Sharing Uncertain Graphs with Syntactic Anonymity

Dongqing Xiao, Mohamed Y. Eltabakh, Xiangnan Kong

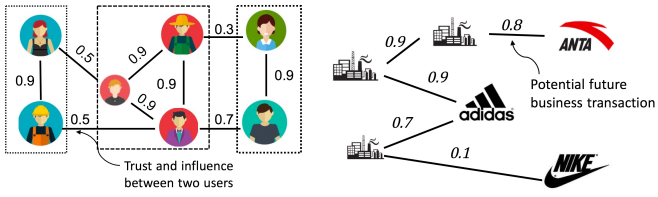
Computer Science Department, Worcester Polytechnic Institute
Worcester, United States of America
{dxiao, meltabakh, xkong}@wpi.edu

***Abstract*—Many graphs in real-world applications, such as social network and business to business network, are not deterministic but are uncertain. Related research requires open access to such uncertain graph datasets. While sharing these datasets often risks exposing sensitive data to the public. Current works mainly concern about privacy issues with deterministic graph sharing. The uncertain scenario is overlooked.**

We first show conventional methods are not applicable for uncertain graph sharing. By disregarding the possible world semantic, they significantly disrupt the stochastic structure. Our work seeks a solution to share meaningful probabilistic graphs without compromising user privacy. We develop a syntactically private algorithm, Squid, for solving this problem. It integrates the possible world semantic into the core of anonymization. It enables a fine-grained, uncertainty-aware control over the injected noise. We apply our method to real uncertain graphs and show its efficiency and practical utility.

I. Introduction

Graphs are widely used to capture the complex relationships in emerging applications, such as business to business (B2B) and social networks. Sometimes, the existence of the relationship between two entities is uncertain. For instance, in social networks, nodes represents individual users, while edges represent friendship or trust link among them. Usually, the link is derived by inference and prediction models built on interaction details [1, 13, 15]. And, edge probability denotes the accuracy of a link prediction, or the trust of one person on another. In these applications, the data can be modeled and shared as uncertain graphs whose edges carries a probability of existence. The probability represents the confidence that the relationship holds in reality.



(a) Social Trust Network (b) B2B Network

Figure 1: Real-world uncertain graphs with privacy concerns.

These uncertain graphs are invaluable for scientific research and commercial applications [5, 13]. However, sharing these uncertain graphs would violate the privacy of users or entities profiled inside. In social trust network, the trust relationships among users— which greatly impact users' behaviors, are usually probabilistic. They are useful in social interaction study and micro-targeting. While users are unwilling to share such confidential information with potential adversaries like Cambridge Analytica. In B2B networks, business operators also hesitate to share transaction patterns as it relates to confidential business models. Such tension is raising the question of sharing uncertain graphs without compromising privacy.

A number of privacy preserving graph sharing schemes have been studied in the deterministic scenario [14, 16–18, 22, 25, 27, 28], though many problems still remain unexplored in the uncertain scenario.

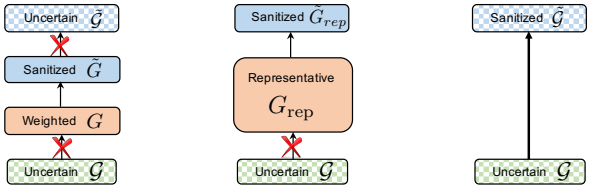


Figure 2: XXXX.

An obvious approach is to convert uncertain graph sharing problem into the deterministic case by casting edge probabilities as edge weights. Note that, one of the most goals of sharing uncertain graphs is to maintain the data utility. However, by disregarding the possible world semantics of the uncertain graph, casting-based approach fails to reflect uncertain graph properties such as connectivity, dense subgraphs correctly [12, 30]. Hence, casting-based scheme could produce very poor result in the uncertain scenario even if the weighted graph anonymization algorithm is good.

Note that connectivity of deterministic subgraphs is generally measured by the concept of cut, which is defined as the sum of weights of intra edges. Generally, the bigger the cut, the harder to separate two subgraphs. In Figure 3(a), the equal cut $C(SG_1, SG_2) = C(SG_3, SG_2) = 1$ implies the identical connectivity of SG_1 and SG_3 w.r.t SG_2. However, with the possible world semantics, we know the probability to separate SG_1 and SG_2 is $(1-0.5)^2 = 0.25$, and that to separate SG_2 and SG_3 is $(1-0.3)(1-0.7) = 0.21$. Hence, in fact, SG_2 is closer to SG_1 than to SG_3.

Another approach we proposed in [8], called rep-based anonymization, is based on the idea of processing uncertain graph through representative instances [20]. It first extracts a single deterministic representative instance G that capture structural properties of the uncertain graph. After that, anonymization can be then be proceed efficiently on G using conventional algorithms, regardless of the uncertainty.

However, Rep-An is is not always feasible. The detachment of edge uncertainty deteriorates the data utility.

Conventional graph anonymization schemes are inadequate to share uncertain graphs with a desirable trade-off between privacy and utility. It is worthwhile to consider developing the specially optimized solution for handling following challenges.

• *Stochastic Privacy Attacks.* Edge uncertainty plays an indispensable role in the uncertain graph model. It is impractical to discard them in the release. While the extra release of edge uncertainty makes privacy protection far more difficult as it empowers the adversary and makes the profiled entity more vulnerable.

• *Stochastic Utility Loss Metric.* It is challenging to maintain the structure when the uncertain graph is modified to pursue anonymity. The structural distortion incurred is evaluated by the specially designed utility loss metric. It plays the key role in utility preserving. Unfortunately, existing graph utility loss metrics such as graph edit distance [16], spectrum discrepancy [28], community reconstruction error [25] and shortest path discrepancy [17] are not suitable in the uncertain scenario because of the ignorance of edge uncertainty.

• *Intractable Search Space.* The goal is to find a sanitized graph with the desired level of privacy by as few graph mutations as possible. Even the simple deterministic graph anonymization problem, *i.e.*, only with edge additions and deletions, is a NP-hard problem [9]. In the uncertain scenario, the edge modification is no longer a binary operation (addition/deletion), but can be infinite probability values. Exhaustive search is computationally intractable if the number of edges is large. This makes the problem of uncertain graph anonymization very challenging.

In this work, we propose a solution tailored towards uncertain graphs via incorporating possible world semantics. It preserves as much the stochastic nature of the original uncertain graph as possible, while injecting enough structural noise to guarantee a chosen level of privacy. Specifically, we make the following contributions.

- We are the first to formulate the uncertain graph anonymization problem. We show the potential re-identification attack and present the corresponding privacy notion.
- We propose an utility loss metric on the basis of reliability. It evaluates the connectivity difference in the context of the entire graph and also utilizes the possible world model.
- We propose a randomized algorithm with the hybrid of uncertainty-aware heuristics. It excels in identifying a population of sanitized results with good quality efficiently.
- We conduct extensive experimental studies to demonstrate efficiency and practical utility of our algorithms.

The rest of the paper is organized as follows. In Section 2, we summarize related works and clarify our distinct privacy goal. In Section 3 we formulate the uncertain graph-anonymization problem. Sections 4 – 5 consider the anonymization problem in the context of uncertain graphs. In Section 6 we apply our method to several real-world uncertain graphs, and demonstrate their efficiency and practical utility.

II. Related Work

A significant amount of prior work has been done on protecting the privacy of network datasets. The comprehensive survey is out of the scope of this paper. Here, we briefly summarize related work and clarify our privacy goal.

Syntactic Privacy. Early works on privacy-preserving network publishing focus on developing anonymization techniques. Many of them modify the graph structure in subtle ways that guarantee privacy but keep much of graph structure for release. The released graph is available for all the analysis tasks. These approaches usually provide privacy protection against specific de-anonymization attacks. Most of them leverage syntactic privacy models derived from k-anonymity [24] which requires creating k identical entities (*e.g.* neighborhoods, degree nodes) to blend victims.

Corresponding graph anonymization methods can be classified into four main categories: (1) Clustering-based generalization [2, 10, 11]; (2) *Edge modification* [16, 23, 25, 26, 31], (3) *Edge randomization* [17, 19, 28], and (4) *Uncertainty semantic-based modifications* which add uncertainty to some edges and thus converting the deterministic graph to an uncertain version [4, 18]. The uncertainty semantic-based approaches are known as the state-of-art ones because of their excellent privacy-utility trade-off, brought by the fine-grained perturbation leveraging the uncertain semantics.

Differential Privacy. Another option is to apply ϵ-differential privacy policy to providing privacy guarantee. It roughly falls into two directions. The first direction aims to release specific differentially private mining results, such as degree distributions, sub-graph counts, and frequent graph patterns [7, 27]. These methods only publish query result, early uses of the data can affect the quality of later uses. What's worse, no new queries can be permitted on the data. The second direction aims to share the meaningful graph [22]. Most research in this direction projects an input graph to dK-series and ensures differential privacy on dK-series statistics. Later, private statistics are then either fed into generators or MCMC process to generate a fit synthetic graphs. While current techniques are still inadequate to provide desirable data utility for many graph mining tasks.

All the methods target at providing privacy guarantee to the deterministic graph. The uncertain scenario is unexplored.

A. Our Privacy Goal

Our work seeks a solution to share meaningful uncertain graphs while preserving privacy. As ever discussed, existing schemes fail to provide utility guarantee in the uncertain scenario. We believe the anonymization process needed to be specially optimized. In this paper, we try to move this line of research one step forward from the deterministic context to a broader probabilistic context.

There is a widespread belief that differential privacy and its offsprings are immune to various privacy attacks. It offers a guarantee bound ϵ on the loss of privacy due to the data release [22, 27]. However, there is no clear way to set general policy for choosing the privacy parameter ϵ for sufficient

privacy guarantee [14]. Its implications and impacts on the risk of disclosure in practice heavy depend on data detail. Thus, differential privacy is difficult to apply in practice.

In contrast, the notion of syntactic privacy can be defined and understood based on the data schema. And, its parameters have a clear privacy meaning that can be understood independent of the actual data. Moreover, they have a clear relationship to the privacy regulation of individual identifiability (e.g., European GDPR Law). Hence, we focus on sharing uncertain graphs with syntactic anonymity.

III. Problem Formulation

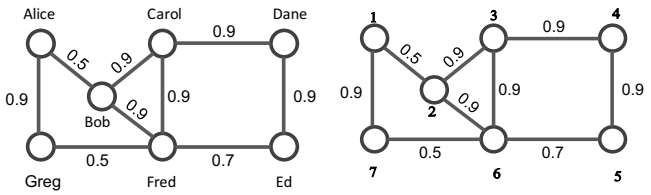
In this section, we provide background on uncertain graph, privacy attack and justify our choice of utility loss metric. On this basis, we present our formulation of the uncertain graph anonymization problem.

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 [6, 21, 30], we assume possible-worlds semantics, and we consider the edge probabilities independent [1]. An uncertain graph $\mathcal{G} = (V, E, p)$ essentially represents a probability distribution over all of the certain graphs G in the forms of which the uncertain graph may actually exist. The probability of observing any possible world $G_i = (V, E_{G_i})$ is

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

B. Privacy Attack



(a) Social Trust Network (b) The naive anonymization

Figure 3: The Structural Re-Identification Issue.

Apparently, simply removing the identities of the nodes before publishing the uncertain graph does not guarantee privacy. The structure of the uncertain graph itself, and in its basic form the degree of the nodes, can be revealing the identities of individuals. In practice, the adversary may have access to external information about the entities in the graphs. This information may be obtained by the adversary's malicious actions. For example, for the uncertain graph in Figure 3, the adversary might know that "Fred has three or more **trust** neighbors". Such information allows the adversary to narrow down the set of candidates in the sanitized graphs. For example, the statement partially re-identify Fred as $\{2, 3, 6\}$ with **probabilities** respectively. Different to the deterministic scenario, there are different posterior probabilities over candidate nodes $\{2, 3, 6\}$ where $P(6|\text{Fred}) \simeq P(3|\text{Fred}) \gg P(2|\text{Fred})$.

[1]We leave the conditional probability model as a future extension.

The node is vulnerable to the re-identification risk. Entity Re-identification (ER) can lead to additional disclosures. In this paper, we focus on the ER attack as this attack is one of the most serious privacy problems.

C. Privacy Notion

To resist re-identification attacks, we adopt the (k, ϵ)-obf, an syntactic privacy notion, where $k \geq 1$ is a desired level of obfuscation and $\epsilon \geq 0$ is a tolerance parameter.

Obfuscation Parameter Similar to $k-$anonymity, $k-$obf requires blending every entity with other fuzzy matching entities. While, the level of obfuscation is quantified as the entropy over posterior probabilities over fuzzy matching ones. It lower bounds the entropy of the distribution by $\log_2 k$. *Though it is initially used to measure the anonymity provided by an uncertain graph to the deterministic graph, the stochastic nature makes it a good fit in the uncertain scenario.*

Tolerance parameter As for the tolerance parameter ϵ, it serves for the following purpose. There might be extreme unique nodes, e.g., Trump in a Twitter network, whose obfuscation is almost impossible. Thus, Boldi *et al.* [4] introduce a tolerance parameter ϵ, which allows skipping up to $\epsilon * |V|$ nodes and makes the privacy goal more practical.

D. Utility Loss Metric: Reliability Discrepancy

As a fundamental property, connectivity plays a vital role in graph mining tasks such as nearest neighbor locating and clustering. The connectivity model can yield a better graph representation than the degree sequence model. Motivated by the above, connectivity discrepancy is widely used to measure the structural difference between deterministic graphs.

The concept of reliability generalizes the connectivity concept uncertain scenario. It captures the probability that two given nodes are reachable over all possible worlds, as shown in Def 1. Analogous to the deterministic case, we use reliability discrepancy as the utility-loss metric in the uncertain scenario, as outlined in Def 2.

Definition 1. ***Two-Terminal Reliability [6]*** *Given an uncertain graph $\mathcal{G}$, and two distinct nodes u and v in the graph, 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. ***Reliability Discrepancy (RD)*** *The reliability difference between a sanitized output $\tilde{\mathcal{G}}$ and the original input $\mathcal{G}$, denoted as $\Delta(\tilde{\mathcal{G}})$, is defined as the sum of the two-terminal reliability discrepancy over all node pairs $(u, v) \in V_{\mathcal{G}}$.*

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

E. Problem Statement

Problem 1. *Given an uncertain graph $\mathcal{G}$ and desired anonymization parameters k and ϵ, the objective is to find a (k,ϵ)-obf uncertain graph $\tilde{\mathcal{G}}$ with the minimal utility loss,*

$$\underset{\tilde{\mathcal{G}}}{\text{argmin}} \quad \Delta(\tilde{\mathcal{G}})$$
$$\textit{Subject to} \quad \tilde{\mathcal{G}} \textit{ is } (k,\epsilon) - obf$$

IV. The State-of-Art Approach

Before presenting our solution Squid for uncertain graph sharing, we first describe the state-of-art approach (Obf) [4]. The main purpose of describing it is to separate the basic framework with the key idea of Squid. They differ in how they represent the graph property and how they perform mutation.

A. Overview

The Obf method obfuscates the (deterministic) graph data by adding or removing edges *partially*. For each edge e, it assigns a probability deviation r_e, where $r_e \leftarrow R(\sigma)$. In particular, the uncertainty injecting scheme proceeds as follows:

$$p(e) = \begin{cases} 1 - r_e & e \in E \\ r_e & otherwise \end{cases} \tag{1}$$

The most widely known member of generating distribution R_σ is the truncated normal distribution with mean 0 and variance σ^2. While, R could in principle be any distribution. As the standard deviation σ decreases, a greater mass of R_σ will concentrate near $r_e = 0$. Then, the amount of injected noise and consequent structural distortion will be smaller. Aiming at high utility, Obf aims at injecting the minimal amount of uncertainty need to achieve the required obfuscation. Thus, it calibrates the minimal amount of uncertainty via a binary search on the value of standard deviation σ, as outlined in Algorithm 1.

Algorithm 1 The obfuscation algorithm

Input: Graph $\mathcal{G}$, obfuscation level k, tolerance parameter ϵ
Output: The result $\mathcal{G}_{obf}$
1: $\sigma_l \leftarrow 0$; $\sigma_u \leftarrow 1$
2: **repeat**
3: $\langle \hat{\epsilon}, \hat{\mathcal{G}} \rangle \leftarrow$ **genObf**(-,σ_u)
4: **if** $\hat{\epsilon} = 1$ (fail) **then** $\sigma_l \leftarrow \sigma_u$; $\sigma_u \leftarrow 2\sigma_u$
5: **until** $\hat{\epsilon} \neq 1$
6: **repeat**
7: $\sigma_{mid} \leftarrow (\sigma_u + \sigma_l)/2$
8: $\langle \hat{\epsilon}, \hat{\mathcal{G}} \rangle \leftarrow$ **genObf**(-,σ_{mid})
9: **if** $\hat{\epsilon} = 1$ **then** $\sigma_l \leftarrow \sigma_{mid}$
10: **else** $\sigma_u \leftarrow \sigma_{mid}$; $\mathcal{G}_{obf} \leftarrow \hat{\mathcal{G}}$
11: **until** $\sigma_u - \sigma_l$ is enough small
12: return $\mathcal{G}_{obf}$

The search flow is determined by the function **genObf**. The function **genObf** handles the search of (k,ϵ)-*obf* instances using a given standard deviation parameter σ. It either returns the found (k,ϵ)-*obf* instance or failure signals. The search starts with an initial guess of an upper bound σ_u, which is iterative doubled until a (k,ϵ)-*obf* instance is found. Then, the binary search is performed over the range $[0,\sigma_u]$. The binary search terminates until the search interval is sufficiently short. The algorithm outputs the best (k,ϵ)-*obf* found (the last one that was successfully generated).

B. The function **genObf**

In reality, finding (k,ϵ)-*obf* instances using a given parameter σ is not that easy. The function **genObf** utilizes the randomized search to handle intractable search space. Multiple attempts are performed (In our experiment, 5 attempts are performed.). Iff all the attempts fail, **genObf** returns failure signals, otherwise, returns the found (k,ϵ)-*obf* instance.

Each attempt begins by selecting a subset of edges subject to alteration. Then, it assigns the deviation among selected edges and injects uncertainty. In particular, they suggest calibrating the perturbation applied to an edge e according to the "uniqueness" of the two nodes u and v. In brief, if both u and v are common nodes w.r.t the property, then r_e should be very small; on the other hand, if u and v are outliers, then r_e should be higher. Meanwhile, edges need to be sampled with the higher probability if they are adjacent to outliers.

C. Limitations

The above-mentioned approach achieves the desired level of obfuscation with the small change in the data, thus maintaining high utility However, the design heavily tailored towards the deterministic context. In the uncertain scenario, it suffers from following issues. First, its scheme does not consider the structural relevance of edges in critical edge selection/alteration steps, which leads to unnecessary structural distortion. Second, its scheme assumes the existence of edges is known with certainty, thus fails to handle uncertain graphs where the existence of edges is probabilistic. All the operators, such as selection and alteration, need to be integrated with possible world semantics carefully. We are left asking the question, *how to generalize existing methods to the probabilistic context?*

V. Privacy Via Squid

In this section we describe our algorithm, Squid, which injects uncertainty to the given uncertain graph so that it becomes (k,ϵ)-*obf* while preserving as much the stochastic nature as possible. A key feature of our method is to seamlessly integrate edge uncertainty and possible world semantics into the core of anonymization operators.

A. Heuristic for Edge Perturbation

The key step of graph anonymization is to select a set of edges subject to modification which balances privacy gain and structural distortion. It involves consideration over the exponential number of edge combinations. Recently, the most popular paradigm for solving such problems has been using a class of heuristics. Successes of this approach include (1) anonymity-aware ones that suggest injecting more considerable noise to unique nodes [4, 10, 29] (2) utility-aware ones that indicate avoiding distortion over "bridge" edges whose deletion/addition would significantly impact the graph structure [19, 25]. The judicious edge selection must involve two types of heuristics which complement each other. Individually, they are far less effective. While, they have not been explored yet in the context of uncertain graphs.

Motived by the above, we first generalize the calibration of uniqueness via the use of KL-divergence. Second, we propose a generalized version of edge relevance from an information-theoretic perspective. Besides, we develop an efficient algorithm for its computation. And, we show the use of such a criterion boosts XXX efficiently and straightforwardly.

Generalized Uniqueness The uniqueness criterion was used to measure how unique a given node is among all the nodes in the graph w.r.t a specific property. For a given node, its uniqueness score is the inverse of the commonness score of its property value w. While, the commonness score of w amounts to the weighted average distance among all other property values.

However, the conventional method merely formulates node properties as discrete values and relies on the geometric distance function to measure their distance. Thus, it fails to handle our problem where the property values are probabilistic. In this work, we extend the preliminary version to handle the probabilistic case.

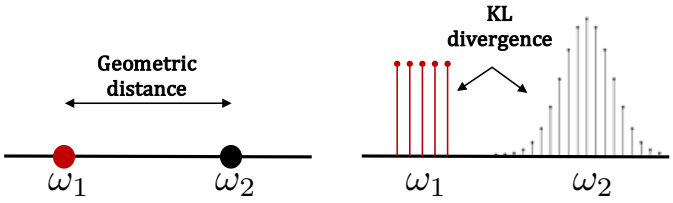


Figure 4: The generalization of uniqueness.

We systematically model uncertain property values in both continuous and discrete domains as continuous and discrete random variables, respectively. We consider the use of probability distributions, which are essential characteristics of uncertain property values, in the measuring similarity between uncertain property values. We use the well known Kullback-Leibler divergence to measure the distance between random variables with parameterized distributions. The generalized uniqueness can then be formalized as

Definition 3. *Uniqueness Score Let $P : V \rightarrow \Omega_P$ be a property on the set of nodes V of the uncertain graph, let d be a KL divergence function, and let $\theta > 0$ be a parameter. Then the $\theta-$commonness of the property values ω is $C_\theta(\omega) := \sum_{u \in V} \Phi_{0,\theta}(KL(\omega, P(v)))$, while the corresponding uniqueness is $U_\theta := \frac{1}{C_\theta(\omega)}$.*

Note that, the weights decays exponentially as a function of the KL divergence, and the parameter θ determines the decay rate. We set $\theta = \sigma$ as the injected noise blurs the meta distribution of property values.

Generalized Edge Relevance It is clear that alteration over a single edge would produce local structural change and send ripples through the rest of the graph. The incurred structural distortion varies on the topological role of the edge subjecting to alteration, even with the same amount of alteration. Targets at the high utility, we should penalize modification over structurally critical edges. It raised the need of measuring the relevance of edges in the uncertain graph.

There are many potential ways to measure it. Importantly, the metric must be fitted to the context of uncertain graphs. Inspired by the importance of reliability, we measure the edge relevance of a given edge e as the amount of structural distortion, measured by reliability discrepancy, caused by the unit noise subjects to the edge e, as follow.

$$\begin{aligned}\mathcal{ERR}(e) &= \frac{\Delta(\mathcal{G} + r_e)}{|r_e|} \\ &= \frac{\sum_{u,v} |R_{u,v}(\mathcal{G} + r_e) - R_{u,v}(\mathcal{G})|}{|r_e|}\end{aligned}$$

In the conventional case (deterministic graphs with edge addition and deletion $|r_e| = 1$), it amounts to the connectivity structural distortion, measured by the number of connected node pairs. In probabilistic graphs, $\mathcal{ERR}$ is used to generalize this concept by quantifying the stochastic impact of partial edge addition/deletion over the connectivity of all the possible worlds. It allows the estimation of structural distortion when r_e lies in the contentious range.

Observation 1. *Let $\mathcal{G}_e$, $\mathcal{G}_{\bar{e}}$ denote two neighbor uncertain graphs $\mathcal{G}$ with $p(e) = 1$ and $p(e) = 0$ respectively. The reliability relevance of an edge e is a constant and equivalent to the following function.*

$$\mathcal{ERR}(e) = \sum_{u,v} R_{u,v}(\mathcal{G}_e) - \sum_{u,v} R_{u,v}(\mathcal{G}_{\bar{e}}) \tag{2}$$

Observe that, the edge relevance only depends on it topological location. It amounts to the difference of the number of connected pairs between two neighbor uncertain graphs.

Proof Sketch. According to the possible world semantic and factorization rule, we can see that

$$R_{u,v}(\mathcal{G}) = p(e) \cdot R_{u,v}(\mathcal{G}_e) \ + \ \big[1 - p(e)\big] \cdot R_{u,v}(\mathcal{G}_{\bar{e}})$$

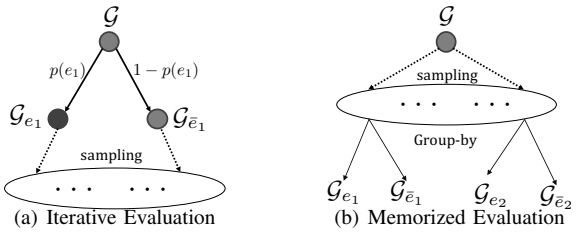
Note that, the two-terminal reliability $R_{u,v}$ in $\mathcal{G}_e$ and $\mathcal{G}_{\bar{e}}$ are constants. Therefore, two-terminal reliability discrepancy introduced by the single deviation r_e over the uncertain graph $\mathcal{G}$ is equivalent to

$$\begin{aligned}\Delta_{u,v}(\mathcal{G} + r_e) \ &= r_e \cdot R_{u,v}(\mathcal{G}_e) - r_e \cdot R_{u,v}(\mathcal{G}_{\bar{e}}) \\ &= r_e \cdot \ \big[R_{u,v}(\mathcal{G}_e) - R_{u,v}(\mathcal{G}_{\bar{e}})\big]\end{aligned}$$

Therefore, after aggregation and eliminating the factor r_e, the reliability relevance of an edge e is equivalent to Equation 2.

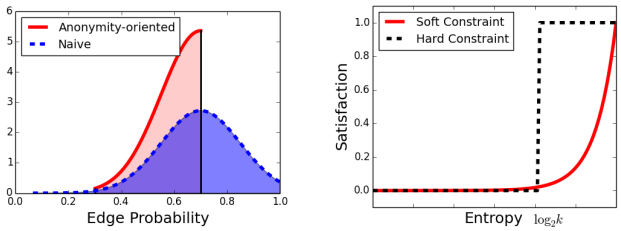
What is the relationship between Equation 2 and the conventional cut-edge definition? In the deterministic scenario, a cut-edge is an edge of a graph whose deletion increase its number of connected components. One can easily note that a cut-edge is a binary version of Equation 2, which is a continuous function regarding the edge deviation r_e and reliability discrepancy. Therefore, $\mathcal{ERR}$ is not only relevant to connectivity discrepancy but also consider its scale.

Re-visiting the Computation Challenge As ever mentioned, the evaluation of $\mathcal{ERR}(e)$ involves a fundamental problem concerning uncertain graphs, which we call the two-terminal reliability detection (TTR) problem. Since this problem is #P-complete, we focus on efficiently and accurately



(a) Iterative Evaluation (b) Memorized Evaluation

Figure 5: Sampling-based reliability detection



(a) (b)

Figure 6: (a) Anonymity-oriented edge perturbing; (b) Relaxing k-obfuscation constraint.

approximate TTR. The Monte-Carlo sampling method can be used to estimate the underling reliability of an uncertain graph. Namely, we create a subset of possible worlds of the input uncertain graph with the use of edge sampling probabilities. Then, we take the average of the number of connected node pairs in the sampled worlds as an approximation.

The $\mathcal{E}RR$ evaluation over all the edges is not trivial. One option is to iteratively invoke the sampling-based reliability computation over all the edges, as illustrated in Figure 5(a). It is straightforward to compute the connected components of a graph in linear time (regarding the numbers of the nodes and edges of the graph) using either breadth-first search or depth-first search. For each edge, we need to perform the connected component detection for N sampled graphs. Thus, the overall time complexity is $\mathcal{O}(|E| \cdot N|E|)$.

Apparently, the baseline method is inefficient when the uncertain input graph is huge. XXXXX Here, we present an efficient method which re-uses the connected components detection result of samples as illustrated in Figure 5(b). For each edge e, we group the sampled possible worlds according to the edge existence, then get the average value of cc over each group as accurate approximation of $cc(\mathcal{G}_e)$ and $cc(\mathcal{G}_{\bar{e}})$. The running time analysis roughly follows the analysis of the single-edge case. The overall time complexity is $\mathcal{O}(N|E|)$. By this way, we bring the evaluation of edge reliability relevance to the realm.

B. The stochastic uncertainty injecting scheme

As discussed in Section **??**, we inject uncertainty in the graph by assigning probability deviation to a subset of edges E_c. The selection of E_c and standard deviation assignment is described in a subsequent section. For each sampled edge e with the distributed standard deviation σ_e, we select the probability deviation r_e where $r_e \leftarrow R_{\sigma_e}$. Thus, we are left asking the question, *how can we safely alter edge probability for higher anonymity?*

It is quite straightforward in the deterministic scenario, as outlined in Eq 1. It is far more different in the uncertain scenario. A naive scheme is to inject uncertainty in a randomized way regardless of its initial existence probability. However, the randomized scheme is far from the optimal.

We will first introduce the proposed *"guided"* injection method, which we refer to as *anonymity-oriented perturbation*, and then in Section V-B1, we sketch why it works. Basically, Squidalters 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 6(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).

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 **??**, $k-$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.

$$\mathbb{C} = \prod_{v \in V} \mathsf{c}_v \tag{3}$$

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).

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.

$$C_v = e^{H(Y_{P(v)}) - \log_2 |V|} \tag{4}$$

Lemma 1. *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:*

$$\sum_{\omega \in \Omega} s(\omega) \cdot H(Y_\omega) \tag{5}$$

Proof Sketch: First we can see that

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

Taking logarithm for both sides and combining with the approximation equation 4, 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 5. It provides us with the relation between the global anonymity and the level of disorder of the degree uncertainty matrix.

Lemma 2. *The maximization of Equation 5 is equivalent to maximization of the following function:*

$$\sum_{\omega \in \Omega} s(\omega) \cdot H(Y_\omega) = \big[\sum_{v \in V} H(d_v) \big] + |V| \log |V| - |V| H(\Omega) \tag{6}$$

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

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

Lemma 3. *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. Empirical Study

We compare our proposed scheme Squid with two alternatives. The first one (Rep-An) obfuscates an uncertain graph via its extracted representative deterministic instance. The other (W-An) first casts the given uncertain graph into a weighted deterministic graph, and utilizes existing weighted graph anonymization methods.

Table I: Characteristics of the datasets and privacy parameters

Graph	Content	Nodes	Edges	Edge Prob	ϵ
PPI	Protein-Protein Interaction	12K	397K	0.29	10^{-2}
BK	Location-based OSN	58K	214K	0.29	10^{-3}
DBLP	Co-authorship Network	824K	5M	0.46	10^{-4}

A. Experiment Settings

Datasets We test them on three uncertain graphs: PPI, BrightKite (BK) and DBLP as described in Table I.

- PPI is a dataset of protein-protein interactions, provided by Disease Module Identification DREAM Challenge. The probability of any edge corresponds to the confidence that the interaction actually exists, which is obtained through biological experiments.
- Brightkite is a location-based social network. In this dataset, each node represent a user. The probability of any edge corresponds to the chance that two users visit each other.
- DBLP is a dataset of scientific publications and authors. Each node represent an author. Two authors are connected by an edge if they have co-authored in a project. The uncertainty on the edge denotes the likelihood that the two authors will collaborate in a new project.

DBLP dataset only has a few probability values, while probability values of Brightkite 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. Namely, they are difficult to be obfuscated.

Parameter Setting We consider the obfuscation level k in the range $[100, 300]$, and possible tolerance values ϵ to explore their performance difference.

Statistics For every obfuscated result, we sampled 1000 possible worlds to compute its statistics of interest: average node degree, degree distribution, average distance, graph diameter, reliability, and clustering coefficient. Note that it has been shown that 1000 possible worlds usually suffices to achieve accuracy converge. In particular, we use Hyper ANF[3] to approximate shortest path-based statistics. Here, we report their discrepancy against the original one. The smaller discrepancy, the better uncertain graph structure preserving.

VII. Conclusion

In this work, we first identify the overlooked problem–uncertain graph anonymization. We develop a new scheme, Squid, which integrates edge uncertainty into the core. It excels in identifying sanitized uncertain graphs with excellent quality. Experiments on three real-world datasets verify its effectiveness. In real-world graphs, edge probabilities sometimes are not independent, but dependent. We leave the conditional probability model as a future extension. Another extension is to investigate sharing uncertain graphs in the differentially private manner.

References

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

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

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

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

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

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

[7] W.-Y. Day, N. Li, and M. Lyu. Publishing graph degree distribution with node differential privacy. *SIGMOD*, 2016.

[8] X. Dongqing, E. Mohamed Y, and K. Xiangnan. Sharing uncertain graphs using syntactic private graph models. *ICDE*, 2018.

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

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

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

[12] M. Hua and J. Pei. Probabilistic path queries in road networks: traffic uncertainty aware path selection. *EDBT*, 2010.

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

[14] J. Lee and C. Clifton. How much is enough? choosing ϵ for differential privacy. *ISC*, 2011.

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

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

[17] L. Liu, J. Wang, J. Liu, and J. Zhang. Privacy preservation in social networks with sensitive edge weights. *SDM*, 2009.

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

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

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

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

[22] A. Sala, X. Zhao, C. Wilson, and H. Zheng. Sharing graphs using differentially private graph models. *IMC*, 2011.

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

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

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

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

[27] Q. Xiao, R. Chen, and K. Tan. Differentially private network data release via structural inference. *KDD*, 2014.

[28] X. Ying and X. Wu. Randomizing social networks: a spectrum preserving approach. *SIAM*, 2008.

[29] X. Ying, K. Pan, X. Wu, and L. Guo. Comparisons of randomization and k-degree anonymization schemes for privacy preserving social network publishing. *SNA-KDD*, 2009.

[30] B. Zhao, J. Wang, M. Li, F. Wu, and Y. Pan. Detecting protein complexes based on uncertain graph model. *TCBB*, 2014.

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