The VLDB Journal (2025) 34:20
https://doi.org/10.1007/s00778-024-00895-4

SPECIAL ISSUE PAPER

Differentially private explanations for aggregate query answers

Yuchao Tao[1] · **Amir Gilad**[2] · **Ashwin Machanavajjhala**[1] · **Sudeepa Roy**[1]

Received: 19 September 2024 / Revised: 19 December 2024 / Accepted: 24 December 2024

Abstract
Differential privacy (DP) is the state-of-the-art and rigorous notion of privacy for answering aggregate database queries while preserving the privacy of sensitive information in the data. In today's era of data analysis, however, it poses new challenges for users to understand the trends and anomalies observed in the query results: Is the unexpected answer due to the data itself, or is it due to the extra noise that must be added to preserve DP? In the second case, even the observation made by the users on query results may be wrong. In the first case, can we still mine interesting explanations from the sensitive data while protecting its privacy? To address these challenges, we present a three-phase framework DPXPLAIN, which is the first system to the best of our knowledge for explaining group-by aggregate query answers with DP. In its three phases, DPXPLAIN (a) answers a group-by aggregate query with DP, (b) allows users to compare aggregate values of two groups and with high probability assesses whether this comparison holds or is flipped by the DP noise, and (c) eventually provides an explanation table containing the approximately 'top-k' explanation predicates along with their relative influences and ranks in the form of confidence intervals, while guaranteeing DP in all steps. We perform an extensive experimental analysis of DPXPLAIN with multiple use-cases on real and synthetic data showing that DPXPLAIN efficiently provides insightful explanations with good accuracy and utility.

Keywords Privacy · Explanations · Aggregate queries

1 Introduction

Differential privacy (DP) [15, 41–43] is the gold standard for protecting privacy in query processing and is critically important for sensitive data analysis. It has been widely adopted by organizations like the U.S. Census Bureau [4, 39, 60, 88] and companies like Google [45, 103], Microsoft [30], and Apple [93]. The core idea behind DP is that a query answer on the original database cannot be distinguished from the same query answer on a slightly different database. This is usually achieved by adding random noise to the query answer to create a small distortion in the answer. Recent works have made significant advances in the usability of DP, allowing for complex query support [33, 58, 61, 62, 71, 95, 103], and employing DP in different settings [33, 46, 49, 81, 95, 105]. These works assist in bridging the gaps between the functionality of non-DP databases and databases that employ DP.

Automatically generating meaningful *explanations* for query answers in response to questions asked by users is an important step in data analysis that can significantly reduce human efforts and assist users. Explanations help users validate query results, understand trends and anomalies, and make decisions about next steps regarding data processing and analysis, thereby facilitating data-driven decision making. Several approaches for explaining aggregate and non-aggregate query answers have been proposed in database research, including intervention [85, 86, 104], Shapley values [69], counterbalance [77], (augmented) provenance [6, 67], responsibility [75, 76], and entropy [44] (discussed in Sect. 7).

One major gap that remains wide open is to provide explanations for analyzing query answers from sensitive data under DP. Several new challenges arise from this need. First,

✉ Amir Gilad
amirg@cs.huji.ac.il

Yuchao Tao
yctao@cs.duke.edu

Ashwin Machanavajjhala
ashwin@cs.duke.edu

Sudeepa Roy
sudeepa@cs.duke.edu

[1] Duke University, Durham, USA

[2] Hebrew University, Jerusalem, Israel

Published online: 27 January 2025

marital-status	occupation	...	education	high-income
Never-married	Machine-op-inspct	...	11th	0
Married-civ-spouse	Farming-fishing	...	HS-grad	0
Married-civ-spouse	Machine-op-inspct	...	Some-college	1
...	...	...	...	...

(a) Example of the `Adult` dataset.

Question-Phase-1: SELECT `marital-status`, AVG(`high-income`) as `avg-high-income` FROM Adult GROUP BY `marital-status`;

Answer-Phase-1:

group marital-status	Priv-answer avg-high-income	True-answer (hidden)
Never-married	0.045511	0.045480
Separated	0.064712	0.064706
Widowed	0.082854	0.084321
Married-spouse-absent	0.089988	0.092357
Divorced	0.101578	0.101161
Married-AF-spouse	0.463193	0.378378
Married-civ-spouse	0.446021	0.446133

(b) Phase-1 of DPXPLAIN: Run a query and receive noisy answers by DP. True-answers are not visible to the user and for illustration only.

Question-Phase-2: Why `avg-high-income` of group `"Married-civ-spouse"` > that of group `"Never-married"`?

Answer-Phase-2: The 95% confidence interval of group difference is $(0.399, 0.402)$, hence the noise in the query is possibly not the reason.

(c) Phase-2 of DPXPLAIN: Ask a comparison question and receive a confidence interval of the comparison.

Answer-Phase-3:

explanation predicate	Rel Influ 95%-CI L	Rel Influ 95%-CI U	Rank 95%-CI L	Rank 95%-CI U
`occupation = "Exec-managerial"`	3.25%	10.12%	1	9
`education = "Bachelors"`	2.93%	9.80%	1	8
`age = "(40, 50]"`	2.76%	9.63%	1	8
`occupation = "Prof-specialty"`	0.94%	7.81%	1	18
`relationship = "Own-child"`	-0.49%	6.38%	1	96

(d) Phase-3 of DPXPLAIN: Receive an explanation table from data for the previous question that passed Phase-2.

Fig. 1 Database instance and the three phases of the DPXPLAIN framework

in DP, the (aggregate) query answers shown to users are distorted due to the noise that must be added for preserving privacy, so the explanations need to separate the contributions of the noise from the data. Second, even after removing the effect of noise, new techniques have to be developed to provide explanations based on the sensitive data and measure their effects. For instance, standard explanation methods in non-DP settings are typically deterministic, while it is known that DP methods must be randomized. Therefore, no deterministic explanations can be provided, and even no deterministic scores or ranks of explanations can be displayed in response to user questions if we want to guarantee DP in the explanation system. Third, the system needs to ensure that the returned explanations, scores, and ranks still have high accuracy while being private.

In this paper, we propose DPXPLAIN, a novel three-phase framework that generates explanations[1] under DP for aggregate queries based on the notion of *intervention* [86, 104][2]. DPXPLAIN surmounts the aforementioned challenges and is the first system combining DP and explanations to the best of our knowledge. We illustrate DPXPLAIN through an example.

Example 1.1 Consider the `Adult` (a subset of Census) dataset [36] with 48,842 tuples. We consider the following attributes: `age`, `workclass`, `education`, `marital-status`, `occupation`, `relationship`, `race`, `sex`, `native-country`, and `high-income`, where `high-income` is a binary attribute indicating whether the income of a person is above 50K or not; some relevant columns are illustrated in Fig. 1a.

In the **first phase (Phase-1)** of DPXPLAIN, the user submits a query and gets the results as shown in Fig. 1b. This query is asking the fraction of people with high income in each `marital-status` group. As Fig. 1b shows, the framework returns the answer with two columns: `group` and `Priv-answer`. Here `group` corresponds to the group-by attribute `marital-status`. However, since the data is private, instead of seeing the actual aggregate values `avg-high-income`, the user sees a perturbed answer `Priv-answer` for each group as output by some differentially private mechanism with a given privacy budget (here computed by the Gaussian mechanism with privacy budget $\rho = 0.1$ [15]). The third column `True-answer` shown in grey (hidden for users) in Fig. 1b shows the **true aggregated output** for each group.

In the **second phase (Phase-2)** of DPXPLAIN, the user selects two groups to compare their aggregate values and asks for explanations. However, unlike standard explanation frameworks [44, 67, 77, 86, 104] where the answers to a query are correct and hence the question asked by the user is also correct, in the DP setting, the answers that the users see are perturbed. Therefore, the user question and the direction of comparison may not be valid. Hence our system first tests the validity of the question. If the question is valid, our system provides a data-dependent explanation of the user question. We explain this below with the running example.

First, consider the question in Fig. 2 comparing the last two groups in Fig. 1b (spouse in armed forces vs. a civilian). In this example, even though the noisy `avg-high-income` for `"Married-AF-spouse"` is larger than the noisy value

[1] The explanations we provided should not be considered causal explanations.

[2] See [101] for a graphical user interface for DPXPLAIN.

Question-Phase-2: Why avg-high-income of group `"Married-AF-spouse"` > that of group `"Married-civ-spouse"`?

Answer-Phase-2: The 95% confidence interval of group difference is $(-0.259, 0.460)$, hence the noise in the query is possibly the reason.

Fig. 2 A user question explained by high noise

for `"Married-civ-spouse"`, this might not be true in the real data (as is the case in the `True-answer` column). Hence, our system tests whether the user question could potentially be explained just using the noise introduced by DP rather than from the data itself. To do this, our system tests the validity of the user question by computing a confidence interval around the difference between these two outputs. In this case, the confidence interval is $(-0.259, 0.460)$. Since it includes 0 and negative values, we cannot conclude with high probability that `"Married-AF-support"` > `"Married-civ-spouse"` is true in the original data. **Since the validity of the user question is uncertain, we know that any further explanation might not be meaningful and the user may choose to stop here.** In other words, the explanation for the comparison in the user question is primarily attributed to the added noise by the DP mechanism. If the user chooses to proceed to the next phase for further explanations from the data, they might not be meaningful.

Now consider the comparison between two other groups `"Never-married"` and `"Married-civ-spouse"`, in Fig. 1c. In this case, the confidence interval about the difference does not include zero and is tight around a positive number of 0.4, which indicates that the user question is correct with high probability. Notice that it is still possible for a valid question to have a confidence interval that includes zero given sufficiently large noise. Since the question is valid, the user may continue to the next phase.

In the **third phase (Phase-3)** of DPXPLAIN, for the questions that are likely to be valid, DPXPLAIN can provide a further detailed data-dependent explanation for the question. To achieve this again with DP, our framework reports an "**Explanation Table**"[3] to the user as Fig. 1d shows, which includes the top-5 *explanation predicates*. The explanation predicates explain the user question using the notion of *intervention* as done in previous work [86, 104] for explaining aggregate queries in the non-DP setting. Intuitively, if we intervene in the database by (hypothetically) removing tuples that satisfy the predicate, and re-evaluate the query, then the difference in the aggregate values of the two groups mentioned in the question will reduce. In the simplest form, explanation predicates are singleton predicates of the form "`attribute = <value>`", while in general, our framework supports more complex predicates involving conjunction, disjunction, and comparison ($>, \geq$ etc.). In Fig. 1d, the top-5 simple explanation predicates, as computed by DPXPLAIN, are shown out of 103 singleton predicates, according to their influences on the question but perturbed by noises to satisfy DP. The amount of noise is proportional to the sensitivity of the influence function, the maximum possible change of the influence of any explanation predicate when adding or removing a single tuple from the database. Once the top-5 predicates are selected, the explanation table also shows both their *relative influence* (intuitively, how much they affect the difference of the group aggregates in the question) and their *ranks* (that might be far away from the true top-5) in the form of confidence interval under DP.

From this table, `occupation = "Exec-managerial"` is returned as the top explanation predicate, indicating that the people with this job contribute more to the average high income of the married group compared to the never-married group. In other words, managers tend to earn more if they are married than those who are single, which probably can be attributed to the intuition that married people might be older and have more seniority, which is consistent with the third explanation `age = "(40, 50]"` in Fig. 1d as well. Although these explanations are chosen at random, we observe that the first three explanations are almost constantly included. This is consistent with the narrow confidence interval of rank for the first three explanation predicates, which are all around [1, 8]. Looking at the confidence intervals of the relative influence and ranks in the explanation table, the user also knows that the first three explanations are likely to have some effect on the difference between the married and unmarried groups. However, for the last two explanations, the confidence intervals of influences are closer to 0 and the confidence intervals of ranks are wider, especially for the fifth one which includes negative influences in the interval and has a wide range of possible ranks (96 out of 103 simple explanation predicates in total).

Our contributions

- We develop DPXPLAIN, the first framework, to our knowledge, that generates explanations for query answers under DP adapting the notion of intervention [86, 104]. It explains user questions comparing two group-by aggregate query answers (COUNT, SUM, or AVG) with DP in three phases: private query answering, private user question validation, and private explanation table. We also discuss the extension of user questions to more than two answers.
- We develop multiple novel techniques that allow DPXPLAIN to provide explanations under DP including (a)

[3] We note that our notion of explanation table is unrelated to that described by Gebaly et al. [44] for summarizing dimension attributes to explain a binary outcome attribute.

computing confidence intervals to check the validity of user questions, (b) choosing explanation predicates, and (c) computing confidence intervals around the influence and rank of the predicates.
- We design a low sensitivity influence function inspired by previous work on non-private explanations [104], which is the key to the accurate selection of the top-k explanation predicates.
- We design an algorithm that uses a noisy binary search technique to find the confidence intervals of the explanation ranks, which overcomes the high sensitivity challenge of the rank function.
- We have implemented a prototype of DPXPLAIN [2] to evaluate our approach. We include two case studies on a real and a synthetic dataset showing the entire process and the obtained explanations. We have further performed a comprehensive accuracy and performance evaluation, showing that DPXPLAIN correctly indicates the validity of the question with 100% accuracy for 8 out of 10 questions, selects at least 80% of the true top-5 explanation predicates correctly for 8 out of 10 questions, and generates descriptions about their influences and ranks with high accuracy.

Extensions of the conference paper This paper is an extended version of our PVLDB 2023 paper [94] and includes full proofs for all the lemmas, propositions and theorems (Sects. 2 and 4), a detailed description of all algorithms of phases 1, 2 and 3 including pseudocodes (Sect. 4), a generalized form of our model (Sect. 5), and a use-case demonstrating it (Sect. 6).

2 Preliminaries

We now give the necessary background for our model. The DPXPLAIN framework supports single-block SELECT - FROM - WHERE - GROUP BY queries with aggregates (Fig. 3) on single tables,[4]. Hence the database schema $\mathbb{A} = (A_1, \ldots, A_m)$ is a vector of attributes of a single relational table. Each attribute A_i is associated with a domain $\texttt{dom}(A_i)$, which can be continuous or categorical. A database (instance) D over a schema $\mathbb{A}$ is a bag of tuples (duplicate tuples are allowed) $t_i = (a_1, \ldots, a_m)$, where $a_i \in \texttt{dom}(A_i)$ for all i. The domain of a tuple is denoted as $\texttt{dom}(\mathbb{A}) = \texttt{dom}(A_1) \times \texttt{dom}(A_2) \times \ldots \times \texttt{dom}(A_m)$. We denote $A_i^{max} = \max\{|a| \mid a \in \texttt{dom}(A_i)\}$ as the maximum absolute value of A_i. The value of the attribute A_i of tuple t is denoted by $t.A_i$.

q =SELECT A_{gb}, agg(A_{agg}) FROM D
WHERE ϕ GROUP BY A_{gb};

Fig. 3 Group-by query with aggregates supported by DPXPLAIN. The true results are denoted by (α_i, o_i) and the noisy results released by a DP mechanism are denoted by $(\alpha_i, \hat{o}_i)$ where α_i is the value of A_{gb} and $o_i, \hat{o}_i$ are aggregate values

We consider group-by aggregate queries q of the form shown in Fig. 3. Here A_{gb} is the group-by attribute and A_{agg} is the aggregate attribute, ϕ is a predicate without subqueries, and $\texttt{agg} \in \{COUNT, SUM, AVG\}$ is the aggregate function. When query q is evaluated on database D, its result is a set of tuples (α_i, o_i), where $\alpha_i \in \texttt{dom}(A_{gb})$ and $o_i = agg(\{t.A_{agg} \mid t \in D, \phi(t) = true, t.A_{gb} = \alpha_i\})$. For brevity, we will use $\phi'(D)$ to denote $\{t \mid \phi'(t) = true\}$ for any predicate ϕ', and $agg(A_{agg}, D')$, or simply $agg(D')$ when it is clear from context, to denote $agg(\{t.A_{agg} \mid t \in D'\})$ for any $D' \subseteq D$. Hence, $o_i = agg(A_{agg}, g_i(D))$, where $g_i = \phi \wedge (A_{gb} = \alpha_i)$.

Example 2.1 Consider Example 1.1. The schema is $\mathbb{A}$ = (marital-status, occupation, age, relationship, race, workclass, sex, native-country, education, high-income). All the attributes are categorical attributes and the domain of high-income is $\{0, 1\}$. The query is shown in Fig. 1b and the true result for each group is shown in the True-answer column. Here $A_{gb} = \texttt{marital} - \texttt{status}$, $A_{agg} = \texttt{high} - \texttt{income}$, and $\texttt{agg} = AVG$.

Differential Privacy In this work, we consider query-answering and providing explanations using *differential privacy (DP)* [42] to protect private information in the data. In standard databases, a query result can give an adversary the option to find the presence or absence of an individual in the database, compromising their privacy. DP allows users to query the database without compromising the privacy by guaranteeing that the query result will not change too much (defined in the sequel) even if it is evaluated on any two different but *neighboring* databases defined below.

Definition 2.1 (Neighboring Database) Two databases D and D' are neighboring (denoted by $D \approx D'$) if D' can be transformed from D by adding or removing [5] a tuple in D.

[4] Unlike some standard explanation framework [104], in DP, we cannot consider materialization of join-result for multiple tables, since the privacy guarantee depends on *sensitivity* and removing one tuple from a table may change the join and query result significantly. We leave it as an interesting future work.

[5] There are two variants of neighboring databases. The definition by addition/deletion of tuples is called "unbounded DP", and by updating tuples is called "bounded DP", since the size of data is fixed. In this work, we assume the unbounded version, while DPXPLAIN can be adapted also for the bounded version by adapting the noise scale.

In this paper, we consider a relaxation of DP called **ρ-zero-concentrated differential privacy (zCDP)** [15, 43] for several reasons, *and refer to it simply as DP if not otherwise stated.* First, we use Gaussian noise to perturb query answers and derive confidence intervals, which does not satisfy pure ϵ-DP [42] but satisfies approximate (ϵ, δ)-DP [42] and ρ-zCDP. Second, ρ-zCDP only has one parameter ρ, compared to (ϵ, δ)-DP which has two parameters, so it is easier to understand and control. Third, ρ-zCDP allows for tighter analyses for tracking the *privacy budget* (controlled by ρ) over multiple private releases, which is the case for this framework. A lower ρ value implies a lower privacy loss.

Definition 2.2 (Zero-Concentrated Differential Privacy (zCDP) [15]) A mechanism $\mathcal{M}$ is said to be *ρ-zero-concentrated differential private*, or ρ-zCDP for short, if for any neighboring datasets D and D' and all $\alpha \in (1, \infty)$ it holds that

$$D_\alpha(\mathcal{M}(D) \| \mathcal{M}(D')) \leq \rho\alpha$$

where $D_\alpha(\mathcal{M}(D) \| \mathcal{M}(D'))$ denotes the Rényi divergence of the distribution $\mathcal{M}(D)$ from the distribution $\mathcal{M}(D')$ at order α [78].

A popular approach for providing zCDP to a query result is to add Gaussian noise to the result before releasing it to a user. This approach is called *Gaussian mechanism* [15, 42].

Definition 2.3 (Gaussian Mechanism) Given a query q and a noise scale σ, Gaussian mechanism $\mathcal{M}^G$ is given as:

$$\mathcal{M}^G(D; q, \sigma) = q(D) + N(0, \sigma^2)$$

where $N(0, \sigma^2)$ is a random variable from a normal distribution[6] with mean zero and variance σ^2.

Example 2.2 Suppose there is a database D with 100 tuples. Consider a query q = "`SELECT COUNT(*) FROM D`", which counts the total number of tuples in a database D. Here $q(D) = 100$. Now we use Gaussian mechanism to release $q(D)$, which is to randomly sample a noise z from distribution $N(0, \sigma^2)$. Here we assume $\sigma = 1$. Finally, we got a noisy result $\hat{q}(D) = 102.32$, which we may round to an integer in postprocessing without sacrificing the privacy guarantee (Proposition 2.1 below).

The privacy guarantee from the Gaussian mechanism depends on both the noise scale it uses and the sensitivity of the query. Query sensitivity reflects how sensitive the query is to the change of the input. More noise is needed for a more sensitive query to achieve the same level of privacy protection.

[6] The probability density function of a normal distribution $N(\mu, \sigma^2)$ is given as $exp(-((x-\mu)/\sigma)^2/2)/(\sigma\sqrt{2\pi})$.

Definition 2.4 (Sensitivity) Given a scalar query q that outputs a single number, its sensitivity is defined as:

$$\Delta_q = \sup_{D \approx D'} |q(D) - q(D')|$$

Example 2.3 Continuing Example 2.2, since the query q returns the database size, for any two neighboring databases, their sizes always differ by 1, so the sensitivity of q is 1.

Theorem 2.1 *(Gaussian Mechanism [15]) Given a query q with sensitivity Δ_q and a noise scale σ, its Gaussian mechanism $\mathcal{M}^G$ satisfies $(\Delta_q^2/2\sigma^2)$-zCDP. Equivalently, given a privacy budget ρ, choosing $\sigma = \Delta_q/\sqrt{2\rho}$ in Gaussian mechanism satisfies ρ-zCDP.*

Composition Rules In our analysis, we will use the following standard composition rules and other known results from the literature of DP [74] (in particular, zCDP [15]) frequently:

Proposition 2.1 *The following holds for zCDP [15, 74]:*

- **Parallel composition:** *if mechanisms take disjoint data as input, the total privacy loss is the maximum privacy loss from each.*
- **Sequential composition:** *if mechanisms take overlapping data as input, the total privacy loss is the sum of each privacy loss.*
- **Post-processing:** *if we run a mechanism and post-process the result without accessing the data, the total privacy loss is only the privacy loss from the mechanism.*

We next survey several basic results that will come in handy in the sequel.

Lemma 2.1 *(Chernoff bound of Q function) Given a Q function: $Q(x) = Pr[X > x]$, where $X \sim N(0, 1)$ is a standard Gaussian distribution, if $x \geq 0$, we have $Q(x) \leq \exp(-x^2/2)$.*

Proof By Chernoff bound, we have $Pr[X > x] \leq E[e^{tX}]/e^{tx}$ for any $t \geq 0$. By the property of Gaussian distribution, we have $E[e^{tX}] = e^{t^2/2}$. Together, we have $Pr[X > x] \leq e^{t^2/2 - tx}$. Since $x \geq 0$, we can choose $t = x$, and have $Pr[X > x] \leq e^{-x^2/2}$. □

Lemma 2.2 *Given two functions f ang g with sensitivities Δ_f and Δ_g, the sum of two functions have sensitivity $\Delta_f + \Delta_g$*

Proof BY definition, we have $\max_{D \approx D'} |f(D) - f(D')| \leq \Delta_f$ and $\max_{D \approx D'} |g(D) - g(D')| \leq \Delta_g$. Therefore, $\max_{D \approx D'} |(f(D) + g(D)) - (f(D') + g(D'))| = \max_{D \approx D'} |(f(D) - f(D') + (g(D) - g(D'))| \leq \max_{D \approx D'} |(f(D) - f(D')| + \max_{D \approx D'} |(g(D) - g(D'))| = \Delta_f + \Delta_g$. The inequality is due to the property of absolute. □

Lemma 2.3 *(Gaussian Confidence Interval [102]) Given a Gaussian random variable $Z \sim N(\mu, \sigma^2)$ with unknown location parameter μ and known scale parameter σ. Let $\mathcal{I}^L = Z - \sigma\sqrt{2}\,\mathrm{erf}^{-1}(\gamma)$ and $\mathcal{I}^U = Z + \sigma\sqrt{2}\,\mathrm{erf}^{-1}(\gamma)$, then $\mathcal{I} = (\mathcal{I}^L, \mathcal{I}^U)$ is a γ level confidence interval of μ.*

Lemma 2.4 *Given events $A_1, A_2, \ldots, A_\ell$, the following inequality holds:*

$$Pr[\bigwedge_{i=1}^{\ell} A_i] \geq \sum_{i=1}^{\ell} Pr[A_i] - (\ell - 1)$$

Proof First we show that given events A and B, we have $Pr[A \wedge B] \geq Pr[A] + Pr[B] - 1$ since $1 \geq Pr[A \vee B] = Pr[A] + Pr[B] - Pr[A \wedge B]$. Next we show that

$$Pr[\bigwedge_{i=1}^{\ell} A_i] \geq Pr[\bigwedge_{i=1}^{\ell-1} A_i] + Pr[A_\ell] - 1$$

using the previous rule. This gives a recursive expression and can be reduced to the final formula in the lemma. □

Lemma 2.5 *Given a COUNT or SUM query q with sensitivity Δ_q, a predicate ϕ, a non-negative query $f : \mathcal{D} \to \mathbb{N}_0$ with sensitivity 1 and another monotonic[7] and positive query $g : \mathcal{D} \to \mathbb{N}^+$ with sensitivity 1. Denote $h(D) = q(\phi(D))\frac{f(D)}{g(D)}$. For any two neighboring datasets D and D' such that $|D'| = |D| + 1$, we have*

$$|h(D') - h(D)| \leq \frac{2|\phi(D)| + f(D) + 1}{g(D)}\Delta_q$$

Proof Denote $x = q(\phi(D))$, $x' = q(\phi(D'))$, $n = |\phi(D)|$. Since x is the aggregation over tuples from $\phi(D)$ and x has sensitivity Δ_q, we have $|x| \leq n\Delta_q$. Denote $\delta_x = x' - x$. Since x has sensitivity Δ_q, we have $|\delta_x| \leq \Delta_q$. Since $g(D)$ is monotonic and has sensitivity 1, we have $g(D) \leq g(D') \leq g(D) + 1$. Since f has sensitivity 1, we have $|f(D) - f(D')| \leq 1$.

$$\begin{aligned}
&|h(D') - h(D)| \\
=&|x'\frac{f(D')}{g(D')} - x\frac{f(D)}{g(D)}| \\
=&|(x + \delta_x)\frac{f(D')}{g(D')} - x\frac{f(D)}{g(D)}| \\
=&|x\left(\frac{f(D')}{g(D')} - \frac{f(D)}{g(D)}\right) + \delta_x\frac{f(D')}{g(D')}|
\end{aligned}$$

Now we divide into two cases depending on the sign of the factor of x in the formula above.

[7] A query q is monotonic if for any two databases D' and D such that $|D'| \geq |D|$, we have $q(D') \geq q(D)$.

Case 1 the factor of x is non-negative.

$$\begin{aligned}
&|h(D') - h(D)| \\
\leq&n\Delta_q\left(\frac{f(D')}{g(D')} - \frac{f(D)}{g(D)}\right) + \Delta_q\frac{f(D')}{g(D')} \\
=&\left[(n+1)\frac{f(D')}{g(D')} - n\frac{f(D)}{g(D)}\right]\Delta_q \\
\leq&\left[(n+1)\frac{f(D)+1}{g(D)} - n\frac{f(D)}{g(D)}\right]\Delta_q \\
\leq&\frac{f(D)+n+1}{g(D)}\Delta_q
\end{aligned}$$

Case 2 the factor of x is non-positive.

$$\begin{aligned}
&|h(D') - h(D)| \\
\leq&n\Delta_q\left(\frac{f(D)}{g(D)} - \frac{f(D')}{g(D')}\right) + \Delta_q\frac{f(D')}{g(D')} \\
\leq&\left[n\left(\frac{f(D')+2}{g(D')} - \frac{f(D')}{g(D')}\right) + \frac{f(D')}{g(D')}\right]\Delta_q \\
\leq&\frac{2n + f(D')}{g(D')}\Delta_q \\
\leq&\frac{2n + f(D) + 1}{g(D)}\Delta_q
\end{aligned}$$

In conclusion, $|h(D') - h(D)| \leq \frac{2n+f(D)+1}{g(D)}\Delta_q$. □

Private Query Answering Recall that we have group-by aggregation query of the form q = `SELECT` A_{gb}, `agg(`A_{agg}`) FROM D WHERE` ϕ `GROUP BY` A_{gb}, and it returns a list of tuples (α_i, o_i) where $\alpha_i \in \mathtt{dom}(A_{gb})$ and o_i is the corresponding aggregate value. Since no single tuple can exist in more than one group, adding or removing a single tuple can at most change the result of a single group. As mentioned earlier, Phase-1 returns noisy aggregate values $\hat{o}_i$ for each α_i instead of o_i. The following holds:

Observation 2.1 *According to the* parallel composition rule *(Proposition 2.1), if for each α_i, its (noisy) aggregate value $\hat{o}_i$ is released under ρ_q-zCDP, the entire release of results including all groups $\{\alpha_i, \hat{o}_i : \alpha_i \in \mathtt{dom}(A_{gb})\}$ satisfies ρ_q-zCDP.*

For a $COUNT$ or SUM query, we use the Gaussian mechanism for each group α_i: $\hat{o}_i = o_i + N(0, \sigma^2)$, where the noise scale $\sigma = \Delta_q/\sqrt{2\rho_q}$ to satisfy ρ_q-zCDP by Theorem 2.1. The sensitivity term Δ_q is 1 for $COUNT$ and A_{agg}^{max} for SUM, the maximum absolute value of the aggregation attribute in its domain. For an AVG query, since $AVG = SUM/COUNT$, we decompose it into a SUM and a $COUNT$ query, privately answer each of them by half of the privacy budget $\rho_q/2$ to get $\hat{o}_i^S$ and $\hat{o}_i^C$ for each group α_i, and release $\hat{o}_i = \hat{o}_i^S/\hat{o}_i^C$ as a post-processing step. The

noisy query answers of the group-by query with AVG satisfy ρ_q-zCDP by the sequential composition rule (Proposition 2.1).

Confidence Level and Interval Confidence intervals are commonly used to determine the error margin in uncertain computations and are used in various fields including machine learning [57] and DP [47]. In our context, we use confidence intervals to measure the uncertainty in the user question and our explanations.

Definition 2.5 (Confidence Level and Interval [102]) Given a confidence level γ and an unknown but fixed parameter θ, a random interval $\mathcal{I} = (\mathcal{I}^L, \mathcal{I}^U)$ is said to be its confidence interval, or CI, with confidence level γ if the following holds:

$$Pr[\mathcal{I}^L \leq \theta \leq \mathcal{I}^U] \geq \gamma$$

Example 2.4 Let $\theta = 0$. Suppose with probability 50% we have $I^L = -1$ and $I^U = 1$, and with another probability 50% we have $I^L = 1$ and $I^U = 2$. Therefore, $Pr[\mathcal{I}^L \leq \theta \leq \mathcal{I}^U] = 50\%$, and we can conclude that the random interval $\mathcal{I} = (\mathcal{I}^L, \mathcal{I}^U)$ is a 50% level confidence interval for θ.

3 Private explanations in DPXPLAIN

In this section, we provide the model for private explanations of query results at the center of DPXPLAIN.

User Question and Standard Explanation Framework In Phase-2 of DPXPLAIN, given the noisy results of a group-by aggregation query from Phase-1, users can ask questions comparing the aggregate values of two groups[8]

Definition 3.1 (User Question) Given a database D, a group-by aggregate query q as shown in Fig. 3, a DP mechanism $\mathcal{M}$, and two noisy answer tuples $(\alpha_i, \hat{o}_i), (\alpha_j, \hat{o}_j) \in \mathcal{M}(D; q)$ where $\hat{o}_i > \hat{o}_j$, a *user question* has the form *"why is the (noisy) aggregate value $\hat{o}_i$ of group α_i larger than the aggregate value $\hat{o}_j$ of group α_j?"*), which is denoted by "why $(\alpha_i, \alpha_j, >)$?".

Example 3.1 The question from Fig. 1c is denoted as *"why* `('Married-civ-spouse', 'Never-married', >)`?".

To explain a user question, several previous approaches return top-k predicates that have the highest influences over the group difference in the question [44, 67, 86, 104]. We follow this paradigm and define explanation predicates.

Definition 3.2 (Explanation Predicate) Given a database D with a set of attributes $\mathbb{A}$, a group-by aggregation query q (Fig. 3) with group-by attribute A_{gb} and aggregate attribute A_{agg} and a predicate size l, an explanation predicate p is a Boolean expression of the form $p = \varphi_1 \wedge ... \wedge \varphi_l$, where each φ_i has the form $A_i = a_i$ such that $A_i \in \mathbb{A} \setminus \{A_{gb}, A_{agg}\}$ is an attribute, and $a_i \in \texttt{dom}(A_i)$ is its value.

We assume $\texttt{dom}(A_i)$ is discrete, finite, and data-independent. We focus here on the conjunction of equality predicates. However, our framework can also handle predicates that contain disjunctions and inequalities of the form $A_i \circ a_i$ where $\circ \in \{>, <, \geq, \leq, \neq\}$ when the constant a_i is from a finite and data-independent set.

New challenges for explanations with DP Unlike standard explanation framework on aggregate queries [67, 86, 104], the existing frameworks are not sufficient to support DP and need to be adapted: (i) the question itself might not be valid due to the noise injected into the queries, (ii) the selection of top-k explanation predicates needs to satisfy DP, which further requires the influence function to have low sensitivity so that the selection is less perturbed, and (iii) since the selected explanation predicates are not guaranteed to be the true top-k, it is also necessary to output extra descriptions under DP for each selected explanation predicate about their actual influences and ranks. We detail the adjustments as follows.

Question Validation with DP (Phase-2) While the user is asking "why is $\hat{o}_i > \hat{o}_j$?", in reality, it may be the case that the true results satisfy $o_i \leq o_j$, i.e., they have opposite relationship than the one observed by the user. This indicates that $\hat{o}_i > \hat{o}_j$ is the result of the noise being added to the results. In this scenario, one option to explain the user's observation of $\hat{o}_i > \hat{o}_j$ will be releasing the true values (equivalently, the added exact noise values), which will violate DP. Instead, to provide an explanation in such scenarios, we generate a confidence interval for the difference of two (hidden) aggregate values $o_i - o_j$, which can include negative values (discussed in detail in Sect. 4.1). This leads to the first problem we need to solve in DPXPLAIN:

Theorem 3.1 *(Private Confidence Interval of Question) Given a dataset D, a query q, a DP mechanism $\mathcal{M}$, a privacy budget ρ_q, a confidence level γ, and a user question $(\alpha_i, \alpha_j, >)$ on the noisy query answers output by $\mathcal{M}$ satisfying ρ_q-zCDP, find a confidence interval (see Definition 2.5) for the user question $\mathcal{I}_{uq} = (\mathcal{I}^L_{uq}, \mathcal{I}^U_{uq})$ for $o_i - o_j$ at confidence level γ without extra privacy cost.*

In Phase-2, the framework returns a confidence interval of $o_i - o_j$ to the user. If it includes zero or negative numbers, it is possible that $o_i \leq o_j$, and the user's observation of $\hat{o}_i > \hat{o}_j$ is the result of the noise added by the DP mechanism. In such cases, the user may stop at Phase-2. If the user is satisfied with the confidence interval for the validity of the question, she can proceed.

[8] Our framework can handle more general user questions involving single group or more than two groups; see more details in Sect. 5.

Influence Function (Phase-3) When considering DP, the order of the explanation predicates is perturbed by the noise we add to the influences according to the sensitivity of the influence function (discussed in detail in Sect. 4.3.1). To provide useful explanations, this sensitivity needs to be low, which means the influence does not change too much by adding or removing a tuple from the database. For example, a counting query that outputs the database size n has sensitivity 1, since its result can only change by 1 for any neighboring databases. Following this concept, we propose a core problem for the DPXPLAIN framework, which is also critical to the subsequent problems defined below.

Theorem 3.2 *(Influence Function with Low Sensitivity) Find an influence function* $\text{INF} : \mathcal{P} \to \mathcal{R}$ *that maps an explanation predicate to a real number and has low sensitivity.*

Private Top-k Explanations (Phase-3) In DPXPLAIN, to satisfy DP, in Phase-3 we output the top-k explanation predicates ordered by the noisy influences, and release the influences and ranks of these predicates in the form of confidence intervals to describe the uncertainty. To achieve this goal, we tackle the following three sub-problems.

Theorem 3.3 *(Private Top-k Explanation Predicates) Given a set of explanation predicates* $\mathcal{P}$*, an integer* k*, and a privacy parameter* ρ_{Topk}*, find the top-k highest influencing predicates* $p_1, p_2, \ldots, p_k$ *from* $\mathcal{P}$ *while satisfying* ρ_{Topk}*-zCDP.*

Theorem 3.4 *(Private Confidence Interval of Influence) Given a confidence level* γ*,* k *explanation predicates* $p_1, p_2, \ldots, p_k$*, and a privacy parameter* ρ_{Influ}*, find a confidence interval* $\mathcal{I}_{influ} = (\mathcal{I}^L_{influ}, \mathcal{I}^U_{influ})$ *for influence* $\text{INF}(p_u)$ *at confidence level* γ *for each* $u \in \{1, \ldots, k\}$ *satisfying* ρ_{Influ}*-zCDP (overall privacy budget).*

Theorem 3.5 *(Private Confidence Interval of Rank) Given a confidence level* γ*,* k *explanation predicates* $p_1, p_2, \ldots, p_k$*, and a privacy parameter* ρ_{Rank}*, find a confidence interval* $\mathcal{I}_{rank} = (\mathcal{I}^L_{rank}, \mathcal{I}^U_{rank})$ *for rank of* p_u *at confidence level* γ *for each* $u \in \{1, \ldots, k\}$ *satisfying* ρ_{Rank}*-zCDP (overall privacy budget).*

4 Computing explanations under DP

Next we provide solutions to problems 3.1, 3.2, 3.3, 3.4, and 3.5 in Sects. 4.1, 4.2, 4.3.1, 4.3.2, and 4.3.3 respectively, and analyze their properties. We summarize the entire DPXPLAIN framework in Sect. 4.4.

4.1 Confidence interval for a user question

For **Theorem** 3.1, the goal is to find a confidence interval of $o_i - o_j$ for the user question at the confidence level γ without extra privacy cost in Phase-2. We divide the solution into two cases. (1) When the aggregation is COUNT or SUM, the noisy difference $\hat{o}_i - \hat{o}_j$ follows Gaussian distribution, which leads to a natural confidence interval. (2) When the aggregation is AVG, the noisy difference does not follow Gaussian distribution, but we show that the confidence interval in this case can be derived through multiple partial confidence intervals. The solutions below only take the noisy query result as input, which does not incur extra privacy loss according to the post-processing property of DP (Proposition 2.1).

We now describe the pseudo code for the algorithm.

Confidence interval for COUNT and SUM In Algorithm 1, at line 2, we set the noise scale σ according to aggregation as $COUNT$ (SUM), and at line 6 and 7, we set the confidence interval from the standard properties of Gaussian distribution by a margin as $\sqrt{2}(\sqrt{2}\sigma)\,\text{erf}^{-1}(\gamma)$ for both bounds [9] [102].

Confidence interval for AVG In Algorithm 1, at line 9, we set the sub confidence level $\beta = 1 - (1-\gamma)/4$ for each individual confidence interval, so that the final confidence level for $o_i - o_j$ is γ. At line 10 and 11, we set the noise level σ for SUM and $COUNT$. From line 12 to 16, we extract all the intermediate numerators and denominators, and construct individual confidence intervals. At line 17 and 18, we compute the infimum and supremum of the image of the cross product of individual confidence intervals, which is also the confidence interval at level γ.

Algorithm 1 Compute Confidence Interval of User Question

Require: A user question $Q = (\alpha_i, >, \alpha_j)$ with respect to the query `SELECT` A_{gb}, `agg(`A_{agg}`) FROM R WHERE` ϕ `GROUP BY` A_{gb}, the noisy results $\hat{o}_i$ and $\hat{o}_j$, the privacy budget ρ_q for the private query answering, and the confidence level γ.

Ensure: A γ-level confidence interval of $o_i - o_j$.

1: **if** agg $= COUNT$ or agg $= SUM$ **then**
2: **if** agg $= COUNT$ **then**
3: $\sigma \leftarrow 1/\sqrt{2\rho_q}$
4: **else if** agg $= SUM$ **then**
5: $\sigma \leftarrow A^{max}_{agg}/\sqrt{2\rho_q}$
6: $\mathcal{I}^L \leftarrow \hat{o}_i - \hat{o}_j - 2\sigma\,\text{erf}^{-1}(\gamma)$
7: $\mathcal{I}^U \leftarrow \hat{o}_i - \hat{o}_j + 2\sigma\,\text{erf}^{-1}(\gamma)$
8: **else if** agg $= AVG$ **then**
9: $\beta \leftarrow 1 - (1-\gamma)/4$
10: $\sigma_S \leftarrow A^{max}_{agg}/\sqrt{2\rho_q/2}$
11: $\sigma_C \leftarrow 1/\sqrt{2\rho_q/2}$
12: **for** $t \in \{i, j\}$ **do** /* *Recall that* $\hat{o}_t = \hat{o}^S_t/\hat{o}^C_t$ */
13: $\hat{o}^S_t \leftarrow$ numerator of $\hat{o}_t$.
14: $\mathcal{I}^S_t \leftarrow (\hat{o}^S_t - \sigma_S\sqrt{2}\,\text{erf}^{-1}(\beta), \hat{o}^S_t + \sigma_S\sqrt{2}\,\text{erf}^{-1}(\beta))$
15: $\hat{o}^C_t \leftarrow$ denominator of $\hat{o}_t$.
16: $\mathcal{I}^C_t \leftarrow (\hat{o}^C_t - \sigma_C\sqrt{2}\,\text{erf}^{-1}(\beta), \hat{o}^C_t + \sigma_C\sqrt{2}\,\text{erf}^{-1}(\beta))$
17: $\mathcal{I}^L \leftarrow \inf\{\mathcal{I}^S_i / \mathcal{I}^C_i - \mathcal{I}^S_j / \mathcal{I}^C_j\}$
18: $\mathcal{I}^U \leftarrow \sup\{\mathcal{I}^S_i / \mathcal{I}^C_i - \mathcal{I}^S_j / \mathcal{I}^C_j\}$
19: $\mathcal{I} \leftarrow (\mathcal{I}^L, \mathcal{I}^U)$
20: **return** $\mathcal{I}$

[9] erf^{-1} is the inverse function of the error function erf.

We next provide the guarantee for the obtained interval.

Lemma 4.1 *Given $\mathcal{I}^S$ and $\mathcal{I}^C$ as two β level confidence intervals of o_i^S and o_i^C separately, the derived interval $\mathcal{I}^A = \{x/y \mid x \in \mathcal{I}^S, y \in \mathcal{I}^C\}$ is a $2\beta - 1$ level confidence interval of o_i^S/o_i^C.*

Proof The following holds: $Pr[o_i^S/o_i^C \in \mathcal{I}^A] \geq Pr[o_i^S \in \mathcal{I}^S \wedge o_i^C \in \mathcal{I}^C] \geq 1 - (Pr[o_i^S \notin \mathcal{I}^S] + Pr[o_i^C \notin \mathcal{I}^C]) \geq 1 - ((1-\beta)+(1-\beta)) = 2\beta - 1$ The first inequality above is due to fact that the second event is sufficient for the first event: if two numbers are from $\mathcal{I}^S$ and $\mathcal{I}^C$, their division belongs to the set $\mathcal{I}^A$ by definition. The next inequality holds by applying the union bound. The third inequality is by definition. □

Furthermore, the difference $\hat{o}_i - \hat{o}_j$ is a subtraction between two ratios of two Gaussian variables, which can be expressed as an arithmetic combination of multiple Gaussian variables: $\hat{o}_i - \hat{o}_j = X_i/Y_i - X_j/Y_j$, where $X_t = N(o_t^S, \sigma_S^2)$ and $Y_t = N(o_t^C, \sigma_C^2)$ for $t \in \{i, j\}$. Similar to Lemma 4.1, we can derive the confidence interval for $\hat{o}_i - \hat{o}_j$ based on 4 partial confidence intervals of o_i^S, o_i^C, o_j^S, and o_j^C instead of 2. The confidence level we set for each partial confidence interval is $\beta = 1 - (1-\gamma)/4$ by applying union bound on the failure probability $1 - \gamma$ that one of the four variables is outside its interval. After we have 4 partial confidence intervals $\mathcal{I}_i^S$, $\mathcal{I}_i^C$, $\mathcal{I}_j^S$, and $\mathcal{I}_j^C$ for o_i^S, o_i^C, o_j^S, and o_j^C separately, similar to Lemma 4.1, we combine them together as $\mathcal{I}^A = \mathcal{I}_i^S / \mathcal{I}_i^C - \mathcal{I}_j^S / \mathcal{I}_j^C$ and derive the confidence interval for $o_i - o_j$ as $(\inf \mathcal{I}^A, \sup \mathcal{I}^A)$, which is guaranteed to be at confidence level γ. If 0 is included in either $\mathcal{I}_i^C$ or $\mathcal{I}_j^C$, we set the confidence interval to be $(\infty, -\infty)$ instead. Although there is no theoretical guarantee of the interval width, from two case studies in Sect. 6.2, we demonstrate narrow confidence intervals of AVG queries in practice, and observe no extreme case $(\infty, -\infty)$ in the experiments.

4.2 Influence function with low sensitivity

For **Theorem** 3.2, the goal is to design an influence function that has low sensitivity. Inspired by PrivBayes [106], we start by adapting a known influence function to our framework.

Our influence function of an explanation predicate with respect to a comparison user question is inspired by the Scorpion framework [104], where the user questions seek explanations for outliers in the results of a group-by aggregate query. Scorpion identifies predicates on the input that cause the outliers to disappear from the output. Given the group-by aggregation query shown in Fig. 3 and a group $\alpha_i \in \mathtt{dom}(A_{gb})$, recall from Sect. 2 that the true aggregate value for α_i is $o_i = agg(A_{agg}, g_i(D))$, where $g_i = \phi \wedge (A_{gb} = \alpha_i)$, i.e., $g_i(D)$ denotes the set of tuples that contribute to the group α_i.

Scorpion measures the influence of an explanation predicate p to some group α_i as the ratio between the change of output aggregate value and the change of group size:

$$\frac{agg(g_i(D)) - agg(g_i(\neg p(D)))}{|g_i(p(D))|} \tag{4.1}$$

Here $\neg p(D)$ denotes $D - p(D)$, i.e., the set of tuples in D that do not satisfy the predicate p. To adapt this influence function to DPXPLAIN, we make the following two changes.

- First, it should measure the influence w.r.t. the comparison from the user question $(\alpha_i, \alpha_j, >)$ instead of a single group.
  A natural extension is to change the target aggregate on g_i in the numerator in (4.1)
  to the difference between the aggregate values of two groups g_i, g_j
  before and after applying the explanation predicate p, and change the denominator
  as the maximum change in g_i or g_j when p is applied, which gives the following influence function:

$$\frac{\big(agg(g_i(D)) - agg(g_j(D))\big) - \big(agg(g_i(\neg p(D))) - agg(g_j(\neg p(D)))\big)}{\max(|g_i(p(D))|, |g_j(p(D))|)} \tag{4.2}$$

- Second and more importantly, in DPXPLAIN, we need to preserve DP when we use influence function to sort and rank multiple explanation predicates, or to release the influence and rank of an explanation predicate. Therefore, **we need to account for the sensitivity of the influence function**, which is determined by the worst-case change of influence when a tuple is added or removed from the database. If the predicate only selects a small number of tuples, the denominator in (4.2)
  is small and thus changing the denominator in (4.2) by one (when a tuple is added or removed) can result in a big change in the influence as illustrated in the following example, making
  (4.2) unsuitable for DPXPLAIN.

Example 4.1 [The Issue of the Influence Sensitivity] Suppose there are two groups α_i and α_j in D with 1000 tuples in each, aggregate function $agg = SUM$ on attribute A_{agg} with domain [0, 100], and the explanation predicate p

matches only 1 tuple from the group α_i with $A_{agg} = 100$ and no tuple from α_j. Suppose $agg(g_i(D)) = 20,000$, $agg(g_j(D)) = 10,000$, then $agg(g_i(\neg p(D))) = 19,900$ and $agg(g_j(\neg p(D))) = 10,000$. Therefore, from Equation (4.2), the influence of p is $((20,000-10,000)-(19,900-10,000))/\max\{1,0\} = 100$ on the original database D. However, suppose a new tuple that satisfies p and belongs to group α_i is added with $A_{agg} = 2$. Now the influence in Equation (4.2) becomes $((20,002-10,000)-(19,900-10,000))/\max\{2,0\} = 102/2 = 51$. While we added a tuple that contributes only 2 to the sum, it led to a change of 100-51 = 49 to the influence function due to the small denominator.

Therefore, we propose a new influence function that is inspired by Equation (4.2) but has lower sensitivity. Note that the denominator in Scorpion's influence function in Equation (4.2) acts as a normalizing factor, whose purpose is to penalize the explanation predicate that selects too many tuples, e.g., to prohibit the removal of the entire database by a dummy predicate. To have a similar normalizing factor with low sensitivity, we multiply the numerator in Equation (4.2) by $\frac{\min(|g_i(\neg p(D))|,|g_j(\neg p(D))|)}{\max(|g_i(D)|,|g_j(D)|)+1}$. From this new normalizing factor, the numerator captures the minimum of the number of tuples that are not removed from each group, and the denominator keeps the normalizing factor in the interval [0, 1] and does not change for different explanation predicates. Similar to Scorpion, if $p(D)$ constitutes a large fraction of D (e.g., if $p(D) = D$), then the normalizing factor is small, reducing the value of the influence. Also note that, unlike standard SQL query answering where only non-empty groups are shown in the results, in DP, all groups from the actual domain have to be considered, hence unlike Equation (4.1), $g_i(D)$, $g_j(D)$ could be zero, hence 1 is added in the denominator to avoid division by zero. When $agg = AVG$, we remove the constant denominator to boost the signal of the influence and keep the sensitivity low, which will be discussed in the sensitivity analysis after Proposition 4.1 and in Example 4.2.

Definition 4.1 [Influence of Explanation Predicates] Given a database D, a query q as shown in Fig. 3, and a user question $(\alpha_i, \alpha_j, >)$, the influence of an explanation predicate p is defined as $\text{INF}(p; (\alpha_i, \alpha_j, >), D)$, or simply $\text{INF}(p)$ when clear from context:

$$\begin{aligned}&\text{INF}(p) = \\ &\big(\big(agg(g_i(D)) - agg(g_j(D))\big) - \\ &\big(agg(g_i(\neg p(D))) - agg(g_j(\neg p(D)))\big)\big) \\ &\times \begin{cases} \frac{\min(|g_i(\neg p(D))|,|g_j(\neg p(D))|)}{\max(|g_i(D)|,|g_j(D)|)+1} & \text{for } agg \in \{COUNT, SUM\} \\ \min(|g_i(\neg p(D))|, |g_j(\neg p(D))|) & \text{for } agg = AVG \end{cases}\end{aligned}$$

The next proposition summarizes the sensitivity of eq. (4.3).

Proposition 4.1 *[Influence Function Sensitivity] Given an explanation predicate p and a user question with respect to a group-by query with aggregation agg, the following holds:*

1. *If $agg = COUNT$, the sensitivity of* $\text{INF}(p)$ *is 4.*
2. *If $agg = SUM$, the sensitivity of* $\text{INF}(p)$ *is* $4\ A^{max}_{agg}$.
3. *If $agg = AVG$, the sensitivity of* $\text{INF}(p)$ *is* $16\ A^{max}_{agg}$.

Proof We next prove each item in the proposition.
(1) COUNT. Recall the influence function definition:

$$\begin{aligned}\text{INF}(p; Q, D) = &\Big(\big(q(g_i(D)) - q(g_j(D))\big) \\ &- \big(q(g_i(\neg p(D))) - q(g_j(\neg p(D)))\big) \Big) \\ &\times \frac{\min\limits_{t\in\{i,j\}} |g_t(\neg p(D))|}{\max\limits_{t\in\{i,j\}} |g_t(D)| + 1}\end{aligned}$$

We interpret and consider the following equations or notations:

$$\begin{aligned} q(D) &= |D| \\ \phi_i &= (\phi \wedge A_{gb} = \alpha_i) \\ g_i(D) &= \phi_i(D) \\ g_i(p(D)) &= (\phi_i \wedge p)(D) \\ g_i(\neg p(D)) &= (\phi_i \wedge \neg p)(D) \\ num(D) &= \min_{t\in\{i,j\}} |g_t(\neg p(D))| \\ denom(D) &= \max_{t\in\{i,j\}} |g_t(D)| + 1 \\ h_i(D) &= q((\phi_i \wedge p)(D)) num(D)/denom(D) \end{aligned}$$

Since q is a counting query, we have $q(g_i(D)) - q(g_i(\neg p(D))) = q(g_i(p(D)))$, and by replacing $g_i(p(D))$ with $(\phi_i \wedge p)(D)$ we have $q(g_i(D)) - q(g_i(\neg p(D))) = q((\phi_i \wedge p)(D))$.

By further replacing the last numerator and denominator in the influence function with $num(D)$ and $denom(D)$, we have $\text{INF}(p; Q, D) = h_i(D) - h_j(D)$.

We prove the sensitivity bound by the following inequality chains.

$$\Delta_{\text{INF}} = max_{D\approx D'} |\text{INF}(p; Q_{CNT}, D) - \text{INF}(p; Q_{CNT}, D'| \tag{4.3}$$

We first replace INF according to $\text{INF}(p; Q, D) = h_i(D) - h_j(D)$, and then apply Lemma 2.2

to bound the sensitivity by the sum of sensitivities of h_i and h_j.

$$\leq \textstyle\sum_{t\in\{i,j\}} \max_{|D'|=|D|+1} |h_t(D') - h_t(D)| \tag{4.4}$$

The second inequality is by Lemma 2.5,

since f is a non-negative query with sensitivity 1 and g is a monotonic positive and positive query with sensitivity 1.

$$\leq \sum_{t\in\{i,j\}} \frac{2|(\phi_t \wedge p)(D)|+num(D)+1}{denom(D)} \Delta_q \tag{4.5}$$

The next equality is by replacing the variables. Since q is a counting query, it has sensitivity $\Delta_q = 1$.

$$= \sum_{t\in\{i,j\}} \frac{2|(\phi_t \wedge p)(D)|+ \min\limits_{s\in\{i,j\}} |(\phi_s \wedge \neg p)(D)|+1}{\max_{s\in\{i,j\}}|g_s(D)|+1} \tag{4.6}$$

The third inequality is by the property of min and max, since $\min\limits_{s\in\{i,j\}} |(\phi_s \wedge \neg p)(D)| \leq |(\phi_t \wedge \neg p)(D)|$ and $\max_{s\in\{i,j\}}|g_s(D)| \geq |g_t(D)|$.

$$\leq \sum_{t\in\{i,j\}} \frac{|(\phi_t \wedge p)(D)|+|(\phi_t \wedge p)(D)|+|(\phi_t \wedge \neg p)(D)|+1}{|g_t(D)|+1} \tag{4.7}$$

The next equality is due to that $\phi_t = (\phi_t \wedge p) \vee (\phi_t \wedge \neg p)$.

$$= \sum_{t\in\{i,j\}} \frac{|(\phi_t \wedge p)(D)|+(|\phi_t(D)|+1)}{|g_t(D)|+1} \tag{4.8}$$

The fourth inequality is due to that $|(\phi_t \wedge p)(D)| \leq |\phi_t(D)| = |g_t(D)| \leq |g_t(D)| + 1$.

$$\leq \sum_{t\in\{i,j\}} \frac{(|g_t(D)|+1)+(|g_t(D)|+1)}{|g_t(D)|+1} \tag{4.9}$$

$$\leq 4 \tag{4.10}$$

(2) SUM. Similar to the proof of the sensitivity of CNT influence, but with $\Delta_q = A^{max}_{agg}$, which should be replaced at Equation (4.5).

(3) AVG.

$$\begin{aligned}&\textsc{Inf}(p; Q_{AVG}, D)\\ &= \Big(\big(\frac{SUM(\phi_i(D), A_{agg})}{|\phi_i(D)|} - \frac{SUM(\phi_j(D), A_{agg})}{|\phi_j(D)|}\big) - \\ &\quad \big(\frac{SUM((\phi_i \wedge \neg p)(D), A_{agg})}{|(\phi_i \wedge \neg p)(D)|} - \frac{SUM((\phi_j \wedge \neg p)(D), A_{agg})}{|(\phi_j \wedge \neg p)(D)|}\big)\Big)\\ &\quad \min_{t\in\{i,j\}} |(\phi_t \wedge \neg p)(D)|\end{aligned}$$

Now we consider decompose this query into four parts (for example, $\frac{SUM(\phi_i(D),A_{agg})}{|\phi_i(D)|} \min\limits_{t\in\{i,j\}} |(\phi_t \wedge \neg p)(D)|$ as one part), and analyze the sensitivity for each part and finally sum up. Consider query q as summing up A_{agg} with sensitivity $\Delta_q = A^{max}_{agg}$. By Lemma 2.5, we can show that the sensitivity of each part is 4 Δ_q. Together, the total sensitivity is bounded by 16 Δ_q. □

Intuitively, the sensitivity of $\textsc{Inf}(p)$ is low compared to its value. When $agg = COUNT$, $\textsc{Inf}(p)$ is $O(n)$ and $\Delta_{\textsc{Inf}}$ is $O(1)$, where n is the size of database. When $agg \in \{SUM, AVG\}$, $\textsc{Inf}(p)$ is $O(nA^{max}_{agg})$ and $\Delta_{\textsc{Inf}}$ is $O(A^{max}_{agg})$. Therefore, the sensitivity of influence $\Delta_{\textsc{Inf}}$ is low compared to the influence itself. However, as the example below shows, if we define the influence function for AVG the same way as $COUNT$ or SUM, both $\textsc{Inf}(p)$ and $\Delta_{\textsc{Inf}}$ will become $O(A^{max}_{agg})$, which makes the sensitivity (relatively) large.

Example 4.2 [The Issue with AVG Influence.] Consider an AVG group-by query where the domain of the aggregate attribute is [0, 100], and an explanation predicate p such that for group α_i we have 2 tuples with $AVG(g_i(D)) = 100/2 = 50$, $AVG(g_i(\neg p(D))) = 0/1 = 0$, and for group α_j we have two tuples with $AVG(g_j(D)) = 100/2 = 50$ and $AVG(g_j(\neg p(D))) = 100/2 = 50$. Suppose we define the influence function for AVG the same way as $COUNT$ or SUM, therefore the influence of p in Equation (4.3) is $\textsc{Inf}(p) = ((50-50)-(0-50))(\min(1,2)/(\max(2,2)+1) = 50/3$. However, suppose we remove the single tuple from g_i, so $|g_i(\neg p(D))|$ becomes 0, now the influence in Equation (4.3) (for COUNT/SUM) becomes 0. Note that a single removal of a tuple completely changes the influence to 0, and this change is equal to the influence itself, which is relatively large and therefore is not a good choice for AVG.

Note that the user question "why $(\alpha_i, \alpha_j, >)$" is asked based on the noisy results $\hat{o}_i > \hat{o_j}$, while the influence function uses the true results, i.e., even if $o_i \leq o_j$, we still consider $agg(g_i(D)) - agg(g_j(D))$ in $\textsc{Inf}(p)$. Hence $\textsc{Inf}(p)$ can be positive or negative and removing tuples satisfying p can make the gap smaller or larger.

We demonstrate this property in the example below without the normalizing factor in the function.

Example 4.3 Start with a database with three binary attributes: A, B, C and two tuples: (0, 0, 0), (1, 0, 1). Consider an $agg = COUNT$ query with group by on A, so we have $agg(g_0(D)) = 1$ and $agg(g_1(D)) = 1$ for two groups $A = 0$ and $A = 1$.

Consider three explanation predicates for the user question $(\alpha_0, \alpha_1, >)$ (note that the noisy values can be different from the true values):

$p_1 : B = 0$, $p_2 : B = 0 \wedge C = 0$ and $p_3 : B = 0 \wedge C = 1$, which satisfy

$p_2 \Rightarrow p_1$ and $p_3 \Rightarrow p_1$. However, while $\textsc{Inf}(p_1) = 0$, we have $\textsc{Inf}(p_2) = 1$ and $\textsc{Inf}(p_3) = -1$, i.e., $\textsc{Inf}(p_3) < \textsc{Inf}(p_1) < \textsc{Inf}(p_2)$.

Note that $denom(D)$ ($num(D)$) denotes the denominator (numerator) in the normalizing factor of $\textsc{Inf}$ designed for $COUNT$ queries. The value of $denom(D)$ is 2 since the size of both groups is 1 and the value of $num(D)$ for p_1 for example, is 0 since both tuples have $B = 0$ and thus $|g_t(\neg p_1(D))| = |g_t(\emptyset)|$ for $t = 1, 2$.

4.3 Private top-k explanations

In this section, we discuss the computation of the top-k explanation predicates and the confidence intervals of influences and ranks.

4.3.1 Problem 3: private top-k explanation predicates

The goal is to find with DP the top-k explanation predicates from a set of explanation predicates $\mathcal{P}$ in terms of their (true) influences $\text{INF}(p)$, which is the first step in Phase-3 of DPXPLAIN (Fig. 1). Note that simply choosing the *true* top-k explanation predicates in terms of their $\text{INF}(p)$ is not differentially private.

In DPXPLAIN, we adopt the **One-shot Top-k mechanism** [37, 38] to privately select the top-k.

We now present the One-shot Top-k mechanism (described in Algorithm 2) which is based on the exponential mechanism [42]. Given a score function $\text{s} : \mathcal{P} \to \mathcal{R}$ that maps an explanation predicate p to a number, the exponential mechanism (EM) [42] randomly samples p from $\mathcal{P}$ with probability proportional to $exp(\epsilon\, \text{s}(p)/(2\Delta_\text{s}))$ with some privacy parameter ϵ and satisfies $(\epsilon^2/8)$-zCDP [17, 32, 38, 83]. The higher the score is, the more possible that an explanation predicate is selected. In DPXPLAIN, we use the influence function as the score function.

We denote the exponential mechanism as $\mathcal{M}_E$. To find 'top-k' explanation predicate satisfying DP, we can first apply $\mathcal{M}_E$ to find one explanation predicate, remove it from the entire explanation predicate space, and then apply $\mathcal{M}_E$ again until k explanation predicates are found. It was shown by previous work [37, 38] that this process is identical to adding i.i.d. Gumbel noise[10] to each score and releasing the top-k predicates by the noisy scores (i.e., there is no need to remove predicates after sampling). We, therefore, use this result to devise a similar solution that is presented in Algorithm 2. In line 1, we set the noise scale. In lines 2–4, we randomly sample Gumbel noise with scale σ and add it to the influence of each explanation predicate from the space $\mathcal{P}$. In line 5, we sort the noisy scores in the descending order, and in line 6, we find the top-k explanation predicates by their noisy scores. This algorithm satisfies ρ_{Topk}-zCDP (as formally stated in Proposition 4.2), and can be applied to questions on SUM, COUNT, or AVG queries, with different score functions and sensitivity values for different aggregates.

Since this algorithm iterates over each explanation predicate, the time complexity is proportional to the size of the explanation predicate set $\mathcal{P}$. By Definition 3.2, this number is $O(\binom{m}{l}N^l)$, where N is the maximum domain size of an attribute, l is the number of conjuncts in the explanation predicate and m is the number of attributes. In our experiments (Sect. 6), we fix $l = 1$ and use all the singleton predicates as the set $\mathcal{P}$, so its size is linear in the number of attributes.

[10] For a Gumbel noise $Z \sim Gumbel(\sigma)$, its CDF is $Pr[Z \le z] = \exp(-\exp(-z/\sigma))$.

Algorithm 2 Noisy Top-k Predicates

Require: An influence function INF with sensitivity Δ_{INF}, a set of explanation predicates $\mathcal{P}$, a privacy parameter ρ_{Topk} and a size parameter k.
Ensure: Top-k explanation predicates.
1: $\sigma \leftarrow 2\Delta_{\text{INF}}\sqrt{k/(8\rho_{Topk})}$
2: **for** $u \leftarrow 1 \ldots |\mathcal{P}|$ **do**
3: $\quad s_u \leftarrow \text{INF}(p_u) + Gumbel(\sigma)$
4: Sort $s_1 \ldots s_{|\mathcal{P}|}$ in the descending order.
5: Let $p_1, p_2, \ldots, p_k$ be the top-k elements in the list.
6: **return** $p_1, p_2, \ldots, p_k$

Proposition 4.2 *Given an influence function* INF *with sensitivity* Δ_{INF}, *a set of explanation predicates* $\mathcal{P}$, *a privacy parameter* ρ_{Topk} *and a size parameter* k, *the following holds:*

1. *One-shot Top-k mechanism finds k explanation predicates while satisfying* ρ_{Topk}*-zCDP.*
2. *Denote by* $OPT^{(i)}$ *the* i*-th highest (true) influence, and by* $\mathcal{M}^{(i)}$ *the* i*-th explanation predicate selected by the One-shot Top-k mechanism. For* $\forall t$ *and* $\forall i \in \{1, 2, \ldots, k\}$, *we have*

$$Pr[\text{INF}(\mathcal{M}^{(i)}) \le OPT^{(i)} - \frac{2\Delta_{\text{INF}}}{\sqrt{8\rho_{Topk}/k}}(\ln(|\mathcal{P}|) + t)] \le e^{-t} \tag{4.11}$$

Proof *(1) Differential Privacy.* it is equivalent to iteratively applying k exponential mechanisms [42] that satisfies $\epsilon^2/8$-zCDP [17, 32, 38, 83] for each, where $\epsilon = \sqrt{8\rho_{Topk}/k}$ [37, 38], therefore in total it satisfies $(k\epsilon^2/8)$-zCDP which is also ρ_{Topk}-zCDP.

(2) Utility Bound. It is extended from the utility theorem of EM in Thm 3.11 of [42], which states that

$$Pr\left[\text{INF}(\mathcal{M}^{(1)}) \le OPT^{(1)} - \frac{2\Delta_{\text{INF}}}{\epsilon}(\ln(|\mathcal{P}|) + t)\right] \le e^{-t}$$

where $\epsilon = \sqrt{8\rho_{Topk}/k}$. To extend from $i = 1$ to $\forall i \in \{1, 2, \ldots, k\}$, we follow the original proof:

$$Pr[\text{INF}(\mathcal{M}^{(i)}) \le c] \le \frac{|\mathcal{P}|\exp(\epsilon c/(2\Delta_{\text{INF}}))}{\exp(\epsilon\, OPT^{(i)}/(2\Delta_{\text{INF}}))}$$

by giving a upper bound and lower bound of the numerator and denominator. Replacing c with the appropriate value will give this theorem. □

Example 4.4 Reconsider the user question in Fig. 1c. For this question, we have in total 103 explanation predicates as the set of explanation predicates. The privacy budget $\rho_{Topk} = 0.05$, the size parameter $k = 5$, and the sensitivity $\Delta_{\text{INF}} = 16$. For each of the explanation predicate, we add a Gumbel noise with scale $\sigma = 113$ to their influences. For example, for

the predicates shown in Fig. 1d, their noisy influences are 990, 670, 645, 475, 440, which are the highest 5 among all the noisy influences. The true influences for these five ones are 547, 501, 555, 434, 118. To see how close it is to the true top-5, we compare their true influences with the true highest five influences: 555, 547, 501, 434, 252, which shows the corresponding differences in terms of influence are 8, 46, 54, 0, 134. By Equation (4.11), the probability that such difference is beyond 864 is at most 5% for each explanation predicate. Finally, we sort explanation predicates by their noisy influences and report the top-k. These k predicates will be reordered as discussed in Sect. 4.4.

4.3.2 Theorem 3.4: Private Confidence Interval of Influence

The goal is to generate a confidence interval of influence $\text{INF}(p)$ (Definition 4.1) of each explanation predicate $\text{INF}(p_1)$, $\text{INF}(p_2)$, ..., $\text{INF}(p_k)$ from the selected top-k (Sect. 4.3.1). For each $\text{INF}(p_i)$, we apply the Gaussian mechanism (Theorem 2.1) with privacy budget ρ_{Influ}/k to release a noisy influence $\widehat{\text{INF}}_i$ with noise scale $\sigma = \Delta_{\text{INF}}/\sqrt{2\rho_{Influ}/k}$. The sensitivity term Δ_{INF} is determined by Proposition 4.1. Following the standard properties of Gaussian distribution, for each $\text{INF}(p_i)$, we set the confidence interval by a center c as $\widehat{\text{INF}}_i$ and a margin m as $\sqrt{2}\sigma\,\text{erf}^{-1}(\gamma)$, or (c-m, c+m), as a γ level confidence interval of $\text{INF}(p_i)$ [102]. Together, it satisfies ρ_{Influ}-zCDP according to the composition property by Proposition 2.1.

The pseudo code of the procedure is presented in Algorithm 3. It takes a privacy budget ρ_{Influ} as input. In Line 2 we divide the privacy budget ρ_{Influ} into k equal portions for each explanation predicate p_u for $u \in \{1, \ldots, k\}$. In Line 3, we calibrate the noise scale according to the sensitivity of the influence function. In Line 9, we add a Gaussian noise to the influence $\text{INF}(p_u)$ of explanation predicate p_u, and finally in Lines 10 and 11, we derive the confidence interval based on the Gaussian property [102].

4.3.3 Theorem 3.5: Private Confidence Interval of Rank

The goal is to find the confidence interval of the rank of each explanation predicate from the selected top-k (Sect. 4.3.1). We denote $\text{rank}(p)$ as the rank of $p \in \mathcal{P}$ by the natural ordering of the predicates imposed by their (true) influences according to the influence function INF, and denote $\text{rank}^{-1}(t)$ (for an integer $1 \le t \le |\mathcal{P}|$) as the predicate ranked in the t-th place according to INF. One trivial example of a confidence interval of rank is $[1, |\mathcal{P}|]$, which has no privacy loss and always includes the true rank.

Unlike the sensitivity of the influence function, the sensitivity of $\text{rank}(p)$ is high, since adding one tuple could possibly change the highest influence to be the lowest and vice versa. Fortunately, we can employ a critical observation about rank and influence.

Algorithm 3 Compute Confidence Interval of Influence

Require: An influence function INF with respect to the question $(\alpha_i, >, \alpha_j)$, k explanation predicates $p_1, p_2, \ldots, p_k$, a private database D, a privacy budget ρ_{Influ}, and a confidence level γ.
Ensure: A list of γ-level confidence intervals of the influence $\text{INF}(p_u)/(\hat{o}_i - \hat{o}_j)$ for $u \in \{1, 2, \ldots, k\}$.
1: **for** $u \in \{1, 2, \ldots, k\}$ **do**
2: $\quad\rho \leftarrow \rho_{Influ}/k$
3: $\quad$**if** agg $= COUNT$ **then**
4: $\quad\quad\sigma \leftarrow 4/\sqrt{2\rho}$
5: $\quad$**else if** agg $= SUM$ **then**
6: $\quad\quad\sigma \leftarrow 4A^{max}_{agg}/\sqrt{2\rho}$
7: $\quad$**else if** agg $= AVG$ **then**
8: $\quad\quad\sigma \leftarrow 16A^{max}_{agg}/\sqrt{2\rho}$
9: $\quad\hat{\text{INF}} \leftarrow \text{INF}(p_u) + N(0, \sigma^2)$
10: $\quad\mathcal{I}^L_u \leftarrow \hat{\text{INF}} - \sqrt{2}\sigma\,\text{erf}^{-1}(\gamma)$
11: $\quad\mathcal{I}^U_u \leftarrow \hat{\text{INF}} + \sqrt{2}\sigma\,\text{erf}^{-1}(\gamma)$
12: $\quad\mathcal{I}_u \leftarrow (\mathcal{I}^L_u, \mathcal{I}^U_u)$
13: **return** $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k$

Proposition 4.3 *Given a set of explanation predicates $\mathcal{P}$, an influence function* INF *with global sensitivity Δ_{INF}, and an integer $1 \le t \le |\mathcal{P}|$,* $\text{INF}(\text{rank}^{-1}(t))$ *has sensitivity Δ_{INF}.*

The intuition behind this proof is that, fixing an explanation predicate $p = \text{rank}^{-1}(t)$, for a neighboring database, if its influence is increased, its rank will be moved to the top which pushes down other explanation predicates with lower influences, so the influence at the rank t in the neighboring database is still low. For a target explanation predicate p, since both $\text{INF}(p)$ and $\text{INF}(\text{rank}^{-1}(t))$ have low sensitivity as Δ_{INF}, intuitively we can check whether t is close to the rank of p by checking whether their influences $\text{INF}(p)$ and $\text{INF}(\text{rank}^{-1}(t))$ are close by adding a little noise to satisfy DP. Given this observation, we devise a binary-search-based strategy to find the confidence interval of rank.

Lemma 4.2 *Given a set of predicates $\mathcal{P}$, an influence function* INF *with global sensitivity Δ_{INF} and a number t, then the function $s(D) = \text{INF}(p; D) - \text{INF}(\text{rank}^{-1}(t; D, \mathcal{P}, \text{INF}); D)$ has global sensitivity $2\Delta_{\text{INF}}$.*

Proof The sensitivity of INF is Δ_{INF} by definition and the sensitivity of $\text{INF}(\text{rank}^{-1}(t; D, \mathcal{P}, \text{INF})$ is Δ_{INF} by Proposition 4.3. By Lemma 2.2, together it has sensitivity $2\Delta_{\text{INF}}$. □

Proof of Proposition 4.3 Drop $\mathcal{P}$ and INF from $\text{rank}^{-1}(t; D, \mathcal{P}, \text{INF})$ for simplicity. Next we show that for any two neighboring datasets $D' \sim D$, we have $|\text{INF}(\text{rank}^{-1}(t; D'); D') - \text{INF}(\text{rank}^{-1}(t; D); D)| \le \Delta_{\text{INF}}$, which is equivalent to showing $-\Delta_{\text{INF}} \le \text{INF}(\text{rank}^{-1}(t; D'); D') - \text{INF}(\text{rank}^{-1}(t; D); D) \le \Delta_{\text{INF}}$.

Case 1 lower bound. This is to show that for any $D' \approx D$, we have $\text{INF}(\text{rank}^{-1}(t; D'); D') - \text{INF}(\text{rank}^{-1}(t; D); D) \geq -\Delta_{\text{INF}}$.

By the definition of global sensitivity, for any explanation predicate p, we have $|\text{INF}(p; D') - \text{INF}(p; D)| \leq \Delta_{\text{INF}}$, and therefore $\text{INF}(p; D') \geq \text{INF}(p; D) - \Delta_{\text{INF}}$. By replacing p with $\text{rank}^{-1}(j; D)$ for some j, we have $\text{INF}(\text{rank}^{-1}(j; D); D') \geq \text{INF}(\text{rank}^{-1}(j; D); D) - \Delta_{\text{INF}}$. For any $j \leq t$, by the property of ranking, we have $\text{INF}(\text{rank}^{-1}(j; D); D) \geq \text{INF}(\text{rank}^{-1}(t; D); D)$. Together, for any $j \leq t$, we have $\text{INF}(\text{rank}^{-1}(j; D); D') \geq \text{INF}(\text{rank}^{-1}(j; D); D) - \Delta_{\text{INF}} \geq \text{INF}(\text{rank}^{-1}(t; D); D) - \Delta_{\text{INF}}$. This means there are at least t elements in D' such that their scores are above $\text{INF}(\text{rank}^{-1}(t; D); D) - \Delta_{\text{INF}}$, which implies for the t-th largest score in D' we have $\text{INF}(\text{rank}^{-1}(t; D'); D') \geq \text{INF}(\text{rank}^{-1}(t; D); D) - \Delta_{\text{INF}}$.

Case 2 upper bound. This is to show that for any $D' \approx D$, we have $\text{INF}(\text{rank}^{-1}(t; D'); D') - \text{INF}(\text{rank}^{-1}(t; D); D) \leq \Delta_{\text{INF}}$.

By the definition of global sensitivity, for any explanation predicate p, we have $|\text{INF}(p; D') - \text{INF}(p; D)| \leq \Delta_{\text{INF}}$, and therefore $\text{INF}(p; D') \leq \text{INF}(p; D) + \Delta_{\text{INF}}$. By replacing p with $\text{rank}^{-1}(j; D)$ for some j, we have $\text{INF}(\text{rank}^{-1}(j; D); D') \leq \text{INF}(\text{rank}^{-1}(j; D); D) + \Delta_{\text{INF}}$. For any $j \geq t$, by the property of ranking, we have $\text{INF}(\text{rank}^{-1}(j; D); D) \leq \text{INF}(\text{rank}^{-1}(t; D); D)$. Together, for any $j \geq t$, we have $\text{INF}(\text{rank}^{-1}(j; D); D') \leq \text{INF}(\text{rank}^{-1}(j; D); D) + \Delta_{\text{INF}} \leq \text{INF}(\text{rank}^{-1}(t; D); D) + \Delta_{\text{INF}}$. This means there are at most $t-1$ elements in D' such that their scores can be above $\text{INF}(\text{rank}^{-1}(t; D); D) + \Delta_{\text{INF}}$, which implies for the t-th largest score in D' we have $\text{INF}(\text{rank}^{-1}(t; D'); D') \leq \text{INF}(\text{rank}^{-1}(t; D); D) + \Delta_{\text{INF}}$.

Recall that we have bounded the sensitivity of the influence function (Δ_{INF}) in Proposition 4.1. Therefore, the sensitivity of $\text{INF}(\text{rank}^{-1}(t))$ has the same exact bounds which depend on the query aggregate function.

Noisy binary search mechanism We decompose the problem into finding two bounds of the confidence interval separately by a subroutine $\text{RANKBOUND}(p, \rho, \beta, dir)$ that guarantees that it will find a lower ($dir = -1$) or upper ($dir = +1$) bound of rank with probability β for the explanation predicate p using privacy budget ρ. We divide the privacy budget ρ into two parts by a parameter $\eta \in (0, 1)$ and return ($\text{RANKBOUND}(p_u, \eta\rho, \beta, -1)$, $\text{RANKBOUND}(p_u, (1-\eta)\rho, \beta, +1)$) as the confidence interval of rank for each predicate p_u for $u \in \{1, \ldots, k\}$, where $\rho = \rho_{Rank}/k$ to divide the total privacy budget equally, and $\beta = (\gamma + 1)/2$ to ensure a confidence of γ.

The subroutine $\text{RANKBOUND}(p, \rho, \beta, dir)$ works as follows. It is a noisy binary search with at most $N = \lceil \log_2 |\mathcal{P}| \rceil$ loops. We initialize the search pointers $t_{low} = 1$ and $t_{high} = |\mathcal{P}|$ as the two ends of possible ranks. Within each loop, we check the difference of influences at $t = \lfloor (t_{high} + t_{low})/2 \rfloor$ by adding a Gaussian noise:

$$\hat{s} = \text{INF}(p) - \text{INF}(\text{rank}^{-1}(t)) + \mathcal{N}(0, \sigma^2) \tag{4.12}$$

The noise scale is set as $\sigma = (2\Delta_{\text{INF}})/\sqrt{2(\rho/N)}$ to satisfy ρ/N-zCDP. Instead of comparing the noisy difference $\hat{s}$ with 0 to check whether t is a close bound of $\text{rank}(p)$, we compare it with the following slack constant ξ so that w.h.p. t is a true bound of $\text{rank}(p)$.

$$\xi = \sigma\sqrt{2\ln(N/(1-\beta))} \times dir \tag{4.13}$$

We update the binary search pointers by the comparison: if $\hat{s} \geq \xi$, we set $t_{high} = \max\{t-1, 1\}$, otherwise $t_{low} = \min\{t+1, |\mathcal{P}|\}$. The binary search stops when $t_{high} \leq t_{low}$ and returns t_{high} as the rank bound.

We next describe the noisy binary search mechanism in more detail, as shown by Algorithm 4. In line 1, RANKBOUND takes four parameters: an explanation predicate p, a privacy budget ρ, a sub confidence level β and a direction $dir \in \{-1, +1\}$. It guarantees that it will find a lower ($dir = -1$) or upper ($dir = +1$) bound of rank with confidence β for the explanation predicate p using privacy budget ρ. In line 2, we set the maximum depth N of the binary search. In line 3, we set the noise scale σ_{-1} or σ_{+1}, which depends on the sensitivity of $\text{INF}(p) - \text{INF}(\text{rank}^{-1}(t))$ (in line 8), which is $2\Delta_{\text{INF}}$; and the number of Gaussian mechanisms used in the binary search, which is N. In line 4, we set the margin ξ_{+1} or ξ_{-1}, which will be discussed in line 9. In line 5, we initialize the binary search by setting two pointers, t_{low} and t_{high}, as the first and last rank. In lines 6–10 there is a while loop for the binary search. In line 7, we pick a rank that is at the middle of two pointers. In line 8, we add a Gaussian noise with scale σ to the difference between the influence of the target explanation predicate p and the influence of the explanation predicate that has rank t. From line 9 to 10 we update one of the pointer according to the relationship between the noisy difference and the margin ξ_{dir}. If we are trying to find a rank upper bound ($dir = +1$), we want the binary search to find the rank such that the difference (without noise) is above zero. Due to the noise injected, even if the noisy difference is above zero, the true difference could be negative. To secure the goal with high probability, we requires the noisy difference to be above a margin ξ_{dir}, as shown in line 9. In this case, we narrow down the search space by moving t_{high} to $\max\{t-1, 1\}$. The strategy is similar when we are looking for a rank lower bound ($dir = -1$).

Now, we describe the usage of the sub-routine RANKBOUND. We repeat the following for each explanation predicate. In line 13, we allocate an even portion from the total privacy budget ρ_{Rank}, and set the sub confidence level to $\beta = (\gamma + 1)/2$ so the final confidence interval has con-

fidence level $2\beta - 1 = \gamma$ by the rule of union bound. In lines 14, we divide the privacy budget ρ, and make two calls to the sub-routine RANKBOUND to find a rank upper bound and a rank lower bound for the explanation predicate p_u, and finally merge them into a single confidence interval. We spend more budget for the rank upper bank since this is more important in the explanation.

Algorithm 4 Compute Confidence Interval of Rank

Require: A dataset D, a predicate space $\mathcal{P}$, an influence function INF with sensitivity Δ_{INF}, explanation predicates $p_1, p_2, \ldots, p_k$, a confidence level γ, and a privacy parameter ρ_{Rank}.
Ensure: A list of γ-level confidence intervals of the influence rank$(p_u; D, \mathcal{P}, \text{INF})$ for $u \in \{1, 2, \ldots, k\}$.
1: **function** RANKBOUND(p, ρ, β, dir)
2: $N \leftarrow \lceil \log_2 |\mathcal{P}| \rceil$
3: $\sigma_{dir} \leftarrow (2\Delta_{\text{INF}})/\sqrt{2(\rho/N)}$
4: $\xi_{dir} \leftarrow \sigma_{dir}\sqrt{2\ln(N/(1-\beta))} \times dir$
5: $t_{low}, t_{high} \leftarrow 1, |\mathcal{P}|$
6: **while** $t_{high} \geq t_{low}$ **do**
7: $t \leftarrow \lfloor \frac{t_{high}+t_{low}}{2} \rfloor$
8: $\hat{s} \leftarrow \text{INF}(p) - \text{INF}(\text{rank}^{-1}(t)) + \mathcal{N}(0, \sigma^2)$
9: **if** $\hat{s} \geq \xi_{dir}$ **then** $t_{high} \leftarrow \max\{t-1, 1\}$
10: **else** $t_{low} \leftarrow \min\{t+1, |\mathcal{P}|\}$
11: **return** t_{high}
12: **for** $u \leftarrow 1, 2, \ldots, k$ **do**
13: $\rho, \beta \leftarrow \rho_{Rank}/k, (\gamma+1)/2$
14: $\mathcal{I}_u \leftarrow$ (RANKBOUND$(p_u, 0.1\rho, \beta, -1)$, RANKBOUND$(p_u, 0.9\rho, \beta, +1)$)
15: **return** $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k$

Example 4.5 Fig. 4 shows an example of RANKBOUND for finding the upper bound of the confidence interval for rank(p) for some explanation predicate p (with true rank 3 shown in red). The upper part of the figure shows the influences of all the explanation predicates in descending order, and the lower part shows the status of the binary search pointers in each loop. The search contains three loops starting from $t_{low} = 1$ and $t_{high} = 15$. Within each loop, to illustrate the idea, it is equivalent to adding a Gaussian noise to INF$(\text{rank}^{-1}(t))$, which is shown as a blue circle, compare it with INF$(p) - \xi$, which is shown as a dashed line, and update the pointers accordingly. For example, in loop 1, the blue circle 1 is in the green region, so the pointer t_{high} is moved from 15 to 7 (shown in the lower part). Finally, it breaks at $t_{low} = t_{high} = 5$.

We now show that **noisy binary search mechanism** satisfies the privacy requirement, and outputs valid confidence intervals. In Sect. 6, we show that the interval width is empirically small.

Theorem 4.1 *Given a database D, a predicate space $\mathcal{P}$, an influence function* INF *with sensitivity Δ_{INF}, explanation predicates $p_1, p_2, \ldots, p_k$, a confidence level γ, and a privacy parameter ρ_{Rank}, noisy binary search mechanism returns confidence intervals $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k$ such that*

1. *Noisy binary search mechanism satisfies ρ_{Rank}-zCDP.*
2. *For $\forall u \in [1, k]$, $\mathcal{I}_u$ is a γ level confidence interval of* rank(p_u).

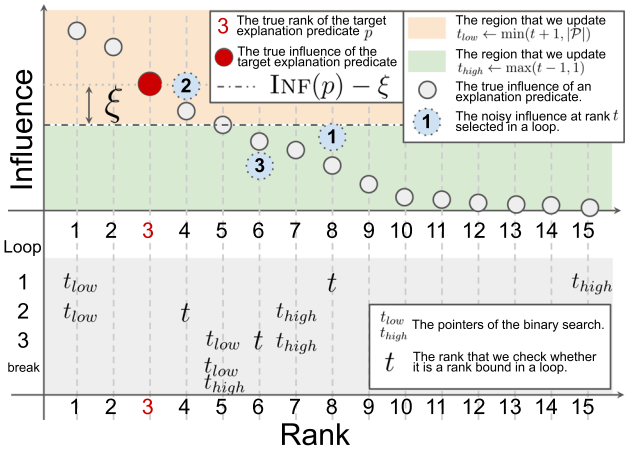


Fig. 4 Execution of RANKBOUND for finding the upper bound of the confidence interval of rank for the predicate p (with true rank 3, in red) in a toy example

Proof *(1) Differential Privacy* We first show that Algorithm 4 satisfies ρ_{Rank}-zCDP.

The main structure of Algorithm 4 is a for-loop of k explanation predicates from line 12 to 14, and within each for-loop, we first prepare the parameters at line 13 and 13, make two calls to the sub-routine RANKBOUND and construct the confidence interval by the sub-routine outputs. We first show below that each call to the sub-routine RANKBOUND with parameters (p, ρ, β, dir) satisfies ρ-zCDP. Given this is true, we then show that our two calls RANKBOUND$(p_u, 0.1\rho, \beta, -1)$ and RANKBOUND$(p_u, 0.9\rho, \beta, +1)$ at Line 14 satisfies 0.1ρ-zCDP and 0.9ρ-zCDP, which together satisfies ρ-zCDP by the composition rule (Proposition 2.1). By line 13, we set $\rho = \rho_{Rank}/k$, therefore each loop satisfies (ρ_{Rank}/k)-zCDP, and after in total k loops, it satisfies ρ_{Rank}-zCDP by the composition rule (Proposition 2.1).

Next we show that RANKBOUND(p, ρ, β, dir) satisfies ρ-zCDP, from line 1 to 11. We first prepare some parameters at the start of the sub-routine, which does not touch the data, and then enters a while loop with at most $N = \lceil \log_2 |\mathcal{P}| \rceil$ loops. Denote $s = \text{INF}(p) - \text{INF}(\text{rank}^{-1}(t))$. Within each loop, we add a Gaussian noise to a secret s at line 8. The value of s touches the sensitive data, but by adding a Gaussian noise to s, the release of $\hat{s}$ satisfies zCDP. By Theorem 2.1, with noise scale σ, it satisfies $(\Delta_q^2)/2\sigma^2$-zCDP where Δ_q is the sensitivity of the function that we want to release. Since we

set $\sigma = (2\Delta_{\text{INF}})/\sqrt{2(\rho/N)}$ at line 3 and the sensitivity of s is $2\Delta_{\text{INF}}$ by Lemma 4.2, it satisfies (ρ/N)-zCDP. Since we have at most N noisy releases of S using the Gaussian mechanism, by composition rule (Proposition 2.1), the entire while loop satisfies ρ-zCDP, and so is the sub-routine.

(2) Confidence Interval Now we show that the confidence interval outputted by RANKBOUND(p, ρ, β, dir), from line 1 to 11, is a γ-level confidence interval.

The sub-routine RANKBOUND with direction 'upper' is a mirror to RANKBOUND with direction 'lower'. We first show that RANKBOUND returns a bound in either upper or lower case such that it is a true bound with probability $\beta = \frac{\gamma+1}{2}$, therefore the target rank is within two bounds with probability γ. We give the proof for the case when direction is upper for the sub-algorithm RANKBOUND, and skip the proof for the case when direction is lower due to the similarity.

The sub-routine RANKBOUND is a random binary search algorithm with in total N loops.

To ensure that the final t_{high} is a rank bound, one sufficient condition is that t_{high} is always an upper bound of rank during all the loops. Recall that in the noisy binary search, in each loop we first find t as the middle of t_{high} and t_{low}, check $s = \text{INF}(p) - \text{INF}(\text{rank}^{-1}(t)) \leq 0$, add noise a Gaussian noise to s to get $\hat{s}$ and compare $\hat{s}$ with margin, which is ξ in this case. If $\hat{s} \geq \xi$, notice that at line 9, we change t_{high} to t. If in this case, $s \leq 0$, which means t is not an upper bound of rank, we never have chance to make t_{high} to be a valid upper bound of rank since it will only decrease in the further loops. Therefore, We say a loop is a failure if during that loop, $s \leq 0$ but $\hat{s} > \xi$. To have a valid rank upper bound, it is necessary to have no loop failure during the entire noisy binary search. We next show that the probability of no such a failure occur is at least β. See the chain of inequalities below.

$$Pr[\mathcal{I}_u^U \text{ is an upper bound of rank}(p_u; D, \mathcal{P}, I)]$$

The first inequality is due to the bound of the number of while loops. To be a rank bound, it cannot fail at each loop, therefore it has to success for all the N loops. These are independent events, so we can use a product for all the events happen together.

$$\geq (1 - Pr[\text{loop failure}])^N$$

The second inequality is due to the bound of $Pr[\text{loop failure}]$. Since any case such that $S \leq 0$ but $\hat{s} > \xi$ is considered as a loop failure, $\hat{s}$ is achieved by adding a Gaussian noise to s and ξ is a constant, the probability of a loop failure only depends on the value of s. Since here we have a condition about $s \leq 0$, $\sup_{s\leq 0} Pr[\hat{s} > \xi]$ is an upper bound of $Pr[\text{loop failure}]$.

$$\geq (1 - \sup_{s\leq 0} Pr[\hat{S} > \xi])^N$$

The next equality is because $\sup_{s\leq 0} Pr[\hat{s} > \xi] = Pr[N(0, \sigma^2) > \xi]$. Recall that $\hat{s} = s + N(0, \sigma^2)$ in line 8, therefore $\sup_{s\leq 0} Pr[\hat{s} > \xi] = \sup_{s\leq 0} Pr[s + N(0, \sigma^2) > \xi] = \sup_{s\leq 0} Pr[N(0, \sigma^2) > \xi - s]$. Since $Pr[N(0, \sigma^2) > \xi - s]$ increases as s increases, it achieves maximum at $s = 0$ for $s \leq 0$. Therefore, $\sup_{s\leq 0} Pr[\hat{s} > \xi] = Pr[N(0, \sigma^2) > \xi]$.

$$= Pr[\mathcal{N}(0, \sigma^2) \leq \xi]^N$$

The third bound is due to Chernoff bound of the Q-function (Lemma 2.1). Since $Pr[\mathcal{N}(0, \sigma^2) \leq \xi] = 1 - Pr[\mathcal{N}(0, 1) > \xi/\sigma]$, by Chernoff bound we have $Pr[\mathcal{N}(0, 1) > \xi/\sigma] \leq \exp(-(\xi/\sigma)^2/2)$ and therefore $Pr[\mathcal{N}(0, \sigma^2) \leq \xi] \geq 1 - \exp(-(\xi/\sigma)^2/2)$.

$$\geq \left(1 - \exp(-(\xi/\sigma)^2/2)\right)^N$$

The fourth bound is due to $(1 + x)^r \geq 1 + rx$ for $x \geq -1$ and $r \geq 1$.

$$\geq 1 - N\exp(-(\xi/\sigma)^2/2)$$

The final equality is by plugging $\xi = \sigma\sqrt{2\ln(N/(1-\beta))}$.

$$= \beta$$

Similarly, we have $Pr[\mathcal{I}_u^L$ is a lower bound of $\text{rank}(p_u; D, \mathcal{P}, I)] \geq \beta$. Together, the probability of $\mathcal{I}_u$ is a γ level confidence interval of $\text{rank}(p_u; D, \mathcal{P}, I)$ equals to both events $\mathcal{I}_u^U$ is an upper bound of $\text{rank}(p_u; D, \mathcal{P}, I)$ and $\mathcal{I}_u^L$ is a lower bound of $\text{rank}(p_u; D, \mathcal{P}, I)$ happen together, which is greater than or equal to the probability sum of each single event minus one (Lemma 2.4, which is $\beta + \beta - 1 = 2\beta - 1$. By plugging $\beta = (\gamma + 1)/2$ from line 13, we have $2\beta - 1 = \gamma$, which is the confidence interval level for the final confidence interval. □

4.4 Putting it all together

We now show how all steps fit into DPXPLAIN

Relative Influence Recall that the influence defined by Definition 4.1 is the difference of $(o_i - o_j)$ before and after removing the tuples related to an explanation predicate (first term), and multiplies with a normalizer to penalize trivial predicates (second term). Since the absolute value of influence is hard to interpret, to help user better understand the confidence interval of influence, we show the *relative influence* compared to the original difference $|o_i - o_j|$ as a percentage. However, we cannot divide the influence by $|o_i - o_j|$ since using the actual data values will incur additional privacy loss, hence, for SUM and COUNT we divide the true influence by $|\hat{o}_i - \hat{o}_j|$ as an approximation since the normalizer in

the second term is bounded in [0, 1]. However, when $agg = AVG$, the normalizer $\min(|g_i(\neg p(D))|, |g_j(\neg p(D)|)$ (second term) is not bounded in [0, 1], so we further divide the influence by another constant, the minimum of the noisy counts/sizes of the groups, i.e., $|\min(\hat{o}_i^C, \hat{o}_j^C)|$ (approximating the upper bound $\min(|g_i(D)|, |g_j(D)|)$ of the normalizer to avoid additional privacy loss). In summary, we define the relative influence $\widetilde{\text{INF}}(p; (\alpha_i, \alpha_j, >), D)$, or simply $\widetilde{\text{INF}}(p)$, as follows, which is only used for display purposes.

$$\widetilde{\text{INF}}(p) = \text{INF}(p) / \begin{cases} |\hat{o}_i - \hat{o}_j| & , \text{for } agg \in \{COUNT, SUM\} \\ |\hat{o}_i - \hat{o}_j| \times |\min(\hat{o}_i^C, \hat{o}_j^C)| & , \text{for } agg = AVG \end{cases}$$

Explanation Table We define the explanation table as follows.

Definition 4.2 [Explanation Table containing top-k explanations] Given a database D, a group-by aggregate query q as shown in Fig. 3, a user question $(\alpha_i, \alpha_j, >)$, a predicate space $\mathcal{P}$, a confidence level γ, and an integer k, a table of top-k explanations is a list of k 5-element tuples $(p_u, \mathcal{I}^L_{relinflu_u}, \mathcal{I}^U_{relinflu_u}, \mathcal{I}^L_{rank_u}, \mathcal{I}^U_{rank_u})$ for $u = 1, 2, \ldots, k$ such that p_u is an explanation predicate, $(\mathcal{I}^L_{relinflu_u}, \mathcal{I}^U_{relinflu_u})$ is a confidence interval of relative influence $\widetilde{\text{INF}}(p_u)$ with confidence level γ, and $(\mathcal{I}^L_{rank_u}, \mathcal{I}^U_{rank_u})$ is a confidence interval of $\text{rank}(p_u)$ with confidence level γ

Sorting the explanations in the explanation table Since this table contains the bounds of the influences and ranks it is natural to present the table as a sorted list. Since the numbers in the table are generated by random processes, each column may imply a different sorting. In this paper, we sort the selected top-k explanations by the upper bound of the relative influence CI (the third column in Fig. 1d) in descending order; if there is a tie, we break it using the upper bound of the rank confidence interval (the fifth column in Fig. 1d). Finding a principled way for sorting the explanation predicates is an intriguing subject of future work.

Overall DP guarantee We summarize the privacy guarantee of DPXPLAIN as follows: (i) the private noisy query answers returned by Gaussian mechanism in Phase-1 satisfy ρ_q-zCDP together (see Sect. 2); (ii) Phase-2 only returns the confidence intervals of the noisy answers in Phase-1 with zero additional privacy loss (discussed in Sect. 4.1); (iii) Phase-3 returns k explanation predicates and their upper and lower bounds on relative influence and ranks given a required confidence interval with three privacy parameters $\rho_{Topk}, \rho_{Influ}, \rho_{Rank}$ (discussed in Sect. 4.3.1, 4.3.2 and 4.3.3). The following theorem summarizes the total privacy guarantee.

Theorem 4.2 *Given a group-by query q and a user question comparing two aggregate values in the answers of q, the* DPXPLAIN *framework guarantees $(\rho_q + \rho_{Topk} + \rho_{Influ} + \rho_{Rank})$-zCDP.*

5 Extension to general user questions

In this section, we introduce a generalization of the user question (Definition 3.1) through weighted sum, such that more groups can be involved in the question and the comparison between groups can be more flexible. We also discuss how the explanation framework should be adapted to this general form. In Sect. 6, we give a use case for privately explaining a general user question.

Definition 5.1 [General User Question] Given a database D an aggregate query q, a DP mechanism $\mathcal{M}$, and noisy group aggregation releases $\hat{o}_{i_1}, \hat{o}_{i_2}, \ldots, \hat{o}_{i_m}$ of the groups $\alpha_{i_1}, \alpha_{i_2}, \ldots, \alpha_{i_m}$ from the query q, a general user question Q is represented by m weights and a constant c: $(w_{i_1}, w_{i_2}, \ldots, w_{i_m}, c)$. Intuitively, the question is interpreted as "Why $\sum_{j=i_1}^{i_m} w_j \hat{o}_j \geq c$".

Definition 5.1 allows more interesting questions, such as "Why the total salary of group A and B is larger than the total salary of group C and D?" or "Why the average salary of group A is 10 times larger than the one of group B?". Next we illustrate how the algorithms for each problem related to our framework should be adapted in the case of general user question.

Private Confidence Interval of Question Given a general user question $(w_{i_1}, w_{i_2}, \ldots, w_{i_m}, c)$, we discuss how to derive the confidence interval of $\sum_{j=i_1}^{i_m} w_j o_j - c$. Comparing to the case of a simple user question $(\alpha_i, >, \alpha_j)$, where the target of confidence interval is $o_i - o_j$, here we have a weighted sum of multiple group results. Therefore, when agg is CNT or SUM, the noisy weighted sum follows the Gaussian distribution with scale $\sqrt{\sum_{j=i_1}^{i_m} w_j^2}\sigma$, where σ is the noise scale used in query answering. When agg is AVG, the noisy weighted sum can also be viewed as a combination of multiple Gaussian variables. Thus, we consider these adaptions:

1. For $agg = CNT$ or $agg = SUM$, update the margin $\sqrt{2}(\sqrt{2}\sigma)\,\text{erf}^{-1}(\gamma)$ as $\sqrt{2}(\sqrt{\sum_{j=i_1}^{i_m} w_j^2}\sigma)\,\text{erf}^{-1}(\gamma)$.
2. For $agg = AVG$, update the sub confidence level β to be $(\gamma - 1)/(2m) + 1$, and the image of sub confidence intervals to be $\sum_{j=i_1}^{i_m} \mathcal{I}_j^S / \mathcal{I}_j^C - c$.

Adaptation to Finding Private Top-k Explanation Predicates Since the user question has a new form, the influence

function and its corresponding score function should also be adapted. We consider their natural extensions as follows:

Definition 5.2 [General Influence Function] Given a database D and a general user question $Q = (w_{i_1}, w_{i_2}, \ldots, w_{i_m}, c)$ with respect to the query SELECT A_{gb}, agg(A_{agg}) FROM R WHERE ϕ GROUP BY A_{gb}, the influence of an explanation predicate p is defined follows:

$$\text{INF}(p; Q, D) = \left(\sum_{j=i_1}^{i_m} w_j q(g_j(D)) - \sum_{j=i_1}^{i_m} w_j q(g_j(\neg p(D))) \right) \times \begin{cases} \frac{\min_{t \in \{i_1, i_2, \ldots, i_m\}} |g_t(\neg p(D))|}{\max_{t \in \{i_1, i_2, \ldots, i_m\}} |g_t(D)| + 1} & , agg \in \{COUNT, SUM\} \\ \min_{t \in \{i_1, i_2, \ldots, i_m\}} |g_t(\neg p(D))| & , agg = AVG \end{cases}$$

We can plug-in the new influence function into algorithm 2 to find the noisy top-k explanation predicates. The corresponding sensitivity of the new influence function is given as follows:

Theorem 5.1 *Given an explanation predicate p and a general user question $Q = (w_{i_1}, w_{i_2}, \ldots, w_{i_m}, c)$ with respect to a group-by query with aggregation agg, the following holds:*

1. *If $agg = CNT$, the sensitivity of* INF$(p; Q, D)$ *is* $2 \sum_{j=i_1}^{i_m} |w_j|$.
2. *If $agg = SUM$, the sensitivity of* INF$(p; Q, D)$ *is* $2 \sum_{j=i_1}^{i_m} |w_j| A_{agg}^{max}$.
3. *If $agg = AVG$, the sensitivity of* INF$(p; Q, D)$ *is* $8 \sum_{j=i_1}^{i_m} |w_j| A_{agg}^{max}$.

Proof This is a weighted version of Proposition 4.1. □

In particular, when only two groups are involved in the user question, each with weight 1, we get the results from Proposition 4.1. For example, in Item 1 when $agg = CNT$, we get from Theorem 5.1 that the sensitivity of INF is $2 \cdot (1 + 1) = 4$ which is consistent with Proposition 4.1.

We also allow explanation predicates to include disjunction and allow the framework to specify a specific set of explanation predicates by enumeration.

Adaptation to Finding a Private Confidence Interval of Influence We can plug-in the new influence function and their sensitivities into the original algorithm to find the confidence interval of influence.

Adaptation to Finding a Private Confidence Interval of Rank We can plug-in the new influence function and their sensitivities into algorithm 4 to find the confidence interval of rank.

6 Experiments

In this section, we evaluate the quality and efficiency of the explanations generated by DPXPLAIN. To our knowledge, there are no existing benchmarks for explanations for query answers (even without privacy consideration) in the database research literature. We have implemented DPXPLAIN [1] in Python 3.7.4 using the Pandas [97], NumPy [52], and SciPy [100] libraries. All experiments were run on Intel i7-7700 CPU @ 3.60GHz with 32 GB of RAM.

6.1 Experiment setup

We detail the data, queries, questions, and parameters.

Datasets We consider three datasets.

- **IPUMS-CPS (real data)**: A dataset of Current Population Survey from the U.S. Census Bureau [48] with 1,146,552 tuples
  from the year 2011 to 2019. The dataset contains 8 categorical attributes where domain sizes vary from 3 to 36 and one numerical attribute. The attribute AGE is discretized as 10 years per range, e.g., [0,10] is considered a single value. To set the domain of numerical attributes, we only include tuples with attribute INCTOT (the total income) smaller than 200k as a domain bound.
- **Greman-Credit (synthetic data)**: A corrected collection of credit data [51]. It includes 20 attributes where the domain sizes vary from 2 to 11 and a numerical attribute. Attributes duration, credit-amount, and age are discretized. The domain of attribute good-credit is zero or one. We synthesize the dataset to 1 million rows by combining a Bayesian network learner [8] and XGBoost [14] following the strategy of QUAIL [84].
- **New York City taxi trips (real data)**: Contains information from January and February 2019 [3] and is used to demonstrate a use case for our general form of user question (Definition 5.2). We preprocessed the dataset such that it includes 4 columns: PU_Zone, PU_Borough, DO_Zone, DO_Borough.

Queries and Questions The queries and questions used on the experiments are shown in Table 1.

Default setting of DPXPLAINUnless mentioned otherwise, the following default parameters are used (also for the motivating example) : $\rho_q = 0.1$, $\rho_{Topk} = 0.5$, $\rho_{Influ} = 0.5$, $\rho_{Rank} = 1.0$, $\gamma = 0.95$, $k = 5$, $\eta = 0.1$, and the number of conjuncts in explanation predicates $l = 1$ (Definition 3.2). We choose $\eta = 0.1$ to allocate more privacy budget for the rank upper bound by our observation that the scores of explanation predicates have a long and flat tail, which intuitively means that a tight rank upper bound indicates a precise score and, thus, costs more privacy. For the total privacy budget, which is 2.1 by default, we provide experiments to show that reducing the budget of each component can still lead to a high utility for all questions except I2 and I5 in Table 1 (Figs. 7, 8a, 9a, 9b).

Table 1 Examined queries and questions; `Valid` indicates if it is a valid question on the hidden true data

Data	Query	Question	Valid
`IPUMS-CPS`	q_1: AVG(`INCTOT`) by `SEX`	I1: Why `Male` > `Female` ?	Yes
	q_2: `INCTOT` by `RELATE`	I2: Why `Grandchild` > `Foster children` ?	Yes
		I3: Why `Head/householder` > `Spouse` ?	No
	q_3: `INCTOT` by `EDUC`	I4: Why `Bachelor` > `High school` ?	Yes
		I5: Why `Grade 9` > `None or preschool` ?	No
`German-Credit`	q_4: AVG(`good-credit`) by `status`	G1: Why `no balance` > `no chk account` ?	Yes
	q_5: AVG(`good-credit`) by `purpose`	G2: Why `car (new)` > `car (used)` ?	Yes
		G3: Why `business` > `vacation` ?	No
	q_6: AVG(`good-credit`) by `residence`	G4: Why `"< 1 yr"` > `">= 7 yrs"` ?	Yes
		G5: Why `"[1, 4) yrs"` > `"[4, 7) yrs"` ?	No

Explanation p.	Rel Inf 95%-CI L	Rel Inf 95%-CI U	Rank 95%-CI L	Rank 95%-CI U	Rel Inf (hidden)	Rank (hidden)
RELATE = "Head/ householder"	12.18%	12.52%	1	1	12.41%	1
EDUC = "Bachelor's degree"	7.10%	7.45%	2	3	7.32%	2
RACE = "White"	6.41%	6.75%	2	5	6.54%	3
RELATE = "Spouse"	5.70%	6.04%	2	5	6.01%	4
CLASSWKR = "NIU"	3.83%	4.18%	2	6	4.22%	5

Fig. 5 Phase-3 of DPXPLAIN in Case-1

Explanation p.	Rel Inf 95%-CI L	Rel Inf 95%-CI U	Rank 95%-CI L	Rank 95%-CI U	Rel Inf (hidden)	Rank (hidden)
existing-credits = "1"	77.90%	78.99%	1	1	78.16%	1
job = "skilled employee / official"	71.21%	72.29%	1	2	71.83%	2
sex-marst = "male : married/widowed"	54.34%	55.42%	2	4	55.10%	3
credit-amount = "(500, 2500]"	50.01%	51.10%	2	5	50.27%	4
credit-history = "no credits taken/all credits paid back duly"	49.07%	50.16%	4	5	49.14%	5

Fig. 6 Phase-3 of DPXPLAIN in Case-2

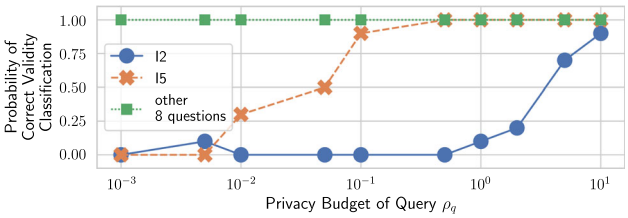


Fig. 7 The probability of correctly validating user questions. All questions except I2 and I5 (Fig. 7) are at 100%

6.2 Case studies

Case-1 IPUMS-CPS In **Phase-1**, the user submits a query q_1 from Table 1, and gets a noisy result: ("Female", 31135.25) and ("Male", 45778.46). The hidden true values are ("Female", 31135.78) and ("Male", 45778.39). Next, in **Phase-2**, since there is a gap of 14643.21 between two groups, the user asks a question I1 from Table 1. The framework then quantifies the noise in the question by reporting a confidence interval of the gap as (14636.63, 14649.79). Since the interval does not include zero, DPXPLAIN suggests that this is a valid question, which is correct. Finally, in **Phase-3**, the framework presents top-5 explanations to the user as Fig. 5 shows. The last two columns are the true relative influences and ranks. We correctly find the top-5 explanation predicates, and the first and fourth explanations together suggests that a married man tends to earn more than a married woman, which is supported by the wage disparities in the labor market [99]. The second and third explanations also match the wage disparities within the educated group and white people. The total runtime for preparing the explanations in Phase-2 and Phase-3 is 67 s.

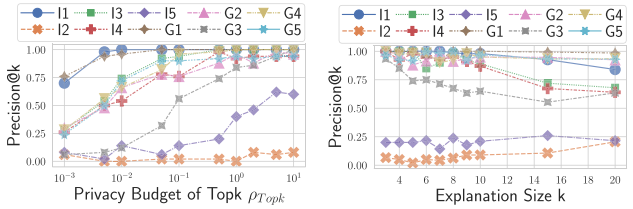
Case-2 German-Credit In **Phase-1**, the user submits a query q_4 from Table 1, and gets a noisy result: ("no checking account", 0.526571) and ("no balance", 0.574447). The true hidden result is ("no checking account", 0.526574) and ("no balance", 0.574466). Next, in **Phase-2**, since there is a gap of 0.047876 between two groups, the user asks a question G1 from Table 1. The framework then quantifies the noise in the question by reporting a confidence interval of the gap as (0.047786, 0.047967). Since the interval does not include zero, the framework suggests that this is a valid question, which is correct. Finally, in **Phase-3**, the framework presents top-5 explanations to the user as Fig. 6 shows. The last two columns are the true relative influences and ranks. We correctly find the top-5 explanations, and the first explanation suggests that for a person who already has a credit in the bank, the bank tends to mark the credit as good with a higher probability than the case of no account if she has a checking account even with zero balance, which follows the intuition that a person having a credit account but no checking account is risky to the bank. The total runtime for preparing the explanations in Phase-2 and Phase-3 is 40 s.

6.3 Accuracy and performance analysis

We detail our experimental analysis for the different questions and configurations of DPXPLAIN. All results are averaged over 10 runs.

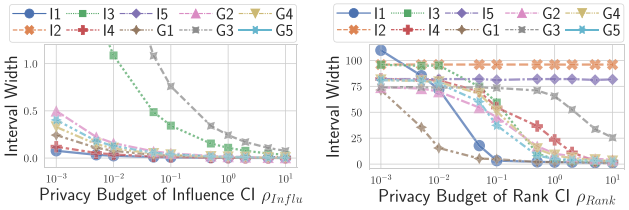
Correctness of noise interval In Phase-2 of DPXPLAIN, the validity of the question is suggested as follows: if the confidence interval contains non-positive numbers, the question is invalid, otherwise valid. From Fig. 7, we find that 8 out of 10 questions (plotted together for clarity) from Table 1 are classified correctly with an accuracy of 100% given a wide range of privacy budget of query ρ_q. However, there are two questions, I2 and I5, only show high accuracy given a large privacy budget of $\rho_q = 10$. One reason is that the minimum group size involved in I2 and I5 is at least 600 and 60 times smaller compared to other questions, and, therefore, the partial confidence intervals in the denominators of the AVG query are low, which makes the final confidence intervals wider including negative numbers when it should not.

Accuracy of top-k explanation predicates In Phase-3 of DPXPLAIN, we first select top-k explanation predicates. We measure the accuracy of the selection by Precision@k [54], the fraction of the selected top-k explanation predicates that are actually ranked within top-k. Another experiment on the full ranking is included in the full version [2]. From Fig. 8a, we find that the privacy budget of top-k selection ρ_{Topk} has



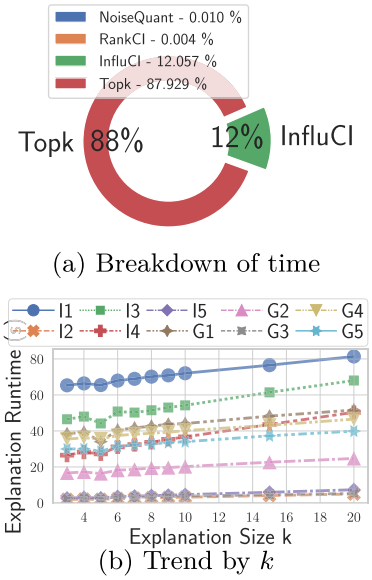
(a) Trend by ρ_{Topk}. (b) Trend by k.

Fig. 8 Precision@k of top-k selection by DPXPLAIN



(a) Relative Influence (b) Rank

Fig. 9 The width of confidence intervals by DPXPLAIN. The numbers are beyond 2 for the relative influence of I2 and I5



(a) Breakdown of time

(b) Trend by k

Fig. 10 Runtime analysis of DPXPLAIN

a positive effect to Precision@k at k = 5 for various questions. When $\rho_{Topk} = 1.0$, all the questions except I2 and I5 have Precision@k ≥ 0.8. The selection accuracy of question I2 and I5 are generally lower because of small group sizes, and, therefore, the influences of explanation predicates are small and the rankings are perturbed by the noise more significantly.

From Fig. 8b, we find that the trend of Precision@k by k is different across questions and there is no clear trend that Precision@k increases as k increases. For example, for G3, it first decreases from k=3 to k=5, but increases from k=5 to k=6. When k = 3, most questions have high Precision@k; this is because the highest three influences are much higher than the others, which makes the probability high to include the true top three. With larger k, explanation predicates that have similar scores have an equal probability to be included in top-k and therefore the top-k selected by the algorithm are different from the true top-k selections. The relationship between Precision@k and k depends on the distribution of all the explanation predicate influences.

Precision of relative influence and rank confidence Interval (CI) In Phase-3, the last step is to describe the selected top-k explanation predicates by a CI of relative influence and rank for each. To measure the precision of the description, we adopt the measure of **interval width** [47]. Figure 9 illustrates the average width of k CIs of relative influence and rank. From Fig. 9a and 9b, we find that the increase of privacy budget ρ_{Influ} and ρ_{Rank} shrinks the interval width of relative influence CI and rank CI separately. In particular, when $\rho_{Influ} \geq 0.5$, 6 out of 10 questions have the interval width of relative influence CI ≤ 0.025; when $\rho_{Rank} \geq 1.0$, 2 questions have the interval width of rank CI ≤ 2 and 6 questions have this number ≤ 10. We also measure the **effect of confidence level** γ to the CI by changing γ from 0.1 to 0.9 by step size 0.1 and from 0.95 and 0.99. Figures can be found in the full version [2]. The results show that it has a non-significant effect to the interval width, as it changes < 0.03 for the influence CI of 6 questions, and changes < 5 for the rank CI of 8 questions.

Runtime analysis We analyze the runtime of DPXPLAIN for generating Phase-2 and Phase-3. Figure 10a shows a runtime breakdown on average for all the questions from Table 1 with total runtime of 32 s on average. 88% of the time is used for the top-k explanation predicate selection procedure, especially on computing the influences for all the explanation predicates. The next highest runtime is for computing the confidence interval of influence, which needs to evaluate each sub queries. For the step noise quantification and confidence interval of rank, the time usage is not significant since the first only needs to find the image of two intervals and the second is a binary search. Figure 10b, shows that the runtime is linearly proportional to the size of explanations k, and the difference between questions is due to the difference of group sizes. We also find the runtime grows exponentially with the number of conjuncts l as the number of explanation predicates grows exponentially: for $l = 1, 2, 3$, the runtime about question I1 is 67, 3078 and 79634, and for question G1 it is 40, 1587 and 39922 s.

General User Question Use Case: Taxi-Imbalance We use the New York City taxi trips dataset to analyze the traffic volume between boroughs. With privacy budget $\rho_q = 0.1$, the framework answers the user query as "SELECT PU_Borough, DO_Borough, CNT(*) FROM R GROUP BY

explanation predicate	Rel Influ 95%-CI		Rank 95%-CI	
	L	U	L	U
zone = "JFK Airport"	55.21%	72.18%	1	1
zone = "LaGuardia Airport"	28.75%	45.72%	1	3
zone = "Bay Ridge"	-6.64%	23.60%	3	127
zone = "Queensboro Hill"	-10.75%	6.22%	3	127
zone = "Flushing"	-12.52%	4.25%	3	127

Fig. 11 Top-5 explanations for Taxi-Imbalance

PU_Borough, DO_Borough". There are in total 49 groups, and among the query answers we have *(Brooklyn, Queens): 11,431* and *(Queens, Brooklyn): 121,934*. User then asks "Why Queens to Brooklyn has more than 10 times the number of trips from Brooklyn to Queens?" This corresponds to the question "why $q_1 - 10q_2 \geq 0$", or in the form of weights $(1, -10, 0)$. The confidence interval of the question is $(7580, 7668)$, which validates the question. To explain the question, we consider a predicate space of the form "PU_Zone = <zone> $\vee$ DO_Zone = <zone>" with in total 127 different zones. With $\rho_{Topk} = 0.025$, $\rho_{Influ} = 0.025$, and $\rho_{Rank} = 0.95$, we have the explanation table as shown in Fig. 11. The relative influence is relative to the noisy difference $\hat{o}_1 - 10\hat{o}_2 = 7624$. From this table, we can find that two airports, JFK and LaGuardia airports that are located in Queens, are the major reasons for why there are more traffic volume from Queens to Brooklyn since there are more incoming taxi traffic to the airports instead of outgoing taxi traffic.

7 Related work

We next survey related work in the fields of DP and explanations for query results. *To the best of our knowledge,* DPXPLAIN *is the first work that explains aggregate query results while satisfying DP.*

Explanations for query results The database community has proposed several approaches to explaining aggregate and non-aggregate queries in multiple previous works. Proposed approaches include provenance [18, 27, 28, 55, 56, 64, 65, 98], intervention [85, 86, 104], entropy [44], responsibility [75, 76], Shapley values [69, 82], counterbalance [77] and augmented provenance [67], and several of these approaches have used predicates on tuple values as explanations like DPXPLAIN, e.g., [44, 67, 86, 104]. We note that any approaches that consider individual tuples or explicit tuple sets in any form as explanations (e.g., [27, 65, 69, 75]) cannot be applied in the DP setting since they would violate privacy. Among the other summarization or predicate-based approaches, Scorpion [104] explains outliers in query results with the intervention of most influential predicates. Our influence function (Sect. 4.2) is inspired by the influence function of Scorpion, but has been modified to deliver accurate results while satisfying DP. Another intervention-based work [86] that also uses explanation predicates, models interdependence among tuples from multiple relations with causal paths. DPXPLAIN does not support joins in the queries, which is a challenging future work (see Sect. 8).

Differential privacy Private SQL query answering systems [33–35, 58, 61, 62, 71, 95, 103] consider a workload of aggregation queries with or without joins on a single or multi-relational database, but none supports explanation under differential privacy. The selection of private top-k candidates is well-studied by the community [9, 11, 12, 16, 19, 31, 38, 63, 68, 72, 73, 81, 96]. We adopt One-shot Top-k mechanism [81] since it is easy to understand. Private confidence interval is a new trend of estimating the uncertainty under differential privacy [13, 22, 46], however, the current bootstrap based methods measure the uncertainty from both the sampling process and the noise injection, while we only focus on the second part which is likely to give tighter intervals. The most relevant work to the private rank estimation is private quantile [5, 21, 40, 49, 59, 66, 90], which is to find the value given a position such as median, but the problem of rank estimation in our setting is reversed. A recent work focused on explaining the effect of the privacy budget on the results obtained from DP process on their data [79] while another work provided a model that determines the impact of model explanations on the privacy of the model and developed a defense framework [80].

Privacy and provenance As mentioned earlier, data provenance is often used for explaining query results, mainly for non-aggregate queries. Within the context of provenance privacy [7, 10, 20, 23, 24, 87, 89, 92], one line of work [23–25] studied the preservation of workflow privacy (privacy of data transferred in a workflow with multiple modules or functions), with a privacy criterion inspired by l-diversity [70]. A recent work [29] explored what can be inferred about the *query* from provenance-based explanations and found that the query can be reversed-engineered from the provenance in various semirings [50]. To account for this, a follow-up paper [26] proposed an approach for provenance obfuscation that is based on abstraction. This work uses k-anonymity [91] to measure how many 'good' queries can generate concrete provenance that can be mapped to the abstracted provenance, thus quantifying the privacy of the underlying query. Another work proposed the use of data provenance for improved user understanding in DP settings [53] Devising techniques for releasing provenance of non-aggregate and aggregate queries under DP is an interesting research direction.

8 Future work

There are several interesting future directions. Extending DPXPLAIN to more general queries (like joins) and questions is an important future work. Unlike standard explanation frameworks like [104] where the join results can be materialized before running the explanation mechanism, a careful sensitivity analysis of adding/removing tuples from multiple tables is needed in the DP settings [95]. Second, the complexity of the top-k selection algorithm links to the number of explanation predicates that could be exponentially large, leaving room for future improvements. Additionally, other interesting notions of explanations for query answers (e.g., [67, 69, 77]) can be explored in the DP setting. Finally, evaluating our approach with a comprehensive user study and examining different metrics of understandability of the explanations generated by DPXPLAIN is also an important direction for future investigation.

Acknowledgements This work was supported by NSF awards IIS-2147061, IIS-2016393, IIS-2008107, IIS-1703431, IIS-1552538, by the Israeli Science Foundation (ISF) Grant No. 1702/24, and the Alon Scholarship.

Funding Open access funding provided by Hebrew University of Jerusalem.

References

1. Codebase of DPXPlain. https://github.com/yuchaotao/Private-Explanation-System
2. DPXPlain: Explaining query results under differential privacy. https://arxiv.org/abs/2209.01286
3. New york city taxi and limousine commission (tlc) trip record data. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page. Accessed: 2022-01-01
4. Abowd, J. M.: The us census bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2867–2867, (2018)
5. Alabi, D., Ben-Eliezer, O., Chaturvedi, A.: Bounded space differentially private quantiles. *CoRR*, abs/2201.03380, (2022)
6. Amsterdamer, Y., Deutch, D., Tannen, V.: Provenance for aggregate queries. In *PODS*, pages 153–164, (2011)
7. Anderson, P., Cheney, J.: Toward provenance-based security for configuration languages. In U. A. Acar and T. J. Green, editors, *4th Workshop on the Theory and Practice of Provenance, TaPP*, (2012)
8. Ankan, A., Panda, A.: pgmpy: Probabilistic graphical models using python. In *Proceedings of the 14th Python in Science Conference (scipy 2015)*, pages 6–11. Citeseer, (2015)
9. Bafna, M., Ullman, J.: The price of selection in differential privacy. In *Conference on Learning Theory*, pages 151–168. PMLR, (2017)
10. Bertino, E., Ghinita, G., Kantarcioglu, M., Nguyen, D., Park, J., Sandhu, R.S., Sultana, S., Thuraisingham, B.M., Xu, S.: A roadmap for privacy-enhanced secure data provenance. J. Intell. Inf. Syst. **43**(3), 481–501 (2014)
11. Bhaskar, R., Laxman, S., Smith, A., Thakurta, A.: Discovering frequent patterns in sensitive data. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 503–512, (2010)
12. Bonomi, L., Xiong, L.: Mining frequent patterns with differential privacy. Proc. VLDB Endow. **6**(12), 1422–1427 (2013)
13. Brawner, T., Honaker, J.: Bootstrap inference and differential privacy: Standard errors for free. *Unpublished Manuscript*, (2018)
14. Brownlee, J.: *XGBoost With python: Gradient boosted trees with XGBoost and scikit-learn*. Machine Learning Mastery, (2016)
15. Bun, M., Steinke, T.: Concentrated differential privacy: Simplifications, extensions, and lower bounds. In: Theory of Cryptography Conference, pp. 635–658. Springer (2016)
16. Carvalho, R. S., Wang, K., Gondara, L., Miao, C.: Differentially private top-k selection via stability on unknown domain. In *Conference on Uncertainty in Artificial Intelligence*, pages 1109–1118. PMLR, (2020)
17. Cesar, M., Rogers, R.: Bounding, concentrating, and truncating: Unifying privacy loss composition for data analytics. In *Algorithmic Learning Theory*, pages 421–457. PMLR, (2021)
18. A. Chapman and H. V. Jagadish. Why not? In *SIGMOD*, pages 523–534, 2009
19. Chaudhuri, K., Hsu, D., Song, S.: The large margin mechanism for differentially private maximization. arXiv preprint arXiv:1409.2177, (2014)
20. Cheney, J.: A formal framework for provenance security. In: CSF, pp. 281–293. Cernay-la-Ville, France (2011)
21. Cormode, G., Kulkarni, T., Srivastava, D.: Answering range queries under local differential privacy. Proc. VLDB Endow. **12**(10), 1126–1138 (2019)
22. Covington, C., He, X., Honaker, J., Kamath, G.: Unbiased statistical estimation and valid confidence intervals under differential privacy. arXiv preprint arXiv:2110.14465, (2021)
23. Davidson, S.B., Khanna, S., Milo, T., Panigrahi, D., Roy, S.: Provenance views for module privacy. In PODS (2011). https://doi.org/10.1145/1989284.1989305
24. Davidson, S.B., Khanna, S., Roy, S., Stoyanovich, J., Tannen, V., Chen, Y.: On provenance and privacy. In ICDT (2011). https://doi.org/10.1145/1938551.1938554
25. Davidson, S. B., Khanna, S., Tannen, V., Roy, S., Chen, Y., Milo, T., Stoyanovich, J.: Enabling privacy in provenance-aware workflow systems. In *CIDR*, pages 215–218, (2011)
26. Deutch, D., Frankenthal, A., Gilad, A., Moskovitch, Y.: On optimizing the trade-off between privacy and utility in data provenance. In SIGMOD (2021). https://doi.org/10.1145/3448016.3452835
27. Deutch, D., Frost, N., Gilad, A.: Explaining natural language query results. VLDB J. **29**(1), 485–508 (2020)
28. Deutch, D., Frost, N., Gilad, A., Haimovich, T.: Explaining missing query results in natural language. In *EDBT*, pages 427–430, (2020)

29. D. Deutch and A. Gilad. Reverse-engineering conjunctive queries from provenance examples. In *EDBT*, pages 277–288, 2019
30. Ding, B., Kulkarni, J., Yekhanin, S.: Collecting telemetry data privately. *Advances in Neural Information Processing Systems*, 30, (2017)
31. Z. Ding, D. Kifer, T. Steinke, Y. Wang, Y. Xiao, D. Zhang, et al. The permute-and-flip mechanism is identical to report-noisy-max with exponential noise. arXiv preprint arXiv:2105.07260, 2021
32. Dong, J., Durfee, D., Rogers, R.: Optimal differential privacy composition for exponential mechanisms. Int. Conf. Mach. Learn. **119**, 2597–2606 (2020)
33. Dong, W., Fang, J., Yi, K., Tao, Y., Machanavajjhala, A.: R2t,: Instance-optimal truncation for differentially private query evaluation with foreign keys. ACM SIGMOD Int. Conf. Manage. Data, Proc (2022). https://doi.org/10.1145/3514221.3517844
34. Dong, W., Yi, K.: Residual sensitivity for differentially private multi-way joins. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, page 432-444, New York, NY, USA, 2021
35. Dong, W., Yi, K.: A nearly instance-optimal differentially private mechanism for conjunctive queries. In *Proceedings of the 41st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS '22, page 213-225, New York, NY, USA, 2022. Association for Computing Machinery
36. Dua, D., Graff, C.: UCI machine learning repository, (2017)
37. Durfee, D., Rogers, R.: One-shot dp top-k mechanisms. DifferentialPrivacy.org, 08 2021. https://differentialprivacy.org/one-shot-top-k/
38. D. Durfee and R. M. Rogers. Practical differentially private top-k selection with pay-what-you-get composition. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3527–3537, 2019
39. Dwork, C.: Differential privacy and the us census. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 1–1, (2019)
40. Dwork, C., Lei, J.: Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380, (2009)
41. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of cryptography conference, pp. 265–284. Springer (2006)
42. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014)
43. Dwork, C., Rothblum, G. N.: Concentrated differential privacy. arXiv preprint arXiv:1603.01887, (2016)
44. El Gebaly, K., Agrawal, P., Golab, L., Korn, F., Srivastava, D.: Interpretable and informative explanations of outcomes. Proc. VLDB Endow. **8**(1), 61–72 (2014)
45. Erlingsson, Ú., Pihur, V., Korolova, A.: Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, (2014)
46. C. Ferrando, S. Wang, and D. Sheldon. General-purpose differentially-private confidence intervals. arXiv preprint arXiv:2006.07749, 2020
47. Ferrando, C., Wang, S., Sheldon, D.: Parametric bootstrap for differentially private confidence intervals, (2021)
48. S. Flood, M. King, R. Rodgers, S. Ruggles, J. R. Warren, and M. Westberry. 2021 Integrated public use microdata series, current population survey: Version 9.0 [dataset]. *Minneapolis, MN: IPUMS*, https://doi.org/10.18128/D030.V9.0
49. J. Gillenwater, M. Joseph, and A. Kulesza. Differentially private quantiles. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 3713–3722. PMLR, 2021
50. Green, T.J., Karvounarakis, G., Tannen, V.: Provenance semirings. PODS (2007). https://doi.org/10.1145/1265530.1265535
51. Groemping, U.: South german credit data Correcting a widely used data set. Rep. Math. Phys. Chem. Berlin Germany Tech. Rep. **4**, 2019 (2019)
52. Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E.: Array programming with NumPy. Nature **585**(7825), 357–362 (2020)
53. He, X., Zhang, S.: Differential privacy with fine-grained provenance: Opportunities and challenges. IEEE Data Eng. Bull. **47**(2), 21–49 (2024)
54. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Transa. Inf. Syst. (TOIS) **22**(1), 5–53 (2004)
55. Herschel, M., Hernández, M.A.: Explaining missing answers to SPJUA queries. PVLDB **3**(1), 185–196 (2010)
56. Huang, J., Chen, T., Doan, A., Naughton, J.F.: On the provenance of non-answers to queries over extracted data. PVLDB **1**(1), 736–747 (2008)
57. Jiang, B., Zhang, X., Cai, T.: Estimating the confidence interval for prediction errors of support vector machine classifiers. J. Mach. Learn. Res. **9**, 521–540 (2008)
58. Johnson, N., Near, J.P., Song, D.: Towards practical differential privacy for sql queries. Proc. VLDB Endow. **11**(5), 526–539 (2018)
59. Kaplan, H., Schnapp, S., Stemmer, U.: Differentially private approximate quantiles. *CoRR*, abs/2110.05429, (2021)
60. Kenny, C.T., Kuriwaki, S., McCartan, C., Rosenman, E.T., Simko, T., Imai, K.: The use of differential privacy for census data and its impact on redistricting: The case of the 2020 us census. Sci. Adv. **7**(41), eabk3283 (2021)
61. Kotsogiannis, I., Tao, Y., He, X., Fanaeepour, M., Machanavajjhala, A., Hay, M., Miklau, G.: Privatesql: a differentially private sql query engine. Proc. VLDB Endow. **12**(11), 1371–1384 (2019)
62. Kotsogiannis, I., Tao, Y., Machanavajjhala, A., Miklau, G., Hay, M.: Architecting a differentially private sql engine. In *CIDR*, (2019)
63. J. Lee and C. W. Clifton. Top-k frequent itemsets via differentially private fp-trees. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 931–940, 2014
64. Lee, S., Köhler, S., Ludäscher, B., Glavic, B.: A sql-middleware unifying why and why-not provenance for first-order queries. In ICDE, San Diego, CA, USA (2017)
65. Lee, S., Ludäscher, B., Glavic, B.: PUG: a framework and practical implementation for why and why-not provenance. VLDB J. **28**(1), 47–71 (2019)
66. Lei, J.: Differentially private m-estimators. *Advances in Neural Information Processing Systems*, (2011)
67. Li, C., Miao, Z., Zeng, Q., Glavic, B., Roy, S.: Putting things into context: rich explanations for query answers using join graphs. In SIGMOD (2021). https://doi.org/10.1145/3448016.3459246
68. Li, N., Qardaji, W.H., Su, D., Cao, J.: Priv'basis: frequent itemset mining with differential privacy. Proc. VLDB Endow. **5**(11), 1340–1351 (2012)
69. Livshits, E., Bertossi, L.E., Kimelfeld, B., Sebag, M.: The shapley value of tuples in query answering. ICDT **155**, 20 (2020)

70. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: L-diversity: Privacy beyond k-anonymity. TKDD **1**(1), 3 (2007)
71. McKenna, R., Miklau, G., Hay, M., Machanavajjhala, A.: Optimizing error of high-dimensional statistical queries under differential privacy. Proc. VLDB Endow. **11**(10), 1206–1219 (2018)
72. McKenna, R., Sheldon, D.R.: Permute-and-flip: A new mechanism for differentially private selection. Adv. Neural. Inf. Process. Syst. **33**, 193–203 (2020)
73. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, (2007)
74. McSherry, F. D.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30, 2009
75. Meliou, A., Gatterbauer, W., Moore, K.F., Suciu, D.: The complexity of causality and responsibility for query answers and non-answers. Proc. VLDB Endow. **4**(1), 34–45 (2010)
76. A. Meliou, W. Gatterbauer, S. Nath, and D. Suciu. Tracing data errors with view-conditioned causality. In T. K. Sellis, R. J. Miller, A. Kementsietsidis, and Y. Velegrakis, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June 12-16, 2011*, pages 505–516. ACM, 2011
77. Miao, Z., Zeng, Q., Glavic, B., Roy, S.: Going beyond provenance: Explaining query answers with pattern-based counterbalances. SIGMOD (2019). https://doi.org/10.1145/3299869.3300066
78. I. Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017
79. P. Nanayakkara, M. A. Smart, R. Cummings, G. Kaptchuk, and E. M. Redmiles. What are the chances? explaining the epsilon parameter in differential privacy. In J. A. Calandrino and C. Troncoso, editors, *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*, pages 1613–1630. USENIX Association, 2023
80. Nguyen, T. D. T., Lai, P., Phan, H., Thai, M. T.: Xrand: Differentially private defense against explanation-guided attacks. In B. Williams, Y. Chen, and J. Neville, editors, *AAAI*, pages 11873–11881. AAAI Press, (2023)
81. G. Qiao, W. J. Su, and L. Zhang. Oneshot differentially private top-k selection. arXiv preprint arXiv:2105.08233, 2021
82. Reshef, A., Kimelfeld, B., Livshits, E.: The impact of negation on the complexity of the shapley value in conjunctive queries. In D. Suciu, Y. Tao, and Z. Wei, editors, *PODS*, pages 285–297, (2020)
83. R. Rogers and T. Steinke. A better privacy analysis of the exponential mechanism. DifferentialPrivacy.org, 07 2021. https://differentialprivacy.org/exponential-mechanism-bounded-range/
84. Rosenblatt, L., Liu, X., Pouyanfar, S., de Leon, E., Desai, A., Allen, J.: Differentially private synthetic data: Applied evaluations and enhancements. arXiv preprint arXiv:2011.05537, (2020)
85. Roy, S., Orr, L.J., Suciu, D.: Explaining query answers with explanation-ready databases. Proc. VLDB Endow. **9**(4), 348–359 (2015)
86. Roy, S., Suciu, D.: A formal approach to finding explanations for database queries. In C. E. Dyreson, F. Li, and M. T. Özsu, editors, *SIGMOD*, pages 1579–1590, (2014)
87. Ruan, P., Chen, G., Dinh, A., Lin, Q., Ooi, B.C., Zhang, M.: Fine-grained, secure and efficient data provenance for blockchain. Proc. VLDB Endow. **12**(9), 975–988 (2019)
88. Ruggles, S., Fitch, C., Magnuson, D., Schroeder, J.: Differential privacy and census data: implications for social and economic research. AEA Paper. Proc. **109**, 403–08 (2019)
89. Sanchez, J.L.C., Bernabé, J.B., Skarmeta, A.F.: Towards privacy preserving data provenance for the internet of things. In WF-IoT, Singapore (2018)
90. Smith, A.: Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 813–822, (2011)
91. Sweeney, L.: K-anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzziness. Knowl.-Based Syst. **10**(5), 557–570 (2002)
92. Y. S. Tan, R. K. L. Ko, and G. Holmes. Security and data accountability in distributed systems: A provenance survey. In *HPCC/EUC*, pages 1571–1578, 2013
93. Tang, J., Korolova, A., Bai, X., Wang, X., Wang, X.: Privacy loss in apple's implementation of differential privacy on macos 10.12. arXiv preprint arXiv:1709.02753, 2017
94. Tao, Y., Gilad, A., Machanavajjhala, A., Roy, S.: Dpxplain: privately explaining aggregate query answers. Proc. VLDB Endow. **16**(1), 113–126 (2022)
95. Y. Tao, X. He, A. Machanavajjhala, and S. Roy. Computing local sensitivities of counting queries with joins. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 479–494, 2020
96. Thakurta, A. G., Smith, A.: Differentially private feature selection via stability arguments, and the robustness of the lasso. In *Conference on Learning Theory*, pages 819–850. PMLR, 2013
97. The pandas development team. pandas-dev/pandas: Pandas, Feb. 2020
98. Tran, Q. T., Chan, C.-Y.: How to conquer why-not questions. In *SIGMOD*, pages 15–26, (2010)
99. G. Vandenbroucke. Married men sit atop the wage ladder. 24, 2018
100. Virtanen, P., et al.: Fundamental algorithms for scientific computing in python. Nature Method. **17**, 261–272 (2020)
101. Wang, T., Tao, Y., Gilad, A., Machanavajjhala, A., Roy, S.: Explaining differentially private query results with dpxplain. Proc. VLDB Endow. **16**(12), 3962–3965 (2023)
102. Wasserman, L.: All of statistics: a concise course in statistical inference. Springer (2004)
103. R. J. Wilson, C. Y. Zhang, W. Lam, D. Desfontaines, D. Simmons-Marengo, and B. Gipson. Differentially private sql with bounded user contribution. arXiv preprint arXiv:1909.01917, 2019
104. Wu, E., Madden, S.: Scorpion: Explaining away outliers in aggregate queries. Proc. VLDB Endow. **6**(8), 553–564 (2013)
105. Yan, Z., Li, G., Liu, J.: Private rank aggregation under local differential privacy. Int. J. Intell. Syst. **35**(10), 1492–1519 (2020)
106. Zhang, J., Cormode, G., Procopiuc, C.M., Srivastava, D., Xiao, X.: Privbayes: private data release via bayesian networks. ACM Trans. Database Syst. (TODS) **42**(4), 1–41 (2017)