Sharing Uncertain Graphs Using Syntactic Private Graph Models

Dongqing Xiao, Mohamed Y. Eltabakh, Xiangnan Kong

Computer Science Department, Worcester Polytechnic Institute
Worcester, United States of America
{dxiao, meltabakh, xkong}@wpi.edu

***Abstract*—Research on social and business applications requires open access to real datasets. Such datasets can be shared, generally in the form of *uncertain graphs* whose edges are labeled with a probability of existence. While releasing uncertain graphs often risks exposing sensitive user information to the public. Unfortunately, current works for privacy preserving graph releasing only target deterministic ones and overlook the inherent uncertainty in edges.**

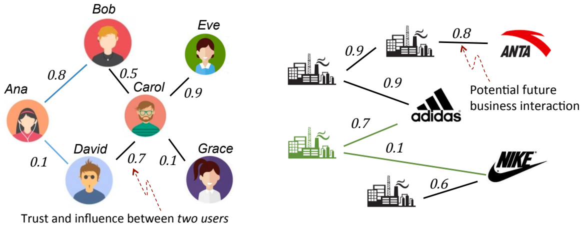
To overcome such limitation, our work seeks a solution to release *uncertain graphs* with high utility while preserving privacy. We show that simply combining the representative extraction strategy and conventional graph anonymization method will results in the addition of noise that significantly disrupts *uncertain graph* structure, making it unsuitable for the further study. Instead, we introduce an uncertainty-aware approach that provides identical privacy guarantees with much less noise. It enables fine-grained control over noise injected the *original* uncertain graph. In particular, we introduce a reliability-based metric for utility loss evaluation, and propose two uncertain-aware schemes including reliability-sensitive edge selection and anonymity-oriented edge alteration. Finally, we apply our approach to real *uncertain* graphs and show that it produces anonymized *uncertain* graphs that closely match the originals in important graph structure metrics.

I. Introduction

In many prevalent application domains, such as business to business (B2B) [23], social networks [2], [20], and sensor networks [37], graphs serve as powerful models to capture the complex relationships inherent in these applications. Most graphs in these applications are uncertain by nature, where each edge carries a degree of uncertainty (probability) representing the probability of its presence in the real world. This uncertainty can be due to various reasons ranging from the use of prediction models to predict the edges (as in social media and B2B networks) to physical properties that affect the edges' reliabilities (as in sensor and communication networks).

These uncertain graphs are of significant importance to support various data mining tasks *e.g.*, understanding graph structures [9], [21], social interactions [12], information discovery and propagation [38], advertising and marketing [20], among many others. When compared to sharing the results of data mining, data publishing gives greater flexibility for unlimited analysis, wider data exploration. However, the publishing of these uncertain graphs might violate participants' privacy.

Motivation Scenario I (Social Trust Networks): *In social networks, the trust and influence relationships among users—which may greatly impact users' behaviors—are usually probabilistic [20] (See Figure 1(a)). The existence of the trust relationship depends on many factors, such as the area of expertise and emotional connections. Studying the structure of* real *social trust networks cam produce insights on product promotion, information dissemination. However, the release of such uncertain graphs with simple anonymization may cause serious privacy issues. Equipped with prior knowledge, the attackers can re-identify private and sensitive information, such as the identity of the users and their trustiness relationship, from the released data.*



(a) Social Trust Network (b) B2B Network

Figure 1: Examples of real-world uncertain graphs with privacy concerns.

Motivation Scenario II (B2B Networks): *Another example comes from Businesses to Businesses network (See Figure 1(b)). In these networks, e.g., "Alibaba", nodes represent companies (or business in general) while edges represent the trust and the likelihood of future transactions among them [23]. Such future interactions are probabilistic and obtained by prediction models. The historical transaction releasing is forbidden by legal regulation. B2B networks can be analyzed and mined for various applications including advertisement targeting [1] and customer segmenting [3]. Certainly, information about a company's transactions with other companies is considered sensitive data since any leak can be used to infer the company's financial conditions.*

These scenarios show the immediate need for efficient methods privacy preserving uncertain graph publishing, where sanitation or anonymization is applied to the input uncertain graph before publishing. Despite the number of graph anonymization techniques have been proposed [7], [10], [24], [26], the ignorance of edge uncertainty makes them inefficient for uncertain graph sanitation task. The inefficiency can be due to various reasons ranging from the wrong assumption of privacy attacks to improper utility loss metrics. We face three key challenges in shifting conventional methods to privacy protection on *uncertain graphs*.

• **Appropriate Utility Loss Metric for Uncertain Graphs:** Ideally, the anonymized graph should preserve the privacy with the smallest utility loss for further analysis tasks. Thus, it is crucial to understand and model the utility loss of *uncertain graph* being published through well-defined metrics. There have been many attempts such as Graph Edit Distance [24], Spectrum Discrepancy [36], Community Reconstruction Error [34] and Shortest Path Discrepancy [25]. Most of them are heavily tailored towards for deterministic graphs and built on the top of *deterministic* graph concepts. Thus, they are unreasonable when dealing with uncertain graphs. What is the appropriate metric? How is it integrated into anonymization process as replacement of conventional ones? These questions should be carefully addressed when dealing with uncertain graph anonymization.

• **Increased Exponential Complexity of Uncertain Graph Anonymization:** The problem of k-anonymizing a given deterministic graph by as few graph contractions (edge addition, edge deletion, vertex addition and vertex deletion) as possible is shown to be NP-hard [16]. Existing techniques usually rely on heuristics to avoid combinatorial intractability. In uncertain graphs, the problem is even harder since an edge operation is no longer a binary operation (addition or deletion), but there can be infinite probability values that can be assigned to each edge. Clearly, efficient uncertainty-aware heuristics should be developed to bring the solution to the realm of feasiblity.

In this paper, we present a principled extension of graph anonymization techniques in the presentence of uncertainty, called Chameleon. Chameleon incorporates edge uncertainties into the core of the anonymization processing such as evaluations of privacy gain and utility loss. In practicular, we propose a new utility metric based on the *reliability* measure—which is a core metric in numerous uncertain graph applications [4], [15], [38]. The anonymization process need to change the graph structure by modifying the edge probabilities of a subset of the edges, which is an exponential search space. Therefore, we propose a ranking algorithm that ranks the edges w.r.t the impact of a change on the graph structure—which we refer to as *"reliability Relevance"*—and that ranking will guide the edge selection process. Moreover, we propose a theoretically-founded probability-alteration strategy based on the entropy of graph degree sequence, which enables achieving maximum privacy gain for an added amount of perturbation.

In summary, the key contributions of this paper are the following:

- Identifying the new and important problem of uncertain graph anonymization where edge uncertainties need to be seamless integrated into the core of the anonymization process. Otherwise, either the privacy will not be protected or the utility will be severely damaged.
- Proposing a new utility-loss metric based on the solid connectivity-based graph model under the possible world semantics, namely the *reliability discrepancy* (Section III).
- Introducing a theoretically-founded criterion, called *reliability relevance*, that encodes the sensitivity of the graph edges and vertices to the possible injected perturbation. The criterion will guide the edges' selection during the anonymization process (Section V).
- Proposing uncertainty-aware heuristics for efficient edge selection and noise injection over the input uncertain graph to achieve anonymization at a slight cost of reliability (Section V).
- Building the Chameleon framework that integrates the aforementioned contributions. Chameleon is experimentally evaluated using several real-world datasets to evaluate its effectiveness and efficiency. The results demonstrate a significant advantage over the conventional methods that do not directly consider edge uncertainties (Section VI).

II. Related Works

A significant amount of prior work has been done protecting the privacy of network datasets. We summarize them here and clarify our privacy goals in this paper.

Syntactic Privacy. Early works on privacy-preserving network publishing mainly focus on developing anonymization techniques for deterministic graphs. Their goal is to *publish* the data in an anonymized manner without making any assumptions of the type of analysis and queries that will be executed on the release one. Once the data is published, it is available for any type of analysis. Most of them leverage *Syntactic* privacy models derived from k-anonymity [32] to create k identical neighborhoods, or k identical degree nodes. Following this path, many graph anonymization techniques have been proposed.

Graph Anonymization Techniques. Current methods for anonymizing "graphs" can be classified into four main categories: (1) Clustering-based generation [6], [17], [18]; (2) *Edge modification* [24], [31], [34], [35], [39], (3) *Edge randomization* [25], [28], [36], and (4) *Uncertainty semantic-based modifications* which add uncertainty to some edges and thus converting the graph to an uncertain version [7], [27]. The uncertainty semantic-based approaches transform the original deterministic graph into an uncertain one to be published. These techniques are known as the state-of-art ones because of their excellent privacy-utility tradeoff brought by the fine-grained perturbation leveraging the uncertain semantics. To the best of our knowledge, these techniques are tailored to *deterministic* graphs (unweighted & weighted) that overlook edge uncertainty.

Diffiential Privacy. Another avenue is to apply differential privacy to network data. It roughly falls into two directions. The first direction aims to release certain differentially private *data mining results*, such as degree distributions, sub-graph counts and frequent graph patterns. Such methods that release only query results require tracking the results: early uses of the data can affect the quality of later uses, thus no new queries can be permitted on the data. The second direction aims to publish a sanitized graph. Most research in this direction projects an input graph to dK-series and ensures differential privacy on dK-series statistics. These private statistics are then either fed into generators or MCMC process to generate a fit Syntactic graphs. While current techniques are still inadequate to provide desirable data utility for many graph mining tasks.

Our Goal: Data Model & Privacy Policy. In this work, we study the problem of privacy preserving *uncertain* graph publishing. Conceptually, the problem can be interpreted as a natural generalization of the *determinitic* graph contexts to a larger probabilistic context, with the anonymization process being specifically optimized.

Here, we remind the reader the approach that first casting the probability of every edge into a weight then applies existing anonymization methods on this weighted graph to attain the anonymized uncertain graph is problematic. First, there is no meaningful way to perform such casting. The casting has been proven to be erroneous in various uncertain graph mining tasks [30], [38]. Second, there is no principled way to additionally encode normal weights on the edge. For example, each link in the road network can be weighted indicating the distance or travel time between them, and a probability can be assigned to model the likelihood of a traffic jam [19]. In summary, existing strategies for weighted graphs anonymization cannot be applied to *uncertain* graphs.

In the context of privacy preserving graph publishing, we can choose to adopt the Syntactic or differential privacy policy. ϵ-differential privacy does relate to individual identifiability and provides strong privacy guarantee without making any assumption of privacy risks. However, there is no clear way to set a general policy for a value ϵ that provides sufficient privacy [22]. In contrast to ϵ-differential privacy, Syntactic privacy model can generally be defined and understood based on the data schema; parameters have a clear privacy meaning that can be understood independent of the actual data and have a clear relationship to the legal concept of individual identifiability of data. In this work, we choose k-obfuscation, a variant of k-anonymity as the basis of our privacy policy for uncertain graph *publishing*.

III. Problem Definition

In this section, we present the notations, definitions and the problem formulation.

A. Uncertain Graph

An uncertain graph $\mathcal{G} = (V, E, p)$, is defined over a set of nodes V, a set of edges E, and a set of probabilities p of edge existence. Following the literature, we consider the edge probabilities independent [19], [30], [38], and we assume *possible-worlds* semantics [13]. Specifically, the *possible world* semantics interprets $\mathcal{G}$ as a set of possible deterministic graphs $W(\mathcal{G}) = \{G_1, G_2, ..., G_n\}$, where each deterministic graph $G_i \in W(\mathcal{G})$ includes all vertices of $\mathcal{G}$ and a subset of edges $E_{G_i} \subset E$. The probability of observing any possible world $G_i = (V, E_{G_i}) \in W(\mathcal{G})$ is

$$Pr[G_i] = \prod_{e \in E_{G_i}} p(e) \prod_{e \in E \setminus E_{G_i}} (1 - p(e))$$

In this work, we assume the input uncertain graph undirected and contains no self-loops or multiple edges.

B. Reliability-Based Utility Loss Metric

A well-chosen utility-loss metric may lead to substantially less sanitized graphs at a minimal loss of information. As be known to all, connectivity is a fundamental graph property and plays an important role in graph mining tasks such as locating $k-$nearest neighbor [30], graph clustering [4] and shortest paths detecting [38]. Moreover, the connectivity model has been shown to be able to yield better representation than degree sequence model. The connectivity discrepancy was proven to be a proper utility-loss metric. Therefore, we use its generalized version – Reliability Discrepancy as the utility-loss metric in the uncertain graph context.

In uncertain graphs, the concept of reliability is used to generalize *connectivity* by capturing the probability that two given (sets of) nodes are reachable over all possible worlds of the uncertain graph as follows:

Definition 1. ***Two-Terminal Reliability [13]*** *Given an uncertain graph $\mathcal{G}$, and two distinct nodes u and $v \in \ V$, the reliability of (u, v) is defined as:*

$$R_{u,v}(\mathcal{G}) = \sum_{G \in W(\mathcal{G})} \mathcal{I}_G(u, v) Pr[G]$$

where $\mathcal{I}_G(u, v)$ is 1 iff u and v are contained in a connected component in G, and 0 otherwise.

Definition 2. ***Graph Reliability Discrepancy*** *The reliability discrepancy of graph $\tilde{\mathcal{G}} = (V, E, \tilde{p})$, denoted as $\Delta(\tilde{\mathcal{G}})$, w.r.t. an original graph $\mathcal{G} = (V, E, p)$ is defined as the sum of the two-terminal reliability discrepancy over all node pair $(u, v) \in V_{\mathcal{G}}$.*

$$\Delta(\tilde{\mathcal{G}}) = \sum_{(u,v) \in V_{\mathcal{G}}} |R_{u,v}(\mathcal{G}) - R_{u,v}(\tilde{\mathcal{G}})|$$

C. Attack Model and Privacy Policy

In this paper, we focus on the "identity disclosure problem" [24] over uncertain graphs, which is one serious privacy leak concern when a graph dataset is published. Formally, give a published graph G, if and adversary can locate the target entity t as a vertex v of G with a high probability via auxiliary information, we said that the identity of t is disclosed. The popular assumption of auxiliary information is node degree [24].

Following the literature, we adopt the syntactic (k, ϵ)-*obfuscation* criterion for privacy guarantee. Analogous to the well known k-anonymity notion, k-obf requires to blend every vertex with *other* fuzzy-matching nodes. Compared to k-anonymity, k-obf, which is global and entropy-based quantification, is more adequate than the previous used local quantification based on a posteriori belief probabilities. An

Table I: Characteristics of the datasets and privacy parameters

Graph	Nodes	Edges	Edge Prob	Tolerance level
DBLP	824,774	5,566,096	0.46	10^{-4}
BRIGHTKITE	58,228	214,078	0.29	10^{-3}
PPI	12,420	397,309	0.29	10^{-2}

excellent discussion on k-obf was presented by Bonchi *et al.* [10]. Moreover, the introduction of a tolerance parameter ϵ, which allows skipping up to $\epsilon * |V|$ nodes, makes it more practical. The skipped nodes might be extreme unique nodes, e.g., Trump in a Twitter network, whose obfuscation is almost impossible. The formal definition is as follows:

Definition 3. ***(k, ϵ)-obf [7]*** *Let P be a vertex property (i.e., vertex degree in our work), $k \geq 1$ be a desired level of anonymity, and $\epsilon > 0$ be a tolerance parameter. An sanitized uncertain graph $\tilde{\mathcal{G}}$ is said to k-obfuscate a given vertex $v \in \mathcal{G}$ w.r.t P if the entropy $H()$ of the distribution $Y_{P(v)}$ over the nodes of $\tilde{\mathcal{G}}$ is greater than or equals to $\log_2 k$:*

$$H(Y_{P(v)}) \geq \log_2 k.$$

The uncertain graph $\tilde{\mathcal{G}}$ is (k, ϵ)-obf w.r.t property P if it k-obfuscates at least $(1 - \epsilon)|V|$ nodes in $\mathcal{G}$.

D. Problem Statement

Given the above foundation, we can now formulate the addressed problem.

Problem 1. ***Reliability-Preserving Uncertain Graph Anonymization*** *Given an uncertain graph $\mathcal{G} = (V, E, p)$ and anonymization parameters k and ϵ, the objective is to find a (k, ϵ)-obfuscated uncertain graph $\tilde{\mathcal{G}} = (V, E, \tilde{p})$ with minimal $\Delta(\tilde{\mathcal{G}})$. That is:*

$$\begin{aligned} &\underset{\tilde{\mathcal{G}}}{\operatorname{argmin}} && \Delta(\tilde{\mathcal{G}}) \\ &\textit{Subject to} && \tilde{\mathcal{G}} \textit{ is } (k, \epsilon) - obf \end{aligned}$$

IV. Benchmark Solution

Before jumping to the design of new methods for *uncertain* graphs, we first consider somehow applying existing methods designed for deterministic graphs. In this work, we introduce the representative anonymization (Rep-An) algorithm that combines isolated but complementary work from literature for uncertain graph anonymization.

As shown in Figure2, we first extract a single *representative* instance from an original uncertain graph. Then, conventional anonymization techniques can be then applied on this representative to attain closely approximate anonymized output of the original uncertain graph. Meanwhile, there has been extensive work on extracting a single representative instance (deterministic one) of uncertain graphs that capturing graph statistics such as the expected vertex degrees [29]. This body of research comes to its aid that anonymization can be carried out on uncertain graphs, regardless of the uncertainty inherent in the data.

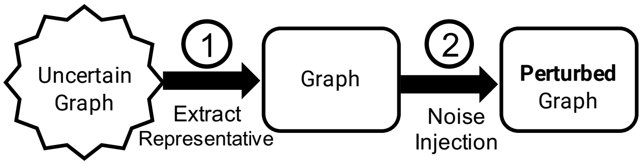
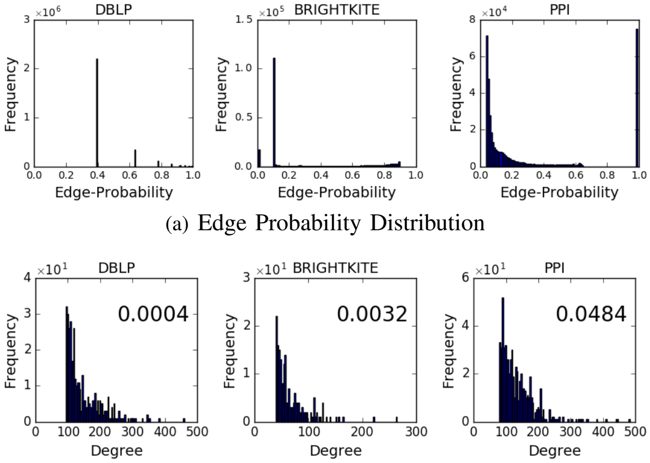


Figure 2: Overview of Rep-An. Noise is added to the extracted *representative* instance.



(a) Edge Probability Distribution

(b) Expected Degree Distribution

Figure 3: Edge probability& degree distributions.

However, this approach has several limitations. First, the input edge uncertainties (probabilities) are no longer integrated into the anonymization process since they are detached from the graph in the first step. Second, the anonymization process (the second step) is oblivious to the *reliability* metric since its input is a made-up deterministic graph. Third, since the two phases are isolated from each other, different phases are optimized for different metrics. As the result, this naive `Rep-An` approach introduces a high level of noise and consequently deteriorates the overall utility of the anonymized graph. In the experiment section, we further study this approach empirically and confirm its impracticality.

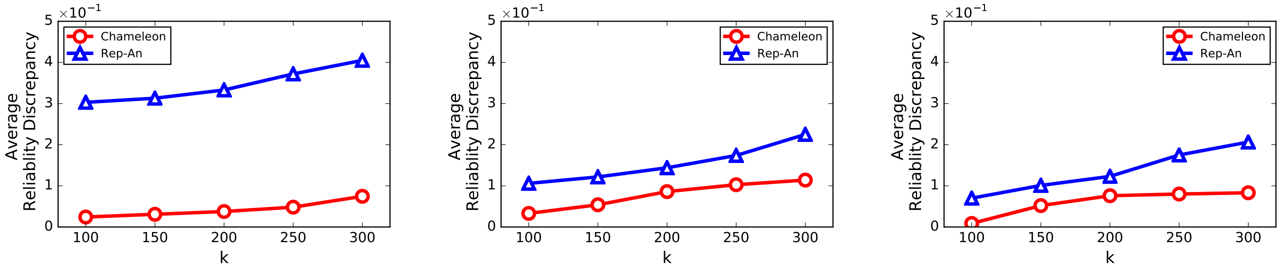
A. Validation on Real Uncertain Graphs

In this section, we empirically evaluate the impact of noise injected to extracted *representative* instances by executing Rep-An on real uncertain graphs.

Uncertain Graph Collections. We use three uncertain graphs that capture different real-world scenarios and have been used in prior uncertain graph mining studies. Table I lists uncertain graphs and their tolerance parameters used in our evaluation.

DBLP is a co-authorship network where the probability of an edge between two authors represents the likelihood two authors will collaborate in the future. The probability is obtained by a predictive model based on historical co-authorship data. [19].

BRIGHTKITE is a location-based social network where the probability of an edge between two users corresponds to the chance that two users visit each other. The probability can be obtained by a prediction model based on historical data [12].



(a) DBLP

(b) BRIGHTKITE

(c) PPI

Figure 4: The structural distortion introduced for different privacy levels quantifies as the average reliability discrepancy between the original ones and perturbed ones.

PPI is a dataset of protein-protein interactions, provided by Disease Module Identification DREAM Challenge, where the probability of any edge corresponds to the confidence that the interaction actually exists. The probability is obtained through biological experiments.

Figure 3(a) shows the edge-probability distribution in these three datasets. Note that the DBLP dataset only has a few probability values distributed in $[0, 1]$, while Brightkite dataset's probability values are generally very small. The PPI dataset has a more uniform probability distribution. We also present their degree distributions of "unique" nodes (with high degree and obfuscation level is smaller than 300). Observe that, all the three graphs have a heavy-tailed degree distribution (i.e, an amount of "unique" nodes). In the context of graph privacy, the larger amount of "unique" nodes requires a larger amount of noise.

Computation. We first extract the single representative instances for each uncertain graph, introduce noise using the state-of-art graph anonymization approach [7], and then compute the Reliability Discrepancy between the perturbed graph and the original uncertain graph as a measure of the level of graph structural error introduced. We approximate its expectation value by the average value obtained over the sampled possible worlds. Here, we use 1,000 samples since it has been shown that 1000 usually suffices to achieve accuracy converge [30].

Results. Figure 4 shows that the Rep-An algorithm produces a large error for large values of k (i.e. strong privacy guarantees). As we mentioned, the low level of utility is due to the large noise Rep-An injects into the representative instance, resulting in a perturbed graph (uncertain one) which is significantly different from the original *uncertain* graph.

To better understand the limitation of Rep-An, we report the potential lower-bound on the error that could be achieved via Chameleon methods in Figure 4. Indeed, Figure 4 shows that the utility loss of the Rep-An strategy can largely be attributed to the overlook of edge uncertainty. The high discrepancy is largely due to the representative extraction step. In the extreme case where $k = 100$ (i.e. a weak privacy guarantee), the sole representative extraction step produces high-reliability errors (i.e, structural distortion). The DBLP graph shows a more evident gap. For small k values, i.e., $k = 100$ and $k = 150$, the largest error is within 30% from the original graph values. The DBLP graph shows a more evident gap because a large amount of edge probability altered in the representative extraction step. Thus, it can produce cumulative structural changes.

V. Privacy Via Chameleon

The results in the previous section demonstrate the huge utility loss in the perturbed representative instance after adding noise to provide privacy guarantee. In this section, we propose a novel uncertainty-aware algorithm called Chameleon that enables uncertainty-aware control over the noise injected into the *original uncertain* graph. This qualifies Chameleon to provide enough privacy guarantee in better utility.

A. The Chameleon framework

Algorithm 1 Chameleon Iterative Skeleton

```
Input: Uncertain graph $\mathcal{G}$, adversary knowledge $\mathcal{K}$, obfuscation level $k$, tolerance level $\epsilon$, size multiplier $c$ and white noise level $q$
Output: The anonymized result $\tilde{\mathcal{G}}_{obf}$
$\sigma_l \leftarrow 0$; $\sigma_u \leftarrow 1$
repeat
    $\langle \tilde{\epsilon}, \tilde{\mathcal{G}} \rangle \leftarrow \texttt{GenObf}(\mathcal{G}, k, \epsilon, c, q, \sigma_u, \mathcal{K})$
    if $\tilde{\epsilon} = 1$ (fail) then $\sigma_l \leftarrow \sigma_u$; $\sigma_u \leftarrow 2\sigma_u$
until $\tilde{\epsilon} \neq 1$
repeat
    $\sigma \leftarrow (\sigma_u + \sigma_l)/2$
    $\langle \tilde{\epsilon}, \tilde{\mathcal{G}} \rangle \leftarrow \texttt{GenObf}(\mathcal{G}, k, \epsilon, c, q, \sigma_u, \mathcal{K})$
    if $\tilde{\epsilon} = 1$ then $\sigma_l \leftarrow \sigma$
    else $\sigma_u \leftarrow \sigma$;   $\tilde{\mathcal{G}}_{obf} \leftarrow \tilde{\mathcal{G}}$
until $\sigma_u - \sigma_l$ is enough small
return $\tilde{\mathcal{G}}_{obf}$
```

We now introduce the state-of-art perturbation algorithm [7] that computes the noise needed to injected into the input *deterministic* graph to obtain the desired privacy level. Each selected edge is altered based on a stochastic variable drawn from a trunated normal distribution, $R(\sigma)$. This distribution has density function proportional to the normal distribution, with mean 0 and variance σ^2. Thus, small values of σ contribute towards better utility, but at the same time they provide lower level of obfuscation. Targeting for high utility, the algorithm aims at injecting the minmal amount of noise need to achieve the required obfuscation. Its computation is achieved via a binary search on the value of the noise parameter σ, as shown in Algorithm 1.

The binary search flow is determined by the `GenObf` function. The function `GenObf` aims at finding a (k, ϵ)-*obfuscation* using a given noise parameter σ. It returns a pair $\langle \tilde{\epsilon}, \tilde{\mathcal{G}} \rangle$, where $\tilde{\epsilon} = 1$ or $\tilde{\epsilon} \leq \epsilon$. In the first case, all the attempts fail. In the latter cases, $\tilde{\mathcal{G}}$ is a (k, ϵ)-*obfuscation*. The function `GenObf` find obfuscation candidate in a randomized way, t attempts are performed. Each attempt performs following core steps:

- Select a subset of edges subjects to further alteration;
- Alter selected edges as the computed amount of noise;
- Check the solution with/out enough privacy guarantee;

Contribution. The conventional schemes are plausible if the operating edge probability is binary, which is unreasonable when dealing with uncertain graphs. Our goal is to develop a anonymization mechanism that reduced the amount of noise that must be added to achieve a given privacy level for *uncertain* graphs. Our insight is to shift an existing framework by integrating uncertainty semantics into core steps such as edge selection & alteration.

B. Uncertainty-aware Edge Selection

Figuring out the optimal subset of edges that balances the privacy gain and the utility loss is a typical combinational optimization problem. It involves the consideration over the exponential number of edge combinations. Let alone the infinite possibilities of probability values on the selected edges, which further complicates the problem.

To alleviate combinational intractability, the heuristics that have been proposed so far for anonymizing deterministic graphs can be classified into two main categories: (1) *Anonymity-oriented* heuristics that suggest injecting larger perturbations to the edges associated with the less-anonymized (more-unique) nodes [7], [24], [33], [39], and (2) *Utility-oriented* heuristics that suggest avoiding perturbations over *"bridge"* and sensitive edges whose deletion or addition would significantly impact the graph structure [11], [25], [28], [36]. Note that these two types are complementary to each other and combining them would introduce an added benefit as confirmed by practice in deterministic graph anonymization [11]. Nevertheless, these two types of heuristics and their combination have not been explored yet in the context of *uncertain* graphs.

In this paper, we first extend the idea of *uniqueness* score via density estimation. Second, we propose a novel edge relevance that extends well-known graph concepts, such as "cut edge" for estimating structural errors incurred by edge probability alterations. In order to compute edge relevance in *uncertain* graphs, we design an algorithm based sampling. Finally, we utilize these *uncertainty*-embedded meta-heuristics to effectively selecting edges for further alteration.

C. Uniqueness Score

Intutively, larger amount of noise should be added at nodes that are less anonymized (i.e, more distinctive) in the original graphs. *Uniqueness score* is shown to be an effective metric of how typical is the node v among the nodes of the graph [7]. The formal defintion of uniqueness score is given as follows.

Definition 4. ***Uniqueness Score [7]*** *Let $P : V \rightarrow \Omega_P$ be a property on the set of nodes V of the graph $\mathcal{G}$, let d be a distance function on Ω_P, and let $\theta > 0$ be a parameter. Then the $\theta - commonness$ of the property values $\omega \in \Omega_P$ is $C_\theta(\omega) := \sum_{u \in V} \Phi_{0,\theta}(d(\omega, P(v)))$, while the θ-uniqueness of $\omega \in \Omega_P$ is $U_\theta := \frac{1}{C_\theta(\omega)}$.*

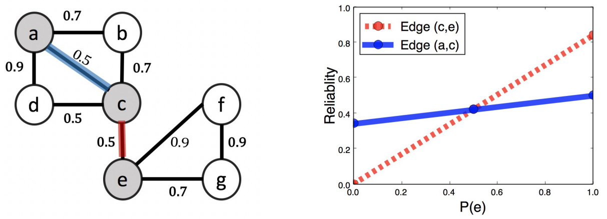
Note that, it adopts a parametric way to estimate the probability density function of the property value ω (i.e., how typical the value is among all the nodes). For the density estimate, we place a normal kernel with standard variance θ which implies the spread out of the property value over the domain. In the previous work [7], they set $\theta = \sigma$, where σ is the standard deviation of the noise generation Gaussian distribution. This is because the larger injected amount of uncertainties indicates that the property values may be spread out in a larger domain.

Here, we set θ equals σ_G as the latter represents the spread of property value in the *uncertain graph*. For a given node with the property ω, we can compute its "uniqueness" score as the reverse of its density. The higher the uniqueness score the less-protected the node and the more anonymization it eventually needs.

D. Reliability Relevance

Observation Clearly, alteration over an single uncertin edge (partial added or deleted) can produce structural changes that send ripples through the rest of the graph. The same amount of alteration performed over different edges may incur significantly different structure change. Referring to the example in Figure 5(a), two vertices $\mathtt{a}$ and $\mathtt{e}$ will be assigned the same *uniqueness score* due to the exact probabilities associated with their edges. As a result, the *anonymity-oriented* strategy would select and perturb either of the two edges $(\mathtt{a}, \mathtt{c})$ and $(\mathtt{c}, \mathtt{e})$ with the same probability. However, the modifications to $(\mathtt{c}, \mathtt{e})$—which is the only link between two *reliable* clusters—clearly incurs much larger structure distortion than that on $(\mathtt{a}, \mathtt{c})$.To control the utility loss, it is cruical to quantify the structural impact of a single edge unit alteration.

Contribution. To this end, we propose a theoretically sound estimation for the *"reliability deviation"* caused by individual uncertain edge modifications in a fine-grained way. Following this path, we introduce a new measure, called *Edge Reliability Relevance* ($\mathcal{E}RR$), at the edge level, and an aggregated measure, called *Vertex Reliability Relevance* ($\mathcal{V}RR$), at the vertex level as will be formally defined next. These measures will enable ranking the edges to be targeted for obfuscation in a meaningful utility-based perspective. Besides, we provide an algorithm for computing them efficiently.



(a) An uncertain graph (b) The reliability $R_{a,e}$ v.s $p(e)$

Figure 5: (a) Different edges have different effect on the overall graph reliability. (b) Formal *Reliability Relevance* measure. A bigger edge's slope indicates big distortion in reliability under small changes to the edge's probability.

Edge Relevance Analysis. In the context of uncertain graphs, the reliability relevance of edge is defined as reliability discrepancy when one single edge is alterred. Changing a single edge in $\mathcal{G}$ will result in one or more point-wise reliability changing in the corresponding *uncertain* graphs. Thus, the edge reliability relevance is computed as the sum of reliability deviation caused by a single edge alteration.

Definition 5. ***Single Edge Reliability Relevance.*** *Given an uncertain graph, the sensitivity of two-terminal reliability $R_{u,v}$ over edge e is defined as follows:*

$$\mathcal{E}RR^{e}_{u,v} = \lim_{h \to 0} \frac{R_{u,v}(\mathcal{G}') - R_{u,v}(\mathcal{G})}{h}$$

where $\mathcal{G}'$ are identical to the original graph expect the edge e with modified probability $p(e) + h$.

Lemma 1. ***Factorization Lemma*** *Given an uncertain graph $\mathcal{G}$, the reliability of the node pair (u, v), i.e., $R_{u,v}(\mathcal{G})$, can be factorized via a specific uncertain edge e as follows:*

$$\begin{aligned} R_{u,v}(\mathcal{G}) &= p(e)R_{u,v}(\mathcal{G}_e) + (1 - p(e))R_{u,v}(\mathcal{G}_{\bar{e}}) \\ &= p(e)\left[R_{u,v}(\mathcal{G}_e) - R_{u,v}(\mathcal{G}_{\bar{e}})\right] + R_{u,v}(\mathcal{G}_{\bar{e}}) \end{aligned}$$

where uncertain graphs $\mathcal{G}_e$ and $\mathcal{G}_{\bar{e}}$ are identical to the original graph $\mathcal{G}$ with the exception that e is certainly present in the former and certainly not present in the later.

Lemma 1 indicates that the *deviation* of $R_{u,v}$ introduced by a specific edge e is *linear* to the amount of edge probability deviation as shown in Figure 5(b). In other words, the edge reliability relevance $\mathcal{E}RR^{e}_{u,v}$ equals to the difference of $R_{u,v}$ in the corresponding neighbor *uncertain graphs* $\mathcal{G}_e$ and $\mathcal{G}_{\bar{e}}$. First, we remind the reader that edge reliability relevance always be positive since all the connected pairs in $\mathcal{G}_e$ are guaranteed to be a superset or at least equal to that in $\mathcal{G}_{\bar{e}}$.

By aggregating edge relevance among all the node pairs u, v, we can get the overall *reliability relevance* of edge e $\mathcal{E}RR^{e}(\mathcal{G})$ defined as

$$\begin{aligned} \mathcal{E}RR^{e}(\mathcal{G}) &= \sum_{u,v} |\mathcal{E}RR^{e}_{u,v}(\mathcal{G})| \\ &= \sum_{u,v} |R_{u,v}(\mathcal{G}_e) - R_{u,v}(\mathcal{G}_{\bar{e}})| \\ &= \sum_{u,v} R_{u,v}(\mathcal{G}_e) - \sum_{u,v} R_{u,v}(\mathcal{G}_{\bar{e}}) \end{aligned}$$

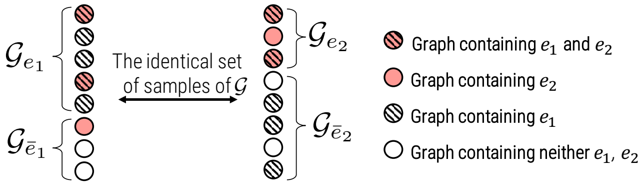

$\mathcal{G}_{e_1}$
$\mathcal{G}_{\bar{e}_1}$
The identical set of samples of $\mathcal{G}$
$\mathcal{G}_{e_2}$
$\mathcal{G}_{\bar{e}_2}$
Graph containing e_1 and e_2
Graph containing e_2
Graph containing e_1
Graph containing neither e_1, e_2

Figure 6: Reused sampling estimator for $\mathcal{E}RR$

Note that $\mathcal{E}RR^{e}$ equals to the difference of the expected number of connected pairs between two uncertain graphs $\mathcal{G}_e$ and $\mathcal{G}_{\bar{e}}$. In the context of edge relevance, reliability relevance can be seen as generalization of cut-edges, which quantifies the impact of partial edge deletion or addition on the connectivity in the uncertain graph. When the perturbation hits those edges with higher reliability relevance, it would produce bigger structural distortion over the overall *uncertain* graph.

On the basis of these edge-level reliablity relevance, we can now compute a vertex-level reliability relevance of a given vertex (Say u) as a weighted sum of reliability relevance of u's edges $\mathtt{E}^{u}$. Fix $u \in \mathcal{G}$ and let $\mathtt{E}^{u}$ be the pairs of vertices that include v, we have

$$\mathcal{V}RR^{u}(\mathcal{G}) = \sum_{e \in \mathtt{E}^{u}} p(e)\mathcal{E}RR^{e}(\mathcal{G})$$

The $\mathcal{V}RR^{u}(\mathcal{G})$ is a measure of the expected impact of vertex modification on the graph reliability. Namely, the higher the vertex's reliability relevance, the larger reliability distortion introduced by modification associated with its edges.

Reliability Relevance Evaluation. Given this theoretical foundation, the challenge is how to evaluate the reliability relevance of edges in a given uncertain graph efficiently ($\mathcal{E}RR-$eval). For each edge e, we need to measure the reliability difference over $\mathcal{G}_e$ and $\mathcal{G}_{\bar{e}}$. This evaluation involves the two-terminal reliability detection problem, which is known to be NP-complete [5].

A baseline algorithm for $\mathcal{E}RR-$eval is to use the Monte Carlo sampling. More precisely, we sample N possible worlds of the input uncertain graph, where N is large enough (around $1,000$) to guarantee high approximation accuracy. Over each sampled possible world (Say G), we carry out a connected-component computation algorithm to count the number of connected pairs $cc(G)$. Then, the count on the original uncertain graph $cc(\mathcal{G})$ can be estimated by taking the average over the sampled deterministic graphs.

Lemma 2. *The complexity of the baseline $\mathcal{E}RR-$eval algorithm is $\mathcal{O}(|E| \cdot N\alpha(|V|)|E|)$ where α is the inverse Ackermann function.*

PROOF. The time complexity of the connected component detection algorithm based on the union-find method is $\mathcal{O}(\alpha(|V|)|E|)$ [14]. Consequently, computing the $\mathcal{E}RR$ for an edge over the N possible worlds takes time $\mathcal{O}(N\alpha(|V|)|E|)$, and the total time complexity for all the edges is $\mathcal{O}(|E| \cdot N\alpha(|V|)|E|)$.

Algorithm 2 Edge Reliability Relevance Evaluation

```
    Input: $\mathcal{G} = (V, E, p)$, $N$ is the number of sampled graphs;
    Output: $\mathcal{E}RR$ Reliability relevance of edges in $\mathcal{G}$
1:  $CC_e \leftarrow \mathbf{0}$, $CC_{\bar{e}} \leftarrow \mathbf{0}$
2:  for i=1 to N do
3:      $G \leftarrow$ A deterministic sampled instance
4:      $Ind(G)$ is edge existence of sampled graph $\mathcal{G}$
5:      $cc(G) \leftarrow$ the number of connected pairs of $G$
6:      $CC_e + = Ind(G) \cdot cc(G)$,$CC_{\bar{e}} + = (\mathbf{1} - Ind(G)) \cdot cc(G)$
7:  end for
8:  $\mathcal{E}RR = CC_e/p - CC_{\bar{e}}/\mathbf{1} - p$
```

Algorithm 3 GenObf

```
    Input: Uncertain graph $\mathcal{G} = (V, E, p)$, $\mathcal{K}, k, \epsilon, c, q$, and standard deviation $\sigma$
    Output: A pair $\langle \tilde{\epsilon}, \tilde{\mathcal{G}} \rangle$ where $\tilde{\mathcal{G}}$ is a $(k, \epsilon)-$obfuscation, or $\tilde{\epsilon} = 1$ if fail to find a $(k, \epsilon)$-obfuscation.
1:  compute the uniqueness $U^v$ for $v \in V$
2:  compute the reliability relevance $\mathcal{V}RR^v$ for $v \in V$
3:  $Q^v \leftarrow U^v \cdot \mathcal{V}RR^v$ for $v \in V$
4:  $H \leftarrow$ the set of $\lceil \frac{\epsilon}{2}|V| \rceil$ with largest $Q^v$
5:  Normalized $\mathcal{V}RR^v$ for $v \in V \setminus H$
6:  $Q^v \leftarrow U^v \cdot 1 - \mathcal{V}RR^v$ for $v \in V \setminus H$
7:  $\tilde{\epsilon} \leftarrow 1$
8:  for $t$ times do
9:      repeat
10:         $E_C \leftarrow E$
11:         randomly pick a vertex $u \in V \setminus H$ according to $Q$
12:         randomly pick a vertex $v \in V \setminus H$ according to $Q$
13:         if $(u, v) \in E$
14:         then $E_C \leftarrow E_c \setminus \{(u, v)\}$ with the probability $p(e)$
15:         else $E_c \leftarrow E_c \cup \{(u, v)\}$
16:     until $E_C = c|E|$
17:     for all $e \in E_C$ do
18:         compute $\sigma(e)$
19:         draw $w$ uniformly at random from $[0, 1]$
20:         if $w < q$ then $r_e \leftarrow U(0, 1)$
21:         else $r_e \leftarrow R_{\sigma(e)}$
22:         $\hat{p}(e) \leftarrow p(e) + (1 - 2p(e)) \cdot r_e$
23:     end for
24:     $\hat{\epsilon} \leftarrow$ anonymityCheck($\hat{\mathcal{G}}$)
25:     if $\hat{\epsilon} < \epsilon$ and $\hat{\epsilon}$ then $\tilde{\epsilon} \leftarrow \hat{\epsilon}$; $\tilde{\mathcal{G}} \leftarrow \hat{\mathcal{G}}$
26: end for
27: return $\langle \tilde{\epsilon}, \tilde{\mathcal{G}} \rangle$
```

Obviously, the baseline algorithm is inefficient when the input uncertain graph is very large (it is quadratic in the number of edges). Here, we present a efficient algorithm for $\mathcal{E}RR$ evaluation in Algorithm 2. Its basic idea is to reuse the connected components detection result of samples as illustrated in 6. For each edge e, we group the sampled possible worlds according to the edge existence (Line 4-6), then get the sampled average of cc for each group as accurate approximation of $cc(\mathcal{G}_e)$ and $cc(\mathcal{G}_{\bar{e}})$. By this way, we bring the the evaluation of edge reliability relevance to the realm.

Lemma 3. *The time complexity of Algorithm 2 $\mathcal{E}RR-val$ is $\mathcal{O}(N\alpha(|V|)|E|)$ where N is the number of samples.*

E. The `GenObf` *Function*

Now, we are ready to present the details of the `GenObf()` function for finding a (k, ϵ)-obf instance for an input uncertain graph $\mathcal{G}$ in Algorithm 3. The function receives the parameters that are originally passed to Chameleon skeleton (Algorithm 1) including noise parameter σ.

Uniquness & Relevance Computation. The function begins the computation of the uniqueness score and reliability relevance. (Lines 1 & 2). These two invariants correspond to our goals of preserving privacy & utility. Based on these weighting factors, the `GenObf` then heuristically performs edge selection & perturbation, i.e, use the noise budget in the most effective way.

Exclusion. Since it is allowed not to obfuscate $\epsilon|V|$ of the nodes per the problem definition, the algorithm leverages the two invariants highlighted above and selects a set H of $\frac{\epsilon}{2}|V|$ nodes with the largest combined uniqueness and reliability relevance scores, and excludes them from subsequent obfuscation efforts.

Unifying Uniqueness and Relevance Score. Nodes not in H are candidates for anonymization. To anonymize high-uniqueness vertices, higher noise needs to be injected. Thus, edges associated with those vertices need to be sampled with a higher probability. Meanwhile, to better preserve the graph structure, edges associated with high reliability-relevance nodes need to be sampled with a smaller probability. In order to implement such sampling strategy, our algorithm assigns a probability Q^v to every $v \in V \setminus H$ (v in V but not in H), which is proportional to v's uniqueness U^v and inverse proportional to v's reliability relevance $\mathcal{V}RR^v$.

Edge Selection. After that, the algorithm starts its t trials for finding (k, ϵ)-*obfuscation*. Each trial first selects a set of candidate edges E_c, which will be subject to probability perturbation. Initially E_c is set to E. Then, the algorithm randomly selects two distinct vertices u and v, according to their assigned probabilities. The edge (u, v) is then excluded from E_c with the probability $p(e)$ if it is an edge in the original graph (Lines 14), otherwise it is added to E_c (Line 15). The process is repeated until E_c reaches the required size, which is controlled by the input parameter c. In typical uncertain graphs, the number of absent edges is usually significantly larger than the number of present uncertain edges. Thus, the loop usually ends very quickly for small values of c. And, the resulting set E_c includes most of edges in E.

Edge Perturbation. Next, we redistribute the noise budgets among all selected edges in proportion to their unify weighting factors. pecially, we define for each $e = (u, v) \in E_c$, its uncertainty level,

$$Q^e := \frac{Q^u + Q^v}{2}$$

and then set

$$\sigma(e) = \sigma|E_c| \cdot \frac{Q^e}{\sum_{e\in E_c} Q^e}$$

so that the average of $\sigma(e)$ over all $e \in E_C$ equals σ.

Edge Probability Perturbation. If we carefully perform edge prob alteration with the edge uncertainty levels, $\sigma(e)$, we effectively obfuscate node. In the following section, we will instantiate our ideas.

Success or Failure. Finally, If the algorithm successfully finds (k, ϵ)-obfuscated graph in one of its t trials, it returns the obfuscated graph with minimal ϵ. Otherwise, it indicates the failure by returning $\tilde{\epsilon} = 1$.

F. Anonymity-Oriented Edge Perturbing

In this section, we focus on the details of injecting noise and perturbation to the set of candidate edges E_c (Lines 19-26 in Algorithm 2). There are few techniques that inject uncertain noise to deterministic graphs (4^{th} cat. [7], [26], [27]). However, as ever discussed, these techniques assume the initial state of the edges is binary either exist or not, which is different from uncertain graphs.

Given an uncertain edge e with an initial probability $p(e)$ in the original graph, we first estimate a perturbation level $\sigma(e)$, which shapes the perturbation distribution allowed over e (Line 18 in Algorithm 2). A naive strategy to create the noise is to inject the perturbation in a random way (either addition or subtraction) as illustrated in Figure 7(a). However, we can theoretically prove that this *"un-guided"* injection is not optimal and with the same amount of injected noise a better anonymization can be achieved if the injection distribution is more controlled.

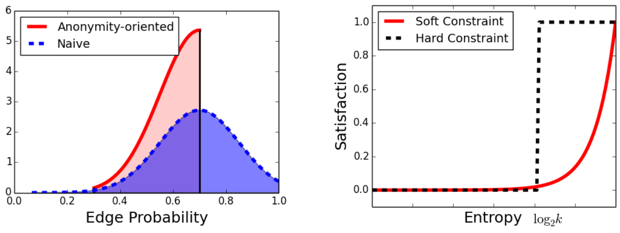
We will first introduce the proposed *"guided"* injection method, which we refer to as *anonymity-oriented perturbation*, and then in Section V-F1, we sketch why it works. Basically, Chameleon alters the probability of a given edge $e \in E_c$ according to the following equation:

$$\tilde{p}(e) := p(e) + (1 - 2p(e)) \cdot r_e$$

where r_e is a stochastic variable drawn from the truncated normal distribution.

Namely, for a given edge e with the probability $p(e)$, we only consider the potential edge probability $\tilde{p}$ in the limited range that is more likely to contribute to a higher graph anonymity by maximizing the entropy level. In Figure 7(a), we show an example where the initial $p(e) = 0.7$ and the assigned perturbation level $\sigma_e = 0.5$. In the naive strategy, $\hat{p}(e)$ will spread out in the wide range $[0, 1]$, whereas under the proposed *anonymity-oriented perturbation* strategy, $\hat{p}(e)$ is more focused in a specific range that should lead to a higher entropy.

Clearly, existing schemes in literature—which are defined over deterministic graphs—become a special case of the proposed scheme (by setting $p(e)$ to either 0 or 1).



(a) (b)

Figure 7: (a) Anonymity-oriented edge perturbing; (b) Relaxing k-obfuscation constraint.

1) Proof Sketching the Heuristic: We proceed to elaborate the rationality this anonymity-oriented edge perturbing scheme briefly. The formal detail proof of our heuristic is available in tech report. The core idea is to maximize the entropy of degree uncertainty matrix (referred to as ME).

To facilitate further discussion, we consider the extreme case k-obf, which poses a set of hard constraints over the anonymized solution. Let the constraint being k-obf be $\mathbb{C}$, $k-$obfuscate a vertex v be c_v. According to Definition 3, $k-$obf can be expressed as joint satisfaction of $\{c_v \ : \ v \in V\}$ since the uncertain graph is said to be k-obf iff it $k-$obfuscates all the vertices. The formal definition as follows.

$$\mathbb{C} = \prod_{v\in V} \mathsf{c}_v \tag{1}$$

where

$$\mathsf{c}_v := \begin{cases} 1 & H(Y_{P(v)}) \geq \log_2 k \\ 0 & otherwise \end{cases}$$

In other words, given an uncertain graph, its satisfaction evaluation of $\mathbb{C}$ indicates whether it achieves the desirable anonymity level (k-obf).

However, as shown in Figure 7(b), a single constraint at the vertex level is either fully satisfied or fully violated. It limits the optimization opportunity of methods based on local search. In this work, we model the individual constraint c_v to a fuzzy relation in which the satisfaction of a constraint is a continuous function of its variables' values (*i.e.*, the entropy $H(Y_{P(v)})$, going from fully satisfied to fully violated as follows.

$$C_v = e^{H(Y_{P(v)}) - \log_2 |V|} \tag{2}$$

Lemma 4. *Let Ω presents the domain of degree values in the original uncertain graph, the maximization of the provided anonymity $\mathbb{C}$ is equivalent to the maximization of the following function:*

$$\sum_{\omega\in\Omega} s(\omega) \cdot H(Y_\omega) \tag{3}$$

Proof Sketch: First we can see that

$$\mathcal{C} = \prod_{v\in V} \mathsf{c}_v = \prod_{\omega\in\Omega} \underbrace{\mathsf{c}_\omega \dots \mathsf{c}_\omega}_{s(\omega)}$$

Taking logarithm for both sides and combining with the approximation equation 2, we can see that

$$
\begin{aligned}
\log(\mathcal{C}) &= \sum_{\omega\in\Omega} s(\omega)\log(\mathsf{c}_\omega) \\
&= \sum_{\omega\in\Omega} s(\omega)\big[H(Y_\omega) - \log_2|V|\big] \\
&= \sum_{\omega\in\Omega} s(\omega)H(Y_\omega) - \sum_{\omega\in\Omega} \log_2|V|
\end{aligned}
$$

Therefore, after removing the constant $\sum_\omega \log_2|V|$ from $\log(\mathcal{C})$, our goal is actually to maximize Equation 3. It provides us with the relation between the global anonymity and the level of disorder of the degree uncertainty matrix.

Lemma 5. *The maximization of Equation 3 is equivalent to maximization of the following function:*

$$\sum_{\omega\in\Omega} s(\omega)\cdot H(Y_\omega) = \big[\sum_{v\in V} H(d_v)\big] + |V|\log|V| - |V|H(\Omega) \tag{4}$$

The equation stems from the coding length of degree uncertainty matrix from different perspectives (row and column).[1]

It provides us with the mechanism for gaining better anonymity, namely increasing the degree uncertainty per vertex $H(d_v)$.

Lemma 6. *As implied by the Central Limit Theorem, d_v may be approximated by the normal distribution $\mathcal{N}(\mu,\sigma^2)$, where $\mu = \sum_{e\in\mathcal{E}^v} p(e)$ and $\sigma^2 = \sum_{e\in\mathcal{E}^v} p(e) - p(e)^2$. Therefore, its entropy may be approximated by the differential entropy of the normal distribution $\frac{1}{2}\ln(2\pi\sigma^2) + \frac{1}{2}$. For a given $p(e)$, its gradient ascent is proportion to $1 - 2\cdot p(e)$.*

Targeting at high entropy, we apply the gradient ascent method—$\hat{p}(e) = p(e) + \big(1 - 2\cdot p(e)\big)\cdot r_e$ for achieving the increase of degree entropy and the anonymity gain.

VI. Experiments

In this section, we evaluate how well Chameleon and its variants preserve a graph's structural statistics by comparing its anonymized *uncertain* graphs against the original one in terms of graph metrics. The evaluated methods are Chameleon (RSME), RS(Reliability Sensitive), ME(Maximization Entropy) and Rep-An (Representative Anonymization). Strong structural similarity (small error) in these results would establish the utility of these anonymized graphs in real research analysis and experiments.

A. Evaluation Metrics

Metrics. Besides reliability, our evaluation includes three classes of graph metrics. One group includes degree-based metrics such as Average Node Degree, Degree Distribution, Maximal Degree. These are basic topological metrics capture how degree distributed among nodes and how nodes with particular degree connect among others. The second group includes node separation metrics such as Average Distance, Graph Diameter. They are used to quantifying the interconnectivity of the graph. The third group metrics include Clustering Coefficient which measures how close neighbors of a node are to forming cliques.

Table II: Summary of compared methods.

Method	Uncertainty -aware	Reliability -oriented	Anonymity -oriented	Source
Rep-An	–	–	✓	[29]+ [7]
RSME	✓	✓	✓	This work
ME	✓	–	✓	This work
RS	✓	✓	–	This work

Computation. Since there does not exist the closed formula for graph metrics expect Average Node Degree, the results are approximated by Monte Carlo sampling. Specifically, we create a number of random instances of an uncertain graph, and we compute the expected value of each metric using the average of the sampled graphs. Here, we use 1,000 samples since it has been shown that 1000 usually suffices to achieve accuracy converge [19], [30]. In particular, we use Approximate Neighborhood Function (ANF) [8], to approximate shortest path-based statistics. For each metric, we report the ratio of absolute difference against the original one.

Parameter Setting. We generate anonymized *uncertain* graphs for $k \in [100, 300]$ and compare the graph metrics of the resulting *uncertain* graphs against those original graphs. We limit ourselves to obfuscation levels, $k \in [100, 300]$ following reasons. First, we aim to explore the case the desired privacy level requires a small amount of noise ($k = 100$). This way, we can quantify the utility loss difference introduced by Chameleon against Rep-An method. Second, it naturally requires a high level of noise to provide strong levels of privacy guarantees. We want to explore the sensitivity of different variants. Unfortunately, very large values of k require large noise, thus producing anonymized graphs that are extremely different from the original ($k = 300$).

B. Results

Degree-based Metrics. For brevity, we report results for Average Node Degree. Figure 9 compares the average node degrees. For each of the DBLP, BRIGHTKITE and PPI graphs, the average node degrees of Chameleon ($k = 100$) output graphs are very close the ones of the original graphs. When we increase the strength of the privacy guarantees, i.e., larger k values of 200 and 300, the error of average degree progressively increases. For example, DBLP graph shows a small deviation even for $k = 300$. The worst-case average degree deviation is still within 15% of the original.

On the other hand, BRIGHTKITE and PPI show slightly different behaviors. For large k values, i.e., $k = 200$ and $k = 300$, the largest error is over 300% from the original values. We attribute them to the existence of heavy-tailed degree distribution in these two graphs which requires more perturbation.

Node Separation Metrics. For brevity, we report only the Average Distance as a representative of the node separation metrics. Figure 10 shows the Average Distance (AD) values computed on DBLP, BRIGHTKITE, and PPI graphs compared

[1] More detail of it is available in tech report.

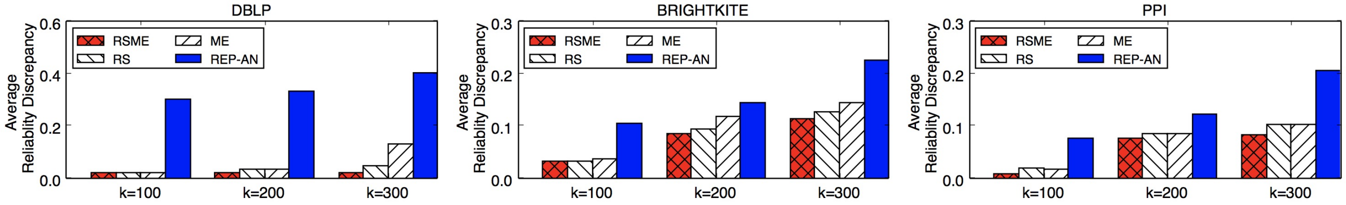


Figure 8: Comparison of anonymization methods in terms of their ability to preserve Reliability.

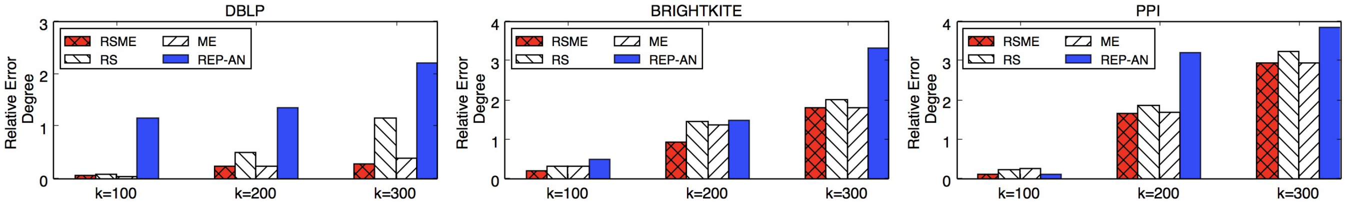


Figure 9: Comparison of anonymization methods in terms of their ability to preserve Average Node Degree.

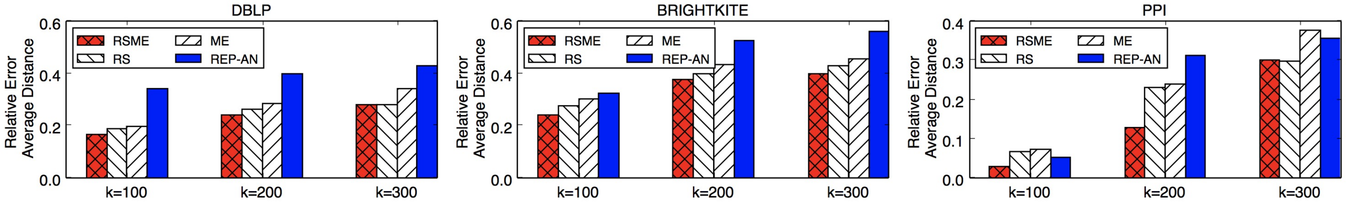


Figure 10: Comparison of anonymization methods in terms of their ability to preserve Average Distance.

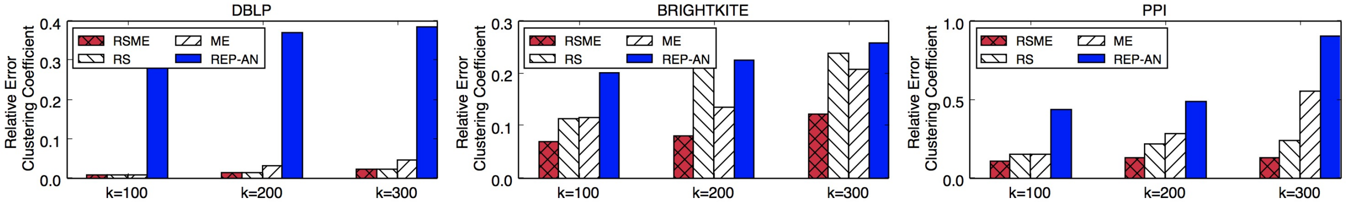


Figure 11: Comparison of anonymization methods in terms of their ability to preserve Clustering Coefficient.

to the AD values on their anonymized graphs. In this case, all of Chameleon output graphs do a good job of preserving the average distance of the original graphs.

Clustering Coefficient

Summary. Our experimental evaluation on real-world datasets confirms the initial and driving intuition: the Chameleon approach which explicitly incorporates edge uncertainty and the possible world semantic in the anonymization process outperforms the benchmark solution Rep-An significantly regarding the uncertain graph utility preservation. The Chameleon introduces limited impact as a result of adding noise to guarantee privacy.

Another take-home message is: by using fine-grained and uncertainty-aware perturbation strategies such as reliability sensitive edge selection (RS) and max entropy based edge prob alteration (ME), one can achieve the same desired level of obfuscation with the smaller change on the uncertain graph thus maintaining higher data utility.

VII. Conclusion

In this work, we study the problem of developing a flexible *uncertain* graph privacy scheme that preserves graph structure while providing the user-specified level of privacy guarantee. First, we introduce the Rep-An scheme that aims these goals using the representative instance as an approximation of *uncertain* graph and show it requires the addition of high levels of noise to obtain privacy guarantee. Second, we develop Chameleon approaches that seamlessly integrate edge uncertainty into the core of the anonymization process such as the evaluation of privacy risk, utility loss and judicious uncertain graph modifications. We present a new utility-loss metric based on the solid connectivity-based graph model under the possible world semantics, namely the reliability discrepancy. Moreover, we introduce the reliability-sensitive edge selection strategy (RS) and max-entropy edge perturbation (ME) strategy for better utility preserving. Experimental studies on different real-world datasets demonstrate that our approach can anonymize uncertain graphs to the desired anonymity level with at a slight cost of utility, better than Rep-An.

References

[1] A. S. Abrahams, E. Coupey, E. X. Zhong, R. Barkhi, and P. S. Manasantivongs. Audience targeting by b-to-b advertisement classification: A neural network approach. *Expert Systems with Applications*, 40(8):2777 – 2791, 2013.

[2] E. Adar and C. Re. Managing uncertainty in social networks. *IEEE Data Eng. Bull.*, 2007.

[3] T. Alsina, D. Wilson, S. Joshi, and S. Sundaresan. Targeting customer segments, Dec. 3 2015. US Patent App. 14/289,118.

[4] S. Asthana, O. King, F. Gibbons, and F. Roth. Predicting protein complex membership using probabilistic network reliability. *Genome research*, 2004.

[5] M. O. Ball. Computational complexity of network reliability analysis: An overview. *IEEE Transactions on Reliability*, 1986.

[6] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava. Class-based graph anonymization for social network data. *Proc Vldb Endow*, 2009.

[7] P. Boldi, F. Bonchi, A. Gionis, and T. Tassa. Injecting uncertainty in graphs for identity obfuscation. *VLDB*, 2012.

[8] P. Boldi, M. Rosa, and S. Vigna. HyperANF: approximating the neighbourhood function of very large graphs on a budget. *CoRR*, 2011.

[9] K. Bollacker, C. Evans, P. Paritosh, and T. Sturge. Freebase: a collaboratively created graph database for structuring human knowledge. *SIMMOD*, 2008.

[10] F. Bonchi, A. Gionis, and T. Tassa. Identity obfuscation in graphs through the information theoretic lens. *ICDE*, 2014.

[11] J. Casas-Roma. Privacy-preserving on graphs using randomization and edge-relevance. *Modeling Decisions for Artificial Intelligence*, 2015.

[12] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. *kdd*, 2011.

[13] Colbourn and Colbourn. The combinatorics of network reliability. 1987.

[14] M. Fredman and M. Saks. The cell probe complexity of dynamic data structures. *STOC*, 1989.

[15] J. Ghosh, H. Ngo, and S. Yoon. On a routing problem within probabilistic graphs and its application to intermittently connected networks. 2007.

[16] S. Hartung and N. Talmon. The complexity of degree anonymization by graph contractions. *TAMC*, 2015.

[17] M. Hay, G. Miklau, D. Jensen, D. Towsley, and C. Li. Resisting structural re-identification in anonymized social networks. *The VLDB Journal*, 2010.

[18] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing social networks. 2007.

[19] R. Jin, L. Liu, B. Ding, and H. Wang. Distance-constraint reachability computation in uncertain graphs. *VLDB*, 2011.

[20] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. 2003.

[21] N. Krogan, G. Cagney, H. Yu, G. Zhong, and X. Guo. Global landscape of protein complexes in the yeast saccharomyces cerevisiae. *Nature*, 2006.

[22] J. Lee and C. Clifton. *How Much Is Enough? Choosing ϵ for Differential Privacy*. 2011.

[23] M. Lin, M. Lin, and R. J. Kauffman. From clickstreams to searchstreams: Search network graph evidence from a b2b e-market. *ICEC*, 2012.

[24] K. Liu and E. Terzi. Towards identity anonymization on graphs. *SIGMOD*, 2008.

[25] L. Liu, J. Wang, J. Liu, and J. Zhang. Privacy preservation in social networks with sensitive edge weights. *SDM*, pages 954–965, 2009.

[26] P. Mittal, C. Papamanthou, and D. Song. Preserving link privacy in social network based systems. *NDSS*, 2013.

[27] H. Nguyen, A. Imine, and M. Rusinowitch. Anonymizing social graphs via uncertainty semantics. *CCS*, 2015.

[28] M. Ninggal and J. H. Abawajy. Utility-aware social network graph anonymization. *J Netw Comput Appl*, 2015.

[29] Parchas, Gullo, Papadias, and Bonchi. The pursuit of a good possible world: extracting representative instances of uncertain graphs. *SIGMOD*, 2014.

[30] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios. K-nearest neighbors in uncertain graphs. *VLDB*, 2010.

[31] M. Skarkala, M. Maragoudakis, S. Gritzalis, L. Mitrou, H. Toivonen, and P. Moen. Privacy preservation by k-Anonymization of weighted social networks. *ASONAM*, 2012.

[32] L. Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 2002.

[33] B. Thompson and D. Yao. The union-split algorithm and cluster-based anonymization of social networks. 2009.

[34] Y. Wang, L. Xie, B. Zheng, and K. C. K. Lee. Utility-oriented k-anonymization on social networks. *DASFAA*, 2011.

[35] W. Wu, Y. Xiao, W. Wang, Z. He, and Z. Wang. k-symmetry model for identity anonymization in social networks. *EDBT*, 2010.

[36] X. Ying and X. Wu. Randomizing social networks: a spectrum preserving approach. pages 739–750, 2008.

[37] L. Zhang, S. Chen, Y. Jian, Y. Fang, and Z. Mo. Maximizing lifetime vector in wireless sensor networks. *IEEE/ACM Trans. Netw.*, 2013.

[38] B. Zhao, J. Wang, M. Li, F. Wu, and Y. Pan. Detecting protein complexes based on uncertain graph model. *IEEE/ACM Transactions on Computational Biology and Bioinformatics.*, 2014.

[39] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. *ICDE*, 2008.