph neural networks. *arXiv preprint arXiv:2302.13406*, 2023.

[22] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

[23] Claudio Bettini. Privacy protection in location-based services: a survey. *Handbook of Mobile Data Privacy*, pages 73–96, 2018.

[24] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P Wellman. Sok: Security and privacy in machine learning. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 399–414. IEEE, 2018.

[25] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*, 2016.

[26] General data protection regulation 2016.

[27] California consumer privacy act 2018.

[28] Personal information protection and electronic documents act 2000.

[29] General personal data protection law 2018.

[30] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086, 2021.

[31] Stefanie Jegelka. Theory of graph neural networks: Representation and learning. *arXiv preprint arXiv:2204.07697*, 2022.

[32] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[33] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[34] Ameya Daigavane, Gagan Madan, Aditya Sinha, Abhradeep Guha Thakurta, Gaurav Aggarwal, and Prateek Jain. Node-level differentially private graph neural networks. *arXiv preprint arXiv:2111.15521*, 2021.

[35] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32, 2019.

[36] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3832–3842. PMLR, 13–18 Jul 2020.

[37] Ryoma Sato. A survey on the expressive power of graph neural networks. *arXiv preprint arXiv:2003.04078*, 2020.

[38] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[39] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[40] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[41] Anwar Said, Saeed-Ul Hassan, Suppawong Tuarob, Raheel Nawaz, and Mudassir Shabbir. Dgsd: Distributed graph representation via graph statistical properties. *Future Generation Computer Systems*, 119:166–175, 2021.

[42] Anwar Said, Saeed-Ul Hassan, Waseem Abbas, and Mudassir Shabbir. Netki: a kirchhoff index based statistical graph embedding in nearly linear time. *Neurocomputing*, 433:108–118, 2021.

[43] Nils M Kriege, Fredrik D Johansson, and Christopher Morris. A survey on graph kernels. *Applied Network Science*, 5(1):1–42, 2020.

[44] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, Alexander Bronstein, and Emmanuel Müller. Netlsd: hearing the shape of a graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2347–2356, 2018.

[45] Jie Wang, Zheng Yan, Jiahe Lan, Elisa Bertino, and Witold Pedrycz. Trustguard: Gnn-based robust and explainable trust evaluation with dynamicity support. *arXiv preprint arXiv:2306.13339*, 2023.

[46] Fernando Gama, Alejandro Ribeiro, and Joan Bruna. Diffusion scattering transforms on graphs. *arXiv preprint arXiv:1806.08829*, 2018.

[47] Diego Mesquita, Amauri Souza, and Samuel Kaski. Rethinking pooling in graph neural networks. *Advances in Neural Information Processing Systems*, 33:2220–2231, 2020.

[48] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[49] Kun Wu, Jie Shen, Yue Ning, and Wendy Hui Wang. Fast yet effective graph unlearning through influence analysis. 2022.

[50] Eli Chien, Chao Pan, and Olgica Milenkovic. Certified graph unlearning. *arXiv preprint arXiv:2206.09140*, 2022.

[51] Weilin Cong and Mehrdad Mahdavi. Grapheditor: An efficient graph representation learning and unlearning approach. 2022.

[52] Cheng-Long Wang, Mengdi Huai, and Di Wang. Inductive graph unlearning. *arXiv preprint arXiv:2304.03093*, 2023.

[53] Jiancan Wu, Yi Yang, Yuchun Qian, Yongduo Sui, Xiang Wang, and Xiangnan He. Gif: A general graph unlearning strategy via influence function. In *Proceedings of the ACM Web Conference 2023*, pages 651–661, 2023.

[54] Weilin Cong and Mehrdad Mahdavi. Efficiently forgetting what you have learned in graph representation learning via projection. In *International Conference on Artificial Intelligence and Statistics*, pages 6674–6703. PMLR, 2023.

[55] Xiangrong Zhu, Guangyao Li, and Wei Hu. Heterogeneous federated knowledge graph embedding learning and unlearning. In *Proceedings of the ACM Web Conference 2023*, pages 2444–2454, 2023.

[56] Guangxu Mei, Ziyu Guo, Shijun Liu, and Li Pan. Sgnn: A graph neural network based federated learning approach by hiding structure. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2560–2568. IEEE, 2019.

[57] Sina Sajadmanesh and Daniel Gatica-Perez. Locally private graph neural networks. In *Proceedings of the 2021 ACM SIGSAC conference on computer and communications security*, pages 2130–2145, 2021.

[58] Fan Wu, Yunhui Long, Ce Zhang, and Bo Li. Linkteller: Recovering private edges from graph neural networks via influence analysis. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2005–2024. IEEE, 2022.

[59] Iyiola E Olatunji, Thorben Funke, and Megha Khosla. Releasing graph neural networks with differential privacy guarantees. *arXiv preprint arXiv:2109.08907*, 2021.

[60] Meng Jiang, Taeho Jung, Ryan Karl, and Tong Zhao. Federated dynamic gnn with secure aggregation. *arXiv preprint arXiv:2009.07351*, 2020.

[61] Eli Chien, Wei-Ning Chen, Chao Pan, Pan Li, Ayfer Özgür, and Olgica Milenkovic. Differentially private decoupled graph convolutions for multigranular topology protection. *arXiv preprint arXiv:2307.06422*, 2023.

[62] Enayat Ullah, Tung Mai, Anup Rao, Ryan A Rossi, and Raman Arora. Machine unlearning via algorithmic stability. In *Conference on Learning Theory*, pages 4126–4142. PMLR, 2021.

[63] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR, 06–11 Aug 2017.

[64] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. 2022.

[65] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284, 2019.

[66] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE, 2014.

[67] Sina Sajadmanesh, Ali Shahin Shamsabadi, Aurélien Bellet, and Daniel Gatica-Perez. Gap: Differentially private graph neural networks with aggregation perturbation. In *USENIX Security 2023-32nd USENIX Security Symposium*, 2023.

[68] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

[69] Jun Zhou, Chaochao Chen, Longfei Zheng, Xiaolin Zheng, Bingzhe Wu, Ziqi Liu, and Li Wang. Privacy-preserving graph neural network for node classification. *arXiv preprint arXiv:2005.11903*, 2020.

[70] Yonatan Dukler, Alessandro Achille, Hao Yang, Varsha Vivek, Luca Zancato, Ben Bowman, Avinash Ravichandran, Charless Fowlkes, Ashwin Swaminathan, and Stefano Soatto. Introspective cross-attention probing for lightweight transfer of pre-trained models. *arXiv preprint arXiv:2303.04105*, 2023.

[71] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

[72] Iyiola E Olatunji, Wolfgang Nejdl, and Megha Khosla. Membership inference attack on graph neural networks. In *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pages 11–20. IEEE, 2021.

[73] Yang Liu, Mingyuan Fan, Cen Chen, Ximeng Liu, Zhuo Ma, Li Wang, and Jianfeng Ma. Backdoor defense with machine unlearning. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, pages 280–289. IEEE, 2022.

[74] Xuefeng Li, Yang Xin, Chensu Zhao, Yixian Yang, and Yuling Chen. Graph convolutional networks for privacy metrics in online social networks. *Applied Sciences*, 10(4):1327, 2020.

[75] Nicholas Proferes, Naiyan Jones, Sarah Gilbert, Casey Fiesler, and Michael Zimmer. Studying reddit: A systematic overview of disciplines, approaches, methods, and ethics. *Social Media+ Society*, 7(2):20563051211019004, 2021.

[76] Christian Fuchs. Web 2.0, prosumption, and surveillance. *Surveillance & Society*, 8(3):288–309, 2011.

[77] Paul Glasserman and H Peyton Young. Contagion in financial networks. *Journal of Economic Literature*, 54(3):779–831, 2016.

[78] Kyriaki G Giota and George Kleftaras. Mental health apps: innovations, risks and ethical considerations. *E-Health Telecommunication Systems and Networks*, 2014, 2014.

[79] Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, et al. Eta prediction with graph neural networks in google maps. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3767–3776, 2021.

[80] Hongxin Hu, Gail-Joon Ahn, and Jan Jorgensen. Detecting and resolving privacy conflicts for collaborative data sharing in online social networks. In *Proceedings of the 27th Annual Computer Security Applications Conference*, pages 103–112, 2011.

[81] Jiamin Fan, Kui Wu, Yang Zhou, Zhengan Zhao, and Shengqiang Huang. Fast model update for iot traffic anomaly detection with machine unlearning. *IEEE Internet of Things Journal*, 2022.

[82] Chao Pan, Eli Chien, and Olgica Milenkovic. Unlearning nonlinear graph classifiers in the limited training data regime. *arXiv preprint arXiv:2211.03216*, 2022.

[83] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE transactions on knowledge and data engineering*, 30(9):1616–1637, 2018.

[84] Saurabh Verma and Zhi-Li Zhang. Hunt for the unique, stable, sparse and fast feature learning on graphs. *Advances in Neural Information Processing Systems*, 30, 2017.