**Abstract:** **Differential privacy (DP) is a strong mathematical framework that protects data by ensuring that changing any single tuple in a database only makes the query results distribution to change slightly. However, this framework relies on the assumption of tuple independence, which may not hold true for real-world datasets, where natural dependencies exist due to social, behavioral, and genetic relationships among users. This paper identifies such vulnerabilities and offers some ways to solve them.**

**One contribution is identifying an inference attack on real datasets whereby adversaries exploit the probabilistic dependencies between tuples in differentially private query answers so as to extract sensitive user details hence breaking DP guarantees. We mitigate against this vulnerability through a dependent perturbation mechanism (DPM) that ensures privacy under DDP.**

**Our DPM consistently outperforms other methods in terms of balancing privacy and utility for dependent data through theoretical analysis as well as extensive experiments involving various query types like machine learning across several large-scale real-world datasets.**

*Index Terms*—Dependent perturbation mechanism (DPM) ,Dependent coefficient , Dependent differential privacy , Differential privacy for correlated data .

# INTRODUCTION

Shared data is essential to improving the quality of our environment, enabling innovation and academic research, and encouraging a data-driven approach. However, this raises serious privacy concerns as personal data may contain sensitive information that users would like to keep. While releasing utility-provider information is important, protecting user privacy is equally important. To address this, differential privacy (DP) has received attention due to its strong mathematical foundation in privacy protection. DP ensures minimal change in query results with each database change, and limits adversary detection to the defined private budget. However, DP assumes data independence, which is often unrealistic due to some dependencies such as social relations, which carry the risk of privacy violations.

Previous efforts, such as the Pufferfish framework, attempted to address this issue by incorporating adversarial beliefs about data relationships using a data generation model. However, this framework lacked specific perturbation algorithms to handle dependence. The Blowfish framework , a subset of Pufferfish, allowed users to define adversarial knowledge about the database through deterministic policy constraints and provided perturbation mechanisms to accommodate these constraints.

Furthermore, in network data communication using DP, researchers multiplied the sensitivity of the resulting query by the number of connected records. However, this approach added a lot of noise, which further confounded the usefulness of the classified data. This approach is elementary in our experiments.

The paper introduces Dependent Differential Privacy (DDP) to handle probability dependence constraints between tuples to ensure strong privacy guarantees To obtain these guarantees efficiently a dependent perturbation mechanism (DPM) has been used. This method computes a dependency coefficient that quantifies the probability dependence between tuples, enabling fine-grained analysis. This theory considers the difference between dependent indistinguishability and self-inscriminateness, which accounts for the large variation in query output due to tuple changes

Our contributions are:

1. Inference attack demonstration: Demonstrate the feasibility of inferring sensitive location information from private query results in a novel way by using tuple dependencies. Even partial knowledge of the user’s social network and database allows adversaries to extract more sensitive information than under traditional DP.

2. Dependent Differential Privacy Formalization: Formalize DDP to protect against adversary pre-discovered tuple dependencies. The use of dependency coefficients of the Laplace mechanism enables accurate estimation of query sensitivity for data dependencies, reducing noise and improving performance while maintaining privacy guarantees Our device is also resistant to composition attacks.

3. Analysis: Our DPM covers a variety of query tasks including machine learning and graph querying. Detailed analysis in many real worlds

# Dependent differential privacy

DP mechanisms tend to underestimate privacy risks inside the presence of dependent tuples, leading to a decline in anticipated privateness levels. Therefore, databases with such dependencies require a more potent privacy idea.

Recent studies has centered on capturing and modeling tuple dependence and correlation in databases. For instance, the Pufferfish framework , an extension of DP, considers antagonistic ideals approximately database generation throughout all feasible times. Similarly, the Blowfish framework , a subset of Pufferfish, permits customers to specify opposed information through deterministic coverage constraints.

Building on these frameworks, we formalize dependent differential privateness, a subclass of Pufferfish, which includes probabilistic dependence between tuples in statistical databases. Additionally, we propose an effective perturbation mechanism to make certain rigorous privateness guarantees. Unlike Pufferfish, there are presently no extensively-acknowledged trendy algorithms for attaining this stage of privateness.

For a database D = [D1, D2, · · · , Dn], we outline its dependence size as L, indicating that any tuple in D is depending on at maximum L − 1 different tuples. Denoted via R, the probabilistic dependence dating the various L based tuples may want to stem from the information generating method or other real-international relationships. In our method , we introduce an example of R, wherein dependence inside the Gowalla area dataset is introduced via the Gowalla social community dataset, showing probabilistic instead of deterministic behavior as in the Blowfish framework .

We outline dependent neighboring databases as follows:

Two databases D(L, R) and D**’**(L, R) are dependent neighboring databases if modifying a tuple value in D(L, R) (e.g., changing from Di in D(L, R) to D**’**i) results in changes in at most L − 1 other tuple values in D**’**(L, R) due to the probabilistic dependence relationship R between the data tuples.

Building upon dependent neighboring databases, we define our notion of dependent differential privacy:

**Definition 1 :** (-differential privacy) A randomized algorithm provides -differential privacy if for any two databases D, D**’** that differ in only a single entry, and for any output S

where (D) (resp. (D**’** )) is the output of on input D (resp. D**’**) and is the privacy budget. Smaller value of the privacy budget corresponds to a higher privacy level.

**Definition 2 :** (-Dependent Differential Privacy) A randomized algorithm A provides -dependent differential privacy, if for any pair of dependent neighboring databases D(L, R) and D**’** (L, R) and any possible output S, we have

where L denotes the dependence size and R is the probabilistic dependence relationship between the data tuples.

# Mechanism Design for DDP

In this section, we present an effective mechanism for achieving ε-dependent differential privacy and ensuring private query results over dependent tuples. We also extend the existing Laplace mechanism-based differential privacy scheme to accommodate the DDP setting.

To support arbitrary query functions, we aim to develop a principled perturbation mechanism by introducing an extra parameter, the dependence coefficient, to measure the fine-grained dependence relationship between tuples.

**A. Baseline Approach:**

The baseline approach for achieving ε-dependent differential privacy (DDP) relies on the observation that a database with dependence size L would deplete the privacy budget ε in DP by a factor of L. The theorem states that an ε/L-differentially private mechanism ***A*(D) = Q(D) + Lap(LΔQ/ε)** over a database D with dependence size L achieves ε-DDP, where Q is a query function with global sensitivity ΔQ.

However, this approach assumes complete dependence among tuples, resulting in unnecessary noise addition and rendering query outputs unusable. Real-world datasets often exhibit partial dependence among tuples. Therefore, we aim to design mechanisms that use less noise while still satisfying ε-DDP guarantees.

**B. Our Dependent Perturbation Mechanism:**

To minimize added noise, we aim to identify fine-grained dependence relationships between tuples. We start by analyzing a simple query function over a dataset with two tuples, later extending our analysis to scenarios with multiple tuples. ε-DDP requires bounding the output distributions of a randomized algorithm *A* due to changes in tuples. Motivated by the Laplace mechanism, we continue using Laplace noise for perturbation. Our objective is to find an appropriate scaling factor σ(ε) for the Laplace distribution, ensuring ε-DDP. We transform the left-hand side of the ε-DDP definition using the law of total probability and further analyze its components to derive the proper scaling factor.

Inequality from definition 1 can also be written as :

Taking LHS

where ∆Di is the maximal difference due to the change in Di . If we ignore the second term in the RHS of Eq. 3 and combine the remaining terms with Eq. 1 and Eq. 2, we obtain the scaling factor of the Laplace noise as σ() = ∆Di /  , which is exactly the same form as in traditional DP. Therefore, the LPM that satisfies DP is only a special case for our mechanism. The second term in the RHS of Eq. 5 incorporates the dependence relationship between Di , Dj and we will focus our study on this term. To evaluate the extent of dependence induced in Dj by the modification of Di , we define the dependence coefficient ρij

Next, we aim to prove that 0 ≤ ρij ≤ 1. We first have

Where is the value of dj that minimizes . Comparing Eq. 4 and Eq. 5, we have ρij ≤ 1. Furthermore, it is obvious that

Comparing Eq. 5 and Eq. 6, we have ρij ≥ 0. Finally, combining Eq. 2–5, we have

Thus, the sensitivity under the dependence between Di and Dj can be calculated as ∆Di + ρij∆Dj. The dependence coefficient ρij ∈ [0, 1] provides a nuanced measure of the relationship between two tuples. Observations about ρij include:

* ρij evaluates the dependence between Di and Dj from a privacy perspective.
* When ρij = 0, Dj's probability distribution is independent of Di, making it a special case of our analysis encompassing arbitrary tuple dependence.
* When ρij = 1, Dj can be uniquely determined by Di, representing complete dependence among tuples.
* Compared to the baseline approach, our mechanism requires less noise addition as it considers fine-grained dependence relationships.
* ρij is asymmetric, reflecting the causal and directional nature of dependence. For instance, a celebrity's social network participation may influence her fans' participation, but not necessarily vice versa.

To generalize and derive ρij for any output , we reformulate ρij to avoid the appearance of . After some manipulations we have

(Eq.10)

**Interpreting ρij :** To further understand the dependence coefficient in Eq. 10, we define the Self and Dependent Indistinguishability terms as follows:

Self Indistinguishability represents the maximal difference of Dj caused by the modification of Dj itself. We further define the Dependent Indistinguishability of Dj induced by Di as

*Dependent Indistinguishability*

(Eq.12)

*Dependent Indistinguishability* evaluates the maximal expected difference in Dj caused by the modification of Di . Therefore,

ρij = *Dependent Indistinguishability* / *Self Indistinguishability*

(Eq.13)

To generalize our dependent perturbation mechanism, we consider an arbitrary query function Q and compute the dependent sensitivity of Q over Dj induced by the modification of Di as

DSijQ = ρij∆Qj (Eq.14)

where ∆Qj is the sensitivity of Q with respect to the modification of Dj itself, i.e., ∆Qj = ||Q(· · · , dj1 , · · ·) − Q(· · · , dj2, · · ·)||1.

Furthermore, we can generalize the dependent sensitivity to multiple users as

DSQi = (ρij∆Qj) (Eq.15)

where Ci1, · · · , CiL represent the L tuples that are dependent with i-th tuple and ρii = 1. DSQi measures the dependent sensitivity of Q over all tuples in D caused by the modification of one individual tuple Di .

**Theorem 1.** The dependent sensitivity for publishing any query Q over a dependent (correlated) dataset is

DSQ= max i **(** DSQi) (Eq.16)

Finally, the dependent perturbation mechanism (DPM) for achieving -dependent differential privacy is formalized as

**Theorem 2.** *For any query function Q over an arbitrary domain D with dependent tuples, the mechanism A*

(D) = Q(D) + Lap(DSQ /) (Eq.17)

*gives*  *-dependent differential privacy.*

**Definition 3 :** ((α, β)-Accuracy): A randomization algorithm A satisfies (α, β) accuracy for a query function Q, if max D |(D) − Q(D)| < α with probability 1 − β.

Based on the definition of (α, β)-Accuracy, we have the utility guarantee for DPM as

**Theorem 3**. A DPM A that satisfies  -dependent differential privacy would achieve max D |(D) − Q(D)| < α with probability 1 − exp(−  α /DSQ ).

**Lemma 1.** Under the same privacy budget , DPM achieves better utility performance than the baseline approach.

**Lemma 2.** Under the same (α, β)-accuracy, DPM achieves better privacy performance than the baseline approach.

**C. Addressing Dependence Coefficient Challenges in System Design:**

We now tackle the practical challenge of computing the dependence coefficient ρij. Its computation relies on understanding the probabilistic models of the statistical data, which can be challenging. Here are several strategies to compute ρij:

1) Complete Knowledge of Dependence Relationship:

This approach assumes the data publisher has complete knowledge of the dependence relationship between tuples. Techniques like those use probabilistic graphical models to compute this relationship based on fully known probabilistic databases.

2) Knowledge About Data Generation:

When complete dependence information is unavailable, the data publisher can estimate the dependence relationship by analyzing the data generation process if known.

Even in cases where direct dependence information is lacking, upper bounds on the dependence coefficient can be estimated based on auxiliary data, such as using the Gowalla social datasets.

3) Realistic Scenario Challenges:

We further explore the impact of inaccuracies in computing ρij on our DPM's overall performance. Designers are typically equipped to compute ρij accurately. If ρij is overestimated, our DPM remains conservative and continues to offer rigorous DDP privacy guarantees. In cases of underestimation, two scenarios arise: if our estimated ρeij exceeds the adversary's expectation, our DPM still ensures rigorous DDP guarantees. However, if ρij is underestimated below the adversary's expectation, DDP guarantees may not be fully achieved, but our mechanism still offers improved privacy compared to traditional DP methods.

# Experimental Evaluations and Results

(Eq.18)

H(·) denotes the entropy (information) of a random variable. H(Di) evaluates the adversary’s prior information for Di without utilizing the social relationships and is the entropy of the prior probability evaluates the adversary’s posterior information after the inference attack and is the entropy of the posterior probability in Eq. 21 .

A map with many red lines

Description automatically generated

Map showing the location of users and edge representing the friend’s connection , grid has been used to estimate the location of users.

**Attack 1:**

When adversary don’t use dependencies between use

Can be also inferred as Inference Error

**Attack 2:**

Adversary take benefit of dependencies between the users.



**REFERENCES**

[1] M. E. Andres, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, ´

“Geo-indistinguishability: Differential privacy for location-based systems,” in CCS, 2013.

[2] L. Backstrom, E. Sun, and C. Marlow, “Find me if you can: improving

geographical prediction with social and spatial proximity,” in WWW,

2010.

[3] I. Bilogrevic, K. Huguenin, S. Mihaila, R. Shokri, and J.-P. Hubaux,

“Predicting users” motivations behind location check-ins and utility

implications of privacy protection mechanisms,” in NDSS, 2015.

[4] A. Blum, K. Ligett, and A. Roth, “A learning theory approach to

noninteractive database privacy,” Journal of the ACM, 2013.

[5] A. Campan and T. M. Truta, “Data and structural k-anonymity in social

networks,” in Privacy, Security, and Trust in KDD, 2009.

[6] A. Chaabane, G. Acs, M. A. Kaafar et al., “You are what you like!

information leakage through users’ interests,” in NDSS, 2012.

[7] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, “Differentially

private empirical risk minimization,” The Journal of Machine Learning

Research, 2011.

[8] R. Chen, B. C. Fung, P. S. Yu, and B. C. Desai, “Correlated network

data publication via differential privacy,” The International Journal on

Very Large Data Bases, 2014.

[9] E. Cho, S. A. Myers, and J. Leskovec, “Friendship and mobility: user

movement in location-based social networks,” in SIGKDD, 2011.

[10] T. M. Cover and J. A. Thomas, Elements of information theory. John

Wiley & Sons, 2012.

[11] I. Dinur and K. Nissim, “Revealing information while preserving

privacy,” in PODS, 2003.

14

[12] C. Dwork, “Differential privacy,” in Automata, languages and programming, 2006.

[13] ——, “Differential privacy: A survey of results,” in Theory and Applications of Models of Computation, 2008.

[14] ——, “A firm foundation for private data analysis,” Communications of

the ACM, 2011.

[15] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to

sensitivity in private data analysis,” in Springer Theory of cryptography,

2006.

[16] C. Dwork and A. Smith, “Differential privacy for statistics: What we

know and what we want to learn,” Journal of Privacy and Confidentiality, 2010.

[17] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart,

“Privacy in pharmacogenetics: An end-to-end case study of personalized

warfarin dosing,” in USENIX Security, 2014.

[18] S. R. Ganta, S. P. Kasiviswanathan, and A. Smith, “Composition attacks

and auxiliary information in data privacy,” in SIGKDD, 2008.

[19] N. Z. Gong, W. Xu, L. Huang, P. Mittal, E. Stefanov, V. Sekar,

and D. Song, “Evolution of social-attribute networks: measurements,

modeling, and implications using google+,” in IMC, 2012.

[20] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means

clustering algorithm,” Applied statistics, 1979.

[21] X. He, A. Machanavajjhala, and B. Ding, “Blowfish privacy: Tuning

privacy-utility trade-offs using policies,” in SIGMOD, 2014.

[22] S. Ji, W. Li, M. Srivatsa, and R. Beyah, “Structural data deanonymization: Quantification, practice, and implications,” in CCS,

2014.

[23] D. Kifer and B.-R. Lin, “Towards an axiomatization of statistical privacy

and utility,” in PODS, 2010.

[24] D. Kifer and A. Machanavajjhala, “No free lunch in data privacy,” in

SIGMOD, 2011.

[25] ——, “A rigorous and customizable framework for privacy,” in PODS,

2012.

[26] N. Li, T. Li, and S. Venkatasubramanian, “t-closeness: Privacy beyond

k-anonymity and l-diversity,” in ICDE, 2007.

[27] N. Li, W. Qardaji, D. Su, Y. Wu, and W. Yang, “Membership privacy:

a unifying framework for privacy definitions,” in CCS, 2013.

[28] D. Liben-Nowell and J. Kleinberg, “The link-prediction problem for

social networks,” Journal of the American society for information

science and technology, 2007.

[29] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, “l-diversity: Privacy beyond k-anonymity,” ACM Transactions on

Knowledge Discovery from Data=, 2007.

[30] F. D. McSherry, “Privacy integrated queries: an extensible platform for

privacy-preserving data analysis,” in SIGMOD, 2009.

[31] P. Mittal, C. Papamanthou, and D. Song, “Preserving link privacy in

social network based systems,” in NDSS, 2013.

[32] A. Narayanan and V. Shmatikov, “De-anonymizing social networks,” in

IEEE S&P, 2009.

[33] S. Nilizadeh, A. Kapadia, and Y.-Y. Ahn, “Community-enhanced deanonymization of online social networks,” in CCS, 2014.

[34] P. Sen and A. Deshpande, “Representing and querying correlated tuples

in probabilistic databases,” in ICDE, 2007.

[35] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux,

“Quantifying location privacy,” in IEEE S&P, 2011.

[36] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-

Y. Le Boudec, “Protecting location privacy: optimal strategy against

localization attacks,” in CCS, 2012.

[37] M. Srivatsa and M. Hicks, “Deanonymizing mobility traces: Using

social network as a side-channel,” in CCS, 2012.

[38] L. Sweeney, “k-anonymity: A model for protecting privacy,” International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,

2002.

[39] C. Task and C. Clifton, “A guide to differential privacy theory in social

network analysis,” in Proceedings of the 2012 International Conference

on Advances in Social Networks Analysis and Mining, 2012.