Privately Answering Queries on Skewed Data via Per-Record Differential Privacy

Jeremy Seeman
Tumult Labs, Durham, NC
jeremy.seeman@tmlt.io

William Sexton
Tumult Labs, Durham, NC
william.sexton@tmlt.io

David Pujol
Tumult Labs, Durham, NC
david.pujol@tmlt.io

Ashwin Machanavajjhala
Tumult Labs, Durham, NC
ashwin@tmlt.io

ABSTRACT

We consider the problem of the private release of statistics (like payroll) where it is critical to preserve the contribution made by a small number of outlying large entities. We propose a privacy formalism, per-record zero concentrated differential privacy (PzCDP), where the privacy loss associated with each record is a public function of that record's value. Unlike other formalisms which provide different privacy losses to different records, PzCDP's privacy loss depends explicitly on the confidential data. We define our formalism, derive its properties, and propose mechanisms which satisfy PzCDP that are uniquely suited to publishing skewed or heavy-tailed statistics, where a small number of records contribute substantially to query answers. This targeted relaxation helps overcome the difficulties of applying standard DP to these data products.

PVLDB Reference Format:
Jeremy Seeman, William Sexton, David Pujol, and Ashwin Machanavajjhala. Privately Answering Queries on Skewed Data via Per-Record Differential Privacy. PVLDB, 17(11): 3138 - 3150, 2024.
doi:10.14778/3681954.3681989

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://gitlab.com/dpujol/privately-answering-queries-on-skewed-data-via-per-record-differential-privacy.

1 INTRODUCTION

We consider the problem of releasing private aggregate statistics on data with highly skewed attributes. These kinds of data occur frequently in practice, for example the Census Bureau's County Business Patterns (CBP) dataset [8], USDA's Census of Agriculture [28], and many IRS data products like the Corporation Sourcebook [19]. Moreover, the aggregate statistics are highly sensitive to contributions by a single (or small set of) units. For example, CBP takes business establishments as its unit of analysis, where it is common that one establishment (like a large retailer or hospital) contributes to the majority of the jobs in a rural area. Regardless of their large contributions, the privacy of these units is often protected under federal law [1, 2]. Current disclosure avoidance methods, both traditional and modern, either fail to provide strong privacy guarantees or high utility. Classical statistical disclosure limitation techniques like complementary cell suppression using the p% rule and EZS noise [16, 24] offer no formal privacy guarantee and can deterministically reveal information about large contributions. Differentially private (DP) techniques [13, 14] that globally bound the contribution of any one unit to published statistics require that highly skewed data are either truncated or suppressed, resulting in unreasonably large bias or unreasonably large noise injected into published statistics.

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 17, No. 11 ISSN 2150-8097.
doi:10.14778/3681954.3681989

Table 1: Sample of skewed data containing a subset of a much larger collection of rows.

ID	Industry	Employees
1	Retail	5
2	Retail	5
3	Retail	10
4	Retail	1000
5	Technology	10000
6	Services	5
7	Hospitality	5

Table 1 gives one such data example. Suppose we wanted to execute the following SQL query using DP:

```
SELECT SUM(Employees) FROM table1 WHERE Industry = 'Retail'
```

DP requires bounding the contribution of any one record to the summation, enforced by truncating large values to a clamping bound. Setting this bound too high will require unreasonably large additional noise for DP, whereas setting this bound too low will add unreasonably large bias by truncating record 4. In Section 6.2, we further demonstrate how this problem makes global DP methods ill-suited for summations on skewed data.

Problems like these affect many high-sensitivity queries, where mitigating the effect of one record can have exorbitant effects on utility. For example, DP partition selection algorithms [11] used in keyset selection for group-by queries can neglect the impact of individual records with important, outlying attributes. In Table 1, if we were to execute the query "SELECT Count(*) FROM Table1 GROUP BY Industry" using DP, the probability of including "Services" and "Technology" would be the same, despite the fact that the majority of employees in the dataset work in Technology.

Issues like these stem from inherent connections between DP and robustness [5, 12, 23], where DP techniques cannot successfully answer inherently non-robust queries.

1.1 Contributions

In these settings where standard DP tools do not work, we might consider a relaxation of DP that applies different privacy loss bounds to different units. For example, in Table 1, we might allow additional privacy loss for record 5 (Technology establishment with 10000 employees) than record 1 (Retail establishment with 5 employees), especially because record 5 is an influential record which describes more individuals than record 1. However, existing techniques restrict how this mapping between records and privacy losses can occur. Personalized DP [20] requires that the mapping between units and their privacy losses is public knowledge, but we might require this mapping to non-trivially depend on confidential data. Alternatively, individual DP [15] requires that this mapping is confidential, but this limits our ability to be methodologically transparent or describe how privacy losses differ across units.

We propose per-record zero-concentrated DP (PRzCDP), which aims to address these problems. We publicly release what we call a "policy function" P that maps each possible (hypothetical) record to a maximum privacy loss that said record could incur. The form of P can depend on the record's value, allowing different privacy losses for outlying units when necessary to maintain reasonable data utility. We next propose an algorithm methodology, called *unit splitting*, which indirectly sets P by executing traditional zCDP [7] algorithms on a preprocessed version of the data where influential records are split into sub-records. Our formalism and mechanisms are ideal for non-interactive query workloads involving highly skewed data; as a result, our approach is a candidate methodology for data products like CBP [9]. Our contributions are as follows:

- In Sections 3 and 4, we formally define PRzCDP and demonstrate its formal properties, such as sequential and parallel composition.
- In Section 5, we propose unit splitting, a pre-processing step which allows us to compute DP mechanisms on the output so that the final results satisfy P-PRzCDP.
- In Section 6, we apply these techniques to three datasets: one simulated heavy-tailed dataset, one USDA dataset, and simulated CBP data provided to us by the U.S. Census Bureau[1]. We empirically demonstrate how small changes in privacy loss significantly improve utility for these skewed datasets.

1.2 Related work

Among the hundreds of existing formal privacy definitions [10], many formalisms aim to provide privacy guarantees which differ across units [4, 15, 18, 20] or realized datasets [25, 29]. These are often used in service of broader global DP goals, such as publishing data-dependent privacy guarantees [27] or establishing privacy filters for adaptive composition [17, 30]. Our work differs in a few key areas. First, we consider explicit dependencies between individual privacy loss parameters and confidential record values, relaxing strong assumptions made about this relationship in previous work [4, 15, 18, 20, 25, 27, 29]. Second, we consider data-dependent privacy guarantees without global bounds on privacy loss; we consider cases where the policy loss can become arbitrarily large for certain records. Such relaxations are necessary to address problems that arise with skewed data, for which global DP guarantees cannot provide reasonable privacy loss and utility simultaneously. We provide a detailed comparison between PRzCDP and related definitions in Section 4.1.

[1]The simulated CBP data has been provided to Tumult Labs as part of an ongoing contract with MITRE (Government Contract No. TIRNO-99-D00005 and subcontract MSA-000099) to provide scientific research support to the Economic Directorate of the US Census Bureau.

2 PRELIMINARIES

2.1 Data Model

We assume a single table schema $R(a_1, a_2 \dots a_d)$ where $\mathcal{A} = \{a_1, a_2 \dots a_d\}$ denotes the set of attributes R. Each attribute in a_i has domain, $\mathrm{Dom}(a_i)$, which need not be finite or bounded. The full domain of R is $\mathrm{Dom}(R) = \mathrm{Dom}(a_1) \times \dots \mathrm{Dom}(A_d)$.

A database D is an instance of relation R. D is a multi-set whose elements are tuples in $\mathrm{Dom}(R)$, *i.e.,* a tuple can be written as $r = (x_1, \dots, x_d)$ where $x_i \in \mathrm{Dom}(a_i)$. The number of tuples in D is denoted as $|D| = n$.

2.2 Zero-Concentrated Differential Privacy

Informally, a randomized mechanism M satisfies DP if the output distribution of the mechanism does not change too much with the addition or removal of a single unit's record. We focus on a variant of DP called ρ-Zero-Concentrated Differential Privacy (zCDP)[7], but all the following results could similarly be adapted for ϵ-DP [13]. This DP formulation bounds the Rényi Divergence of output distributions induced by changes in a single record. We first begin by defining neighboring databases.

Definition 1 (Neighboring Databases)**.** Two databases D and D' are considered neighboring databases if D and D' differ by adding or removing at most one row. We denote this relationship by $D' \approx D$.

We often use neighboring databases to measure the impact of any particular individual's input on the output of a function. We call the maximum change to a function due to the removal or addition of a single row the *sensitivity* of the function. We will often refer to a single row in a database as a "unit" and use the terms interchangeably.

Definition 2 (ℓ_2-Sensitivity)**.** Given a vector function f, the sensitivity of f is $sup_{D' \approx D} |f(D) - f(D')|_2$, where $| \cdot |_2$ is the ℓ_2-norm, and is denoted by Δ.

From here, we can define the formal notion of privacy under zCDP.

Definition 3 (Zero-Concentrated Differential Privacy)**.** A randomized mechanism M satisfies ρ-zCDP for $\rho \geq 0$ if, for any two neighboring databases D and D' and for all values of $\alpha \in (1, \infty)$:

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

where $D_\alpha(\cdot\|\cdot)$ is the Rényi divergence of order α between two probability distributions.

zCDP ensures that no unit contributes too much to the final output of the mechanism by bounding the difference with or without their particular record. The parameter ρ is often called the privacy loss and refers to the amount of information that can be learned about any particular individual. A high value of ρ means weaker privacy protection, while a lower value of ρ denotes a stronger privacy protection.

zCDP has a number of properties that are often used to construct more complex mechanisms. These are composability, post-processing invariance, and group privacy. Composition allows for multiple private mechanisms to compose together to create a large mechanism which still satisfies zCDP.

Theorem 1 (zCDP Sequential Composition [7]). *Let M_1, M_2 be randomized mechanisms which satisfy ρ_1-zCDP and ρ_2-zCDP respectively. Then the mechanism $M'(D) = (M_1(D), M_2(D))$ satisfies $(\rho_1 + \rho_2)-$zCDP.*

Additionally, if a mechanism is run on multiple disjoint sections of the database, private mechanisms compose with no additional privacy loss.

Theorem 2 (zCDP Parallel Composition [7]). *Let M_1, M_2 be randomized mechanisms which satisfy ρ_1-zCDP and ρ_2-zCDP respectively. Let D_1, D_2 be two disjoint subsets of a database, D. The mechanism $M'(D) = (M_1(D_1), M_2(D_2))$ satisfies $\max(\rho_1, \rho_2)-$zCDP.*

zCDP also allows for arbitrary post-processing without additional privacy loss.

Theorem 3 (zCDP Post-processing [7]). *Let M be a randomized mechanism which satisfies $\rho-$ zCDP. Let $M'(D) = f(M(D))$ for some arbitrary function f. Then M' satisfies ρ-zCDP.*

Due to the composition and post-processing theorems, most zCDP mechanisms are built out of simple primitive mechanisms such as the Gaussian mechanism [7] which are then post-processed and combined to create more complex mechanisms.

Definition 4 (Gaussian Mechanism [7]). Let q be a sensitivity Δ query. Consider the mechanism M that on input D releases a sample from $\mathcal{N}(q(D), \sigma^2)$. Then M satisfies $\frac{\Delta^2}{2\sigma^2}$-zCDP.

Another property of formally private mechanisms is the notion of group privacy. Mechanisms not only protect the unit but also protect arbitrary groups of units with a privacy loss that scales in the size of the group.

Theorem 4 (zCDP Group Privacy). *Let M be a randomized mechanism which satisfies $\rho-$zCDP. Then M guarantees $(k^2\rho)-$zCDP for groups of size k. That is, for every set of neighboring databases D, D' differing in up to k entries, and $\alpha \in (1, \infty)$ we have the following.*

$$D_\alpha(M(D) \| M(D')) \leq (k^2\rho) \cdot \alpha$$

While all the results that follow will use zCDP, they still hold in the context of pure ϵ-DP.

3 PER-RECORD DIFFERENTIAL PRIVACY

Here, we introduce Per-Record Zero-Concentrated Differential Privacy (PRzCDP), a relaxation of DP designed for highly skewed data. This takes the form of a privacy guarantee that varies as a function of the record's confidential value; for example, when records are real-valued positive numbers, we can consider privacy loss bounds that grow monotonically as a function of these record values. The key feature of PRzCDP is a *record-dependent policy function* which can be publicly released and analytically captures the privacy loss of a hypothetical record.

Definition 5 (Record-dependent policy function). A record-dependent policy function $P : \mathcal{T} \rightarrow \mathbb{R}_{\geq 0}$ denotes a maximum allowable privacy loss associated with a particular record value $r \in \mathcal{T}$, where $\mathcal{T}$ is the universe of possible records.

This record-dependent policy function differs from other individual privacy frameworks in that the parameter value $P(r)$ itself depends on the confidential record values r. We allow the functional form of the policy function $P(\cdot)$ to be made public, but the value of the policy function $P(r)$ for any record r in the confidential database cannot be made public. Record-dependent policy functions are inspired by and generalize binary policy functions of [21]. Given a policy function, Per-Record Zero-Concentrated Differential Privacy is defined as follows.

Definition 6 (P-per-record zero-Concentrated DP (P-PRzCDP)). Let M be a randomized algorithm which outputs a random variable Y over a range $(\mathcal{Y}, \mathcal{F}_Y)$, where $\mathcal{F}_Y$ is an appropriately chosen σ-algebra. M satisfies P-per-record zero-concentrated differential privacy (P-PRzCDP) iff $\forall D, D' \in \mathcal{D}$:

$$D \ominus D' = \{r\} \implies d_\alpha(M(D) || M(D')) \leq \alpha P(r) \quad \forall \alpha \in (1, \infty)$$

where $\mathcal{D}$ is the input database space and $\ominus$ denotes symmetric difference.

In this definition, the privacy loss associated with each record scales according to the policy function, as opposed to having equal privacy loss for all records. Note that under this definition, any traditional ρ-zCDP mechanism also satisfies PRzCDP with a policy function of $P(r) = \rho$, that is, a constant policy function.

Lemma 1. *Let M be a randomized mechanism which satisfies ρ-zCDP. Then M also satisfies P-PRzCDP where $P(r) = \rho$.*

Example 1. Consider Table 2(a) and a policy function $P(r) = \rho \left\lceil \frac{r[Employees]}{50} \right\rceil$, where ρ is a privacy parameter. Under this policy function, Establishment 1 would receive 3ρ privacy loss since it has 150 employees, while Establishment 2 would incur ρ privacy loss since it only has 50 employees. Establishment 5 would still incur ρ privacy loss even though it has less than 50 employees.

4 PROPERTIES OF PER-RECORD DIFFERENTIAL PRIVACY

We demonstrate here that PRzCDP satisfies the traditional properties often associated with Differential Privacy, as well as its variants. First, PRzCDP is closed under post-processing, in that any data independent function computed on the output of a mechanism which satisfies PRzCDP also satisfies PRzCDP.

Lemma 2 (Closure under post-processing). *Given $M : \mathcal{T}^* \rightarrow \mathcal{Y}$, a P-PRzCDP mechanism M, and any function f, it is the case that $f \circ M$ is also P-PRzCDP.*

Closure under post-processing is required for any formal privacy definition because it ensures that any private release will retain its properties regardless of the development or application of future privacy attacks.

PRzCDP satisfies *basic adaptive* sequential composition in that two PRzCDP mechanisms with arbitrary policy functions (chosen prior to running any mechanism) compose together to satisfy PRzCDP in combination.

Lemma 3 (Basic adaptive sequential composition for P-PRzCDP)**.** *Let M_1 satisfy P_1-PRzCDP and let M_2 satisfy P_2-PRzCDP. Then $M_3(D) = M_2\left(M_1(D), D\right)$ satisfies $(P_1(r) + P_2(r))$-PRzCDP.*

This ensures that multiple private releases still ensure privacy and demonstrates how the privacy loss decays over multiple releases. Like in DP, the privacy losses sum when mechanisms are composed together. In PRzCDP, since the privacy loss is encoded into the policy function, this takes the form of the sum of the two policy functions.

PRzCDP also satisfies a form of parallel composition. When multiple mechanisms which satisfy P-PRzCDP are run on disjoint subsets of the database, the joint result also satisfies P-PRzCDP:

Lemma 4 (Parallel composition for P-PRzCDP)**.** *Define the partition of size $J \in \mathbb{N} \cup \{\infty\}$: Let $\mathcal{T}$ define a partition over the universe of possible records. That is,*

$$\mathcal{T} = \bigcup_{j=1}^{J} C_j, \qquad C_i \cap C_j = \emptyset, i \neq j$$

Let $\mathcal{D}_j$ be the space of all databases containing only records in C_j for $j \in [J]$. Let $\{M_j\}_{j=1}^{J}$ be mechanisms satisfying P-PRzCDP for databases $D_j \in \mathcal{D}_j$ for $j \in [J]$, respectively. Then for any realized database D, the mechanism:

$$M(D) = \left\{M_j(D \cap C_j) \mid j \in [J]\right\}$$

satisfies P-PRzCDP. Note that the M_js can depend on their respective C_js, allowing for adaptivity.

Note that Lemma 4 does not require the individual mechanisms $\{M_j\}_{j=1}^{J}$ to be the same for all $j \in [J]$, allowing for adaptive parallel composition. Parallel composition allows for the combination of mechanisms on disjoint sets of the data universe. This results in a policy function which is a piecewise combination of the individual policy functions of each mechanism over their respective partition of the data universe.

PRzCDP satisfies the group privacy notion as well. A mechanism that satisfies P-PRzCDP also protects groups.

Lemma 5 (Simple group privacy for P-PRzCDP)**.** *Consider a sequence of databases $D_0, \ldots, D_J$ where $D_0 = D$ and $D_j \ominus D_{j-1} = \{r_j\}$ for $j \in [J]$. Let M be a randomized mechanism satisfying P-PRzCDP. Then we have:*

$$d_\alpha(M(D_0)||M(D_J)) \leq \alpha J \sum_{j=1}^{J} P(r_j) \tag{1}$$

Lemma 6 (Advanced group privacy for P-PRzCDP)**.** *Consider a sequence of databases $D_0, \ldots, D_J$ where $D_0 = D$ and $D_j \ominus D_{j-1} = \{r_j\}$ for $j \in [J]$. Let M be a randomized mechanism satisfying P-PRzCDP. Define $r_{(1)}, \ldots, r_{(J)}$ such that:*

$$P(r_{(1)}) \geq P(r_{(2)}) \geq \cdots \geq P(r_{(J)})$$

Then we have:

$$d_\alpha(M(D_0)||M(D_J)) \leq \alpha \inf_{k \in (1,\infty)} \sum_{j=1}^{J} \frac{k^j}{k-1} P(r_{(j)}) \tag{2}$$

The group privacy notions ensure that the formal privacy guarantee extends beyond the individual and also protects groups of arbitrary size, with the privacy loss growing as the size increases. Note that Lemma 5 yields a group privacy guarantee that's agnostic to database sequence order, whereas Lemma 6 yields a group privacy guarantee that depends on an optimal order-dependent sequence. By construction, Lemma 6 will always yield a smaller upper bound than Lemma 5.

4.1 Relation to Other Privacy Formulations

Prior work has studied the idea of giving different privacy guarantees to different records, and the idea of accounting privacy loss as a function of the data. The flexibility of PRzCDP, by comparison, lies in the fact that each unit's privacy loss is a *function* of their private record, and only that function is published instead of particular values. Units with knowledge of their private record use this public function to 'look up' their privacy loss. PRzCDP most closely resembles the definition for Personalized Differential Privacy (PDP) [20]. Like PRzCDP, PDP gives different guarantees to different records:

Definition 7 (Personalized zCDP (PDP)[20])**.** Let $\mathcal{T}$ be a universe of participating records and $\Phi : \mathcal{T} \mapsto \mathbb{R}^+$ be a function which maps each unit to a privacy loss. A randomized algorithm M satisfies Φ-PDP if, for any two databases D, D' which differ on the contributions of one unit $r \in \mathcal{T}$, we have

$$d_\alpha(M(D)||M(D')) \leq \alpha\Phi(r)$$

This definition contrasts with PRzCDP by assuming that $\Phi(r)$ is public knowledge for all $r \in \mathcal{T}$ *i.e.,* each unit's privacy guarantee is publishable. Fundamentally, this requires that the guarantee, $\Phi(r)$, of each unit, r, is independent of their private record. By comparison, P-PRzCDP does not make any assumptions about independence of a unit's privacy guarantee and their sensitive record. Thus, each unit's guarantee, $P(r)$ for the unit's record r, remains confidential. Only the policy function, $P(\cdot)$, is published and individual unit with knowledge of their private record can compute their own guarantee.

Other works have also studied computing privacy loss as a function of the data but only in an effort to give tighter accounting of the global privacy loss, which is constant across all participants and is made public. For instance, Papernot et al. [26] give tight privacy loss accounting by deriving a global loss from the data itself. This global loss is then passed through a novel mechanism to publish a noisy private version.

Similarly, the individualized accounting method of [17] computes each record's loss as a function of its data. However, this is only to ensure that no single record exceeds the public global privacy budget, constant across all records. Alternatively, [27] uses an existing global DP guarantee to provide an ex-post characterization for the gap between the global privacy loss and the confidential data-dependent realized privacy loss. The foundational distinguishing factor of PRzCDP from both these approaches is that only the policy is published, as opposed to any global or individual privacy losses.

DP encompasses a wide variety of formalisms [10] which rely on alternative characterizations of scenarios under comparison, measures of privacy loss between those scenarios, and the generality of bounds on these measures. Here, we show how PRzCDP is interoperable with these definitions.

Connections can be made to one-sided DP [21], also adapted to the semantics of zCDP:

Definition 8 (One-sided zCDP (OSzCDP)[21]). Let $P : \mathcal{R} \mapsto \{0, 1\}$ be a function that labels records as privacy-sensitive ($P(r) = 0$) or not ($P(r) = 1$). A randomized algorithm M satisfies (P, ρ)-one-sided zero-concentrated DP if, for any two databases D, D' where

$$D' = D \setminus \{r\} \cup \{r'\}, P(r) = 0, r \neq r' \tag{3}$$

we have

$$d_\alpha(M(D)||M(D')) \leq \alpha\rho \tag{4}$$

Lemma 7. *Suppose there exists $\rho \geq 0$ and a subset of records $R \subseteq \mathcal{T}$ such that:*

$$\sup_{r \in R} P(r) \leq \rho \tag{5}$$

Then for the policy $P^* : \mathcal{T} \mapsto \{0, 1\}$ where $P(r) = 1$ for $\{r \notin R\}$ and 0 otherwise, any mechanism M that is P-PRzCDP is also $(P^*, 2\rho)$-OSzCDP.

Note that for the problems under consideration, PRzCDP enables communication about the policy loss function without relying on assuming $P(r)$ is public knowledge for all r (as is true for Personalized zCDP) or $P(r)$ is confidential (as is true for individual DP).

5 MECHANISMS FOR PRZCDP

In this section, we present a novel class of privacy mechanisms for ensuring PRzCDP. This class of mechanisms is called *Unit Splitting*. As the name suggests, the mechanisms follow this general pattern:

- Preprocess the input dataframe by "splitting" each row or record into many smaller rows or sub-records. The number of splits depends on a measure of how large the row is.
- Next, we run a mechanism that satisfies standard ρ-zCDP.
- By group composition, the privacy loss of a row or record in the original dataframe will be $k^2\rho$, where k is the number of times an original row is split in the preprocessing step.

This allows a PRzCDP mechanism to be built by doing a preprocessing step followed by any arbitrary zCDP mechanism. We introduce the basics of unit splitting in Section 5.1, and describe how to use them to answer SQL aggregation queries and group-by aggregation queries in Sections 5.2 and 5.4.

5.1 Unit Splitting

Unit splitting is a preprocessing step that uses a mapping function $A(r)$ to map each record into one or more other records. Answering queries on the split records using a mechanism that satisfies zCDP results in an overall mechanism that satisfies PRzCDP. We state this more formally as follows.

Lemma 8 (ρ-zCDP with pre-processing implies P-PRzCDP). *Consider a pre-processing function $A : \mathcal{T} \mapsto \mathcal{T}^*$ where A maps each record $r \in \mathcal{T}$ to a multiset of records in $\mathcal{T}^*$. Let $|A(r)|$ be the cardinality of the multiset $A(r)$, i.e., the number of subsequent records generated by $A(r)$. If M is a ρ-zCDP algorithm operating on $\mathcal{D}^*$, then $M(A(D))$ satisfies P-PRzCDP where $P(r) = (\rho|A(r)|^2)$.*

Proof. Let D and D' be neighboring datasets and, without loss of generality, let $A(D') \setminus A(D) = \{s_1, \dots s_{|A(r)|}\} \subseteq \mathcal{T}^*$. By the group-privacy properties of ρ-zCDP:

$$D_\alpha(M(A(D))||M(A(D'))) \leq \alpha\left(|A(r)|^2\rho\right) \tag{6}$$

□

This allows a practitioner to create mechanisms which satisfy P-PRzCDP by using the unit-splitting preprocessing step followed by an off-the-shelf zCDP mechanism. This applies to all zCDP mechanisms from existing private frameworks such as Tumult Analytics [6], to complex mechanisms such as stochastic gradient descent [3] and the matrix mechanism [22]. The policy function in this case is implied by the choice of unit splitting. The number of splits per-record directly impacts the privacy loss of that record, with those that require a higher number of splits receiving a larger privacy loss than those with a smaller number of splits. However, the choice of splitting threshold bounds the sensitivity of the queries and consiquently the amount of noise required. As such, a smaller splitting threshold results in a smaller sensitivity and as such a smaller noise requirement. An empyrical analysis on how the choice of splitting threshold impacts both utility and the distribution of privacy loss can be found in Section 6.

In practice, a practitioner must choose a splitting threshold which balances the privacy loss of the records and the utility of the release. This can be done by either choosing a splitting threshold that results in the desired amount of noise, or by choosing a splitting threshold which results in reasonable privacy loss for the majority of records.

Since each record's privacy loss is now a function of the contents of the record, the privacy loss is considered a private value, and cannot be published. Instead, for transparency, the splitting function itself can be released, which would allow an observer to reason about the privacy loss of hypothetical records without releasing information about the records.

5.2 Answering Aggregation Queries Using Unit Splitting

We can privately compute aggregates such as sums on skewed data by applying unit splitting in the form of a splitting threshold. Doing so splits the few large contributors with possibly unbounded values into several smaller bounded values. This method both bounds and reduces the overall sensitivity of many queries, therefore allowing lower-error private answers. However, it comes at the cost of higher privacy loss for larger records which were split into multiple smaller records. The choice of splitting procedure results in an implicit policy function for PRzCDP.

Table 2: Sample table before and after unit splitting. The splitting thresholds used were as follows. Employees: 50, Payroll: $ 5,000,000

(a) Pre-Split Table

ID	Industry	Employees	Payroll
1	Agriculture	150	$ 10,000,000
2	Agriculture	50	$ 15,000,000
3	Mining	100	$ 10,000,000
4	Mining	50	$ 10,000,000
5	Retail	20	$ 1,000,000

(b) Post-Split Table

ID	Industry	Employees	Payroll
1	Agriculture	50	$ 5,000,000
1	Agriculture	50	$ 5,000,000
1	Agriculture	50	$ 0
2	Agriculture	50	$ 5,000,000
2	Agriculture	0	$ 5,000,000
2	Agriculture	0	$ 5,000,000
3	Mining	50	$ 5,000,000
3	Mining	50	$ 5,000,000
4	Mining	50	$ 5,000,000
4	Mining	0	$ 5,000,000
5	Retail	20	$ 1,000,000

We make a distinction between conditional attributes and measure attributes. Conditional attributes are those which will be used in conditional statements, such as the WHERE clause in an SQL query. These get duplicated across all splits of the data. Measure attributes are those which are computed in aggregations and thus get split across all the split records. For numerical measure attributes, we individually evaluate the minimum number of times each of the magnitude attributes a need to be split (*i.e.,* the smallest integer m such that $mT(a) \geq r(a)$). For example, if a row has $r(a) = 12$, and the splitting threshold is $T(a) = 5$, then the row would need to be split into at least 3 sub-records, corresponding to values of $\{5, 5, 2\}$. This minimum number of splits may be different for each attribute in any particular row; to resolve this difference, we select the magnitude attribute with the *largest* number of required splits. The remaining split elements are zero-padded. Ideally, each magnitude attribute will be split the same number of times, to prevent too many zero-padded elements. As a complete example, Table 2 lists how several records would be split according to these rules.

Once each record has been split, the split records are used to answer each query. For each query on the original unsplit records, we define an equivalent rewritten query on the split records. Table 3 describes how each query is rewritten.

Lemma 9. *As ρ tends to ∞, the difference between the private rewritten queries of Table 3 and the true query answers tends to* 0.

Lemma 9 simply states that the process of unit splitting itself incurs no bias for summation queries and asymptotically negligible bias for other inexact reconstructed queries. Since these aggregations are simply a zCDP mechanism after a mapping, by Lemma 8, the entire process satisfies P-PRzCDP where the policy function is dependent on the number of times each record is split.

Algorithm 1 Unit Splitting Pre-Processing

Require: D: Private dataframe.
Require: $T : \mathcal{A} \mapsto \mathbb{Z}$: splitting threshold function.
Ensure: $\{r_i\}$: multiset of unit splitting rows.

procedure UnitSplit(D, T)
  for $r \in$ Rows(D) **do**
    Find the smallest integer m such that $mT(a) \geq r(a)$ for all $a \in \mathcal{A}$
    Split r into m rows such that for each split and each attribute $r_i(a) \leq T(a)$ and $\sum_i r_i(a) = r(a)$
  end for
end procedure

Lemma 10. *Computing Algorithm 1 followed by an aggregation from Table 3 satisfies P-PRzCDP.*

Conditionals such as group-by and filters can be applied to split data without any adaptation, since the conditional attributes are duplicated across all splits. We give an example of answering a private sum below.

Example 2. Consider taking a sum over the Employees column of Table 2(a). We use the following splitting threshold (Employees: 50, Payroll: $5,000,000). The table after splitting can be found in Table 2(b). After splitting, the maximum value of the Employees column is 50 and each record is split into multiple rows with at most 50 employees. The same holds for Payroll and its associated threshold. Once the table is split, the sum is taken over all the split rows.

The PRzCDP policy function is implied by the splitting threshold. Each record incurs a privacy loss according to the number of times it is split. Establishments 1 and 2 are split 3 times and incurs a privacy loss of 9ρ. Establishments 3 and 4 are only split twice and each incurs a privacy loss of 4ρ. Establishment 5 is never split and incurs a privacy loss of ρ. Larger thresholds would reduce the number of times records are split, but increase errors due to DP noise; for example, doubling the split thresholds would quadruple the variance of the associated PzCDP mechanisms.

In this example, we used a sum, but this could include other aggregations such as averages or other more complex zCDP mechanisms such as partition selection [11], matrix mechanism [22] among others. One benefit of unit splitting is bounding the sensitivity of previously unbounded queries.

Example 3. Consider taking a sum over the Employees column of Table 2(a). Prior to unit splitting, the maximum possible number of employees for any arbitrary establishment is unbounded. There is no limit to the number of employees an establishment can have. After the splitting, the maximum value of the Employees column is set to 50, introducing a bound. Since the maximum number of employees in any record is 50, the sensitivity of the sum over employees is also 50.

In this case, the previously unbounded sensitivity is now bounded by the unit splitting algorithm. Traditionally, one would set a clamping bound on the sum, which would truncate all the values outside the bound to one of the boundary values. For heavily skewed data, this can either introduce bias (if the bound is too small) or a large amount of noise (if the bound is too large). When using unit splitting, this is no longer a concern, as the splitting threshold can be set low enough to avoid incurring too much noise while also incurring no bias. The tradeoff in this case is that the larger records incur a larger privacy loss overall. This makes PRzCDP particularly powerful in the case of highly skewed data, where a small amount of large records makes it impractical to introduce clamping bounds. Instead, these records are split into many smaller records and incur a larger privacy loss as a result.

5.3 Multiple Aggregations

While unit splitting is a versatile technique in its own right, much of its power comes from its ability to compose neatly with much of the existing DP literature as well as other instances of unit splitting. While any two PRzCDP mechanisms compose together due to Lemma 3, this also holds for mechanisms run prior to the unit splitting. This is because ρ-zCDP can be seen as a special case of PRzCDP where the policy function is equal to the privacy parameter ρ.

This allows mechanisms run prior to unit splitting to compose with those run after unit splitting by using Lemma 3. PRzCDP excels in reducing the sensitivity of high or unbounded sensitivity queries, such as sums or means. However, when computing queries with low sensitivities such as counts, medians, or low sensitivity sums, it is more efficient (possibly identically efficient) to use zCDP rather than use unit splitting. In these cases, each record only incurs a constant privacy loss (the ρ parameter given to the zCDP mechanism) as opposed to the variable privacy loss given by PRzCDP and unit splitting. We demonstrate this in the following example where zCDP is used to compute a median followed by a sum computed on unit split data.

Example 4. Consider the sum from Example 2 with the addition of a median query on the Payroll column prior to the unit splitting process. If we first compute the median with a privacy budget of ρ_1, then compute the sum with privacy-loss budget ρ_2, then by Lemma 3 the combination of the two mechanisms satisfies PRzCDP with a policy function of $\rho_1 + \rho_2|A(r)|^2$, where $|A(r)|$ is the maximum number of times record r is split. In this case, Establishments 1 and 2 incur $\rho_1 + 9\rho_2$ privacy loss, Establishments 3 and 4 incur $\rho_1 + 4\rho_2$ privacy loss, and Establishment 5 incurs $\rho_1 + \rho_2$ privacy loss.

By using traditional zCDP to compute the median, each record only incurs a constant (ρ_1) privacy loss. Had that median been computed after the unit splitting, the larger records would have incurred $9\rho_1$ privacy loss instead, a significant increase. In these cases, zCDP mechanisms can be used for tasks that are not subject to high sensitivity, such as medians, or complex tasks for which no unit split equivalent is available, such as stochastic gradient descent [3]. This allows for tighter analysis when using low sensitivity queries and opens up the vast literature of differentially private techniques for use alongside unit splitting.

5.4 Answer GroupBy Aggregation Queries Using Unit Splitting

In addition to aggregations such as sums and counts, unit splitting also supports conditional analysis such as filters and group-by. Filters and group-by can be applied directly on the conditional attributes after unit splitting, since those attributes are duplicated across all splits. This allows for individual analysis for each group.

Lemma 4 allows a practitioner to apply different splitting thresholds for each group in order to better serve the needs of each group. For example, consider if we added Technology as an additional industry in Table 2. Technology firms have significantly higher average pay than agricultural establishments and as such have a significantly higher payroll. In such cases, a different set of splitting thresholds may be necessary for technology firms to avoid extremely high privacy loss.

In cases where the group-by keyset is unknown, or is sparse in the domain of the attribute one can use a private partition selection algorithm, after the unit splitting process. Since the data has been split prior to partition selection, large records are instead split into many small records and are as such more likely to be discovered. We demonstrate an example of using all three techniques: group-by, multiple splitting thresholds, and split partition selection.

Example 5. Consider taking two sums over the Employee column and Payroll column of Table 2(a), grouped by the values of the Industry column. In addition, consider the following splitting thresholds for each industry. For agriculture and retail, use the previous splitting threshold of (Employees: 50, Payroll: $5,000,000). For mining, apply the splitting threshold of (Employees: 50, Payroll: $10,000,000).

To compute the sum, we need to complete three steps. First apply unit splitting to each industry with their own splitting threshold. This bounds the sensitivity of both the Employees and Payroll column, however this bound is now different for each industry. Then use some privacy budget ρ_1 to use a partition selection technique [11] to find the keyset for the group-by. Since each industry is split prior to the partition selection, those with few but large records still have a high probability to be in the resulting keyset. Then for each category in the Industry column, we take the sum over employees using the Gaussian mechanism with sensitivity 50 and privacy budget ρ_2. This sensitivity remains the same since the employee splitting threshold is the same for all industries. For sum over payroll, use the Gaussian mechanism with sensitivity 5,000,000 for agriculture and retail and use sensitivity 10,000,000 for mining. For both, we will use the same privacy budget ρ_3.

Since the partitions over the industry column form disjoint subsets of the dataset by Lemma 4, the sum over Employees and Payroll satisfy $\rho_2|A(r)|^2$ and $\rho_3|A(r)|^2$-PRzCDP respectively where $|A(r)|$ denotes the number of splits for each record for their respective splitting thresholds. We can combine all of these policy functions together using Lemma 3 to get that the overall mechanism satisfies P-PRzCDP with a policy function $P(r) = (\rho_1 + \rho_2 + \rho_3)|A(r)|^2$. For records 1 and 2, the final privacy loss is $9(\rho_1 + \rho_2 + \rho_3)$ since they are each split into 3 rows. Record 3 incurs $4(\rho_1 + \rho_2 + \rho_3)$ privacy loss since it is split into 2 rows under the new splitting threshold, and record 4 only incurs $\rho_1 + \rho_2 + \rho_3$ privacy loss since it is not split under the new splitting thresholds.

Table 3: Common exact queries and their reconstruction strategies

Original query	Rewritten Exact Query	PRzCDP $P(r)$
`COUNT(ROW_ID)`	`COUNT_DISTINCT(ROW_ID)`	ρ
`SUM(·)`	`SUM(·)`	$\rho\|A(r)\|^2$
`AVG(·)`	`SUM(·) / COUNT_DISTINCT(ROW_ID)`	$\rho\|A(r)\|^2 + \rho$

Since these were disjoint sections of the database, we could apply different splitting thresholds to each disjoint section and still satisfy P-PRzCDP. In this case, the new policy function would be a piece-wise function, giving a different functional form for each industry.

Due to the initial unit splitting, the partition selection step has a high probability of selecting each of the populated industries, even if those industries are populated by few but large establishments. This allows practitioners to properly analyze heavily skewed data where much of the data is compressed into relatively few records, which often happens in economic or population statistics.

6 EXPERIMENTS

In this section, we empirically demonstrate the effectiveness of PRzCDP when applied to skewed data. In Section 6.2 we demonstrate how the high bias and error from global zCDP results in an unacceptable trade-off between privacy and utility. Then, in Section 6.3, we focus on univariate queries to show how PRzCDP can be an effective alternative to zCDP with modest reductions in privacy. Finally, in Section 6.4, we demonstrate the methodology on our motivating use case.

6.1 Setup and Datasets

For each experiment, we answer queries using either zCDP or PRzCDP according to the Gaussian mechanism (Def. 4) with clamping-enforced sensitivity Δ, noise variance σ^2, and privacy loss $\rho = \frac{\Delta^2}{2\sigma^2}$, or unit splitting pre-processing (Algorithm 1) with additive Gaussian noise with variance σ^2 according to different splitting functions T. We use the following metrics throughout. Where contextually appropriate, we abuse notation and only include the relevant arguments.

Policy loss: for $r \in D$, we have

$$\text{PolicyLoss}(P, r) = P(r) \tag{7}$$

Note that in practice, we cannot release $\text{PolicyLoss}(P, r)$, only the functional form $P(\cdot)$.

Realized loss: for a record r, the realized dataset D, and mechanism M,

$$\text{RealizedLoss}(M, D, r) \tag{8}$$

$$= \sup_{\alpha \in (1,\infty)} \frac{d_\alpha(M(D)||M(D \setminus \{r\}))}{\alpha} \tag{9}$$

By construction, when M satisfies P-PRzCDP,

$$\text{PolicyLoss}(P, r) \geq \text{RealizedLoss}(M, D, r) \tag{10}$$

for all $r \in D$. Again, in practice, we cannot release $\text{RealizedLoss}(M, D, r)$ nor its functional form, as it depends on the realized dataset D.

Query relative error: for a dataset D, query $M(D)$, and non-private answer $S(D)$, define

$$\text{QueryRelErr}(M, S, D, \gamma) \tag{11}$$

$$= \min \left\{ v \in \mathbb{R}^+ \mid \mathbb{P}\left(\frac{|M(D) - S(D)|}{S(D)} \geq v \right) \leq 1 - \gamma \right\} \tag{12}$$

Absolute relative error (ARE): for a given output $M(D)$ and non-private answer $S(D)$, define

$$\text{ARE}(M(D), S(D)) = \frac{|M(D) - S(D)|}{|S(D)|} \tag{13}$$

Policy minimum: for a policy function P, we will use $\text{PolicyMin}(P) = \min_{r \in \mathcal{T}} P(r)$ to denote the smallest policy loss associated with one record. For unit splitting algorithms, this can be interpreted as the policy loss for records unaffected by unit splitting.

Our experiments are run on three different datasets: a simulated dataset (SIM), the National Agricultural Statistical Service Cattle Inventory Survey (CIS), and a simulated County Business Patterns (CBP) dataset.

The first dataset is simulated data for which we know the precise data generating distribution. This allows us to illustrate the kinds of heavy-tailed behavior that our methodology can better accommodate, as opposed to global DP. We simulate two heavy-tailed variables in $[1, \infty)$ with tail index parameters α, where smaller values of α correspond to heavier tails. The simulated variables are listed in Table 4; note that both variables HT1 and HT2 have infinite variance. Our goal is to answer sum queries of HT1 and HT2 grouped by CatIX.

Table 4: Simulated heavy-tailed variables

Name	Domain	Distribution
CatIX	$\{1, \ldots, 1000\}$	Categorical(ϕ)
HT1	$[1, \infty)$	Pareto$(1, 1.2)$
HT2	$[1, \infty)$	Pareto$(1, 1.5)$

The second dataset is from the U.S. Department of Agriculture (USDA)'s Cattle Inventory survey (CIS), managed by the National Agricultural Statistical Service (NASS) [28]. Our records consist of county-level survey records of total cattle inventory and average pastureland rent cost (in dollars per acre). Each county geography is contained hierarchically within an agricultural district (AD), itself contained within a particular state. We will treat these records as our privacy units, since the individual farm-level records are not publicly available. Still, a small number of records contribute to the majority of the total cattle inventory in any particular state or AD, making the methodology applicable. We consider the queries in

Table 5 under different privacy loss allocations and splitting thresholds shown in the figures. The P-PRzCDP queries are answered using Algorithm 1 and the ρ-zCDP queries are answered using the Gaussian mechanism from Definition 4.

Table 5: CIS Queries

Geographies	Query	Formalism
State	SUM(CattleInventory)	P-PRzCDP
State	AVG(PastureRent)	ρ-zCDP
State x AD	SUM(CattleInventory)	P-PRzCDP
State x AD	AVG(PastureRent)	ρ-zCDP

The final dataset is a proposed use case involving summations over skewed data: the County Business Patterns (CBP) dataset, published by the U.S. Census Bureau. We use simulated data provided by the U.S. Census Bureau to demonstrate PRzCDP methodology on these simulated records. Each row in our tabular data represents an "establishment", or one separate unit of a business; "firms" represent one or more establishments that operate as a single business venture. Our goal is to release the following information about groups of establishments:

- ESTAB: A count of the number of establishments.
- PAYANN: A sum of annual payrolls of establishments.
- PAYQTR1: A sum of first quarter payrolls of establishments.
- EMP: A sum over employee size of establishments.

The groups of establishments correspond to different geographic areas (such as counties, ZIP codes, or congressional districts) and different industry classifications using the North American Industry Classification System (NAICS) codes (such as finance, real estate, agriculture, etc.). For the purposes of this experiment, we consider the subset of all county-level queries at every possible NAICS classification level with no cross-tabulations; moreover, we limit our evaluations to only those queries with 100 or more establishments.

Table 6: CBP Queries

Geographies	Query	Formalism
County x NAICS*	COUNT(ESTAB)	ρ-zCDP
County x NAICS*	SUM(EMP)	P-PRzCDP
County x NAICS*	SUM(PAYANN)	P-PRzCDP
County x NAICS*	SUM(PAYQTR1)	P-PRzCDP

6.2 Global zCDP on Skewed data

First, we show how a theoretical analysis of the ρ-zCDP Gaussian mechanism for sums on heavy-tailed random variables fails to yield reasonable trade-offs between privacy and utility. Specific to our simulation study, we consider the theoretical mean-square error (MSE) of estimators truncated with high probability from heavy tails. In Figure 1, we fix $n = 1000$ and plot the theoretical mean-square error (MSE) over the sum's expected value as a measure of "noise-to-signal" on the y-axis. We show how this ratio varies with different sensitivities Δ on the x-axis, privacy losses ρ, and tail weights α. Note that in every case, privacy-preserving noise exceeds the sum's expected value by orders of magnitude. For each configuration, we additionally calculate the optimal Δ for given α and ρ values which minimize the ratio (shown as the vertical red lines in the subplots). As expected, the optimal Δ value to minimize MSE increases as α decreases and as ρ increases; however, even at these optimal Δ values, the errors are prohibitively large. Recall from [7] that Gaussian noise is tight for zCDP summation queries, meaning any ρ-zCDP mechanism requires noise with variance at least $\Omega(\Delta/\rho^2)$, *i.e.,* as a function of the volume of the sum query space. So even for modest tail weights, the cost of ensuring each record lies in a bounded domain makes it near-impossible to simultaneously maintain modest global privacy losses and MSE guarantees.

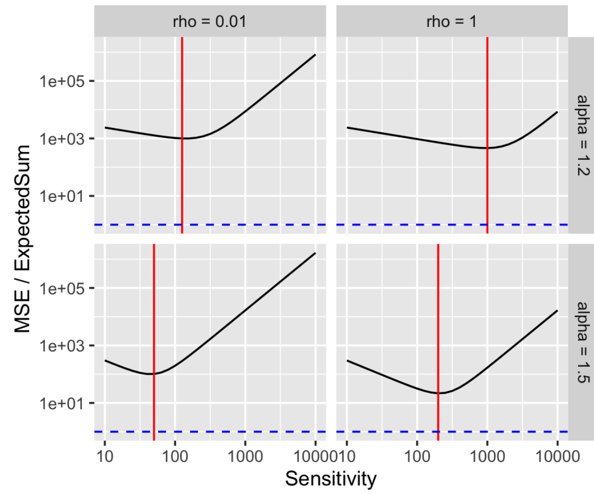

Figure 1: Theoretical MSE over the expected query value for global ρ-zCDP mechanisms with different global sensitivities Δ, privacy loss budgets ρ, and different tail parameters α for $n = 1000$. Optimal Δ for minimizing MSE given ρ and α shown in red. Blue dashed line at 1, for reference.

6.3 PRzCDP privacy-utility trade-offs with univariate splitting

Next, we show how using Algorithm 1 in conjunction with the Gaussian mechanism offers significant improvements to utility with a cost to privacy loss that only affects a small number of units. In Figure 2, we use our proposed method to answer queries on the workloads with different unit splitting thresholds (STs) for one variable (HT1 for SIM, CattleInventory for CIS). The left set of subplots show the workload AREs (y-axis) at different STs; as ST decreases, the proportion of records that are split (for which policy loss is greater than ρ) increases, shown on the x-axis. To simplify, utility increases (y-axis distributions shift downward) as privacy loss increases (x-axis boxplots shift to the right). The plots show that ARE improves significantly, while policy loss for most records remains the same as the global zCDP counterpart. For example, at $\rho = 1$ for the simulated data, we can achieve a median 10% ARE across queries while ensuring less than 1% of records have policy loss greater than $\rho = 1$.

On the right-hand side of Figure 2, we show a more detailed view of the policy loss functions by visualizing their empirical CDFs: namely, for any one unit splitting configuration, what proportion of records (y-axis) have policy losses less than a particular value (x-axis)? We show this for different STs and ρ. As ρ increases, the CDFs shift to the right, as expected since less noise injection increases policy loss uniformly across records. Larger STs correspond to more conservative unit splitting schemes, ensuring that a greater proportion of records have the smallest possible policy loss. As ST decreases, the policy loss grows more rapidly for larger units, which are split more frequently. These plots demonstrate how ρ toggles the privacy-utility trade-off for all records, whereas ST toggles how fast the policy loss grows as records become more skewed.

6.4 End-to-end example: County Business Patterns dataset

We now turn towards a more complex, realistic application of our methodology to CBP. The query workload is described in Table 6; answering these queries requires leveraging more features of our proposed framework. First, we consider multivariate unit splitting as a function of multiple attributes per-record. Second, we combine zCDP with PRzCDP queries. Third, we use both sequential and parallel composition simultaneously to answer queries about the full workload.

We consider two different algorithmic approaches for answering the CBP query workload. First, we consider zCDP mechanisms using sensitivities defined by three possible sets of "top-codes" in Table 7, which we name "Conservative", "Moderate", and "Aggressive", in decreasing order. Second, we consider PRzCDP based on splitting schemes listed in Table 8, which we name "Conservative", "Moderate", and "Median". The three schemes are listed in increasing split cardinality order.

Table 7: Top-code scheme description.

Top-code scheme name	Attribute	Value
Conservative	EMP	10^3
	PAYANN	10^5
	PAYQTR1	$2.5 * 10^4$
Moderate	EMP	$3 * 10^2$
	PAYANN	10^4
	PAYQTR1	$2.5 * 10^3$
Aggressive	EMP	10^2
	PAYANN	10^3
	PAYQTR1	$2.5 * 10^2$

In Figure 3 we plot the ARE of each query for the top-coding algorithm and the establishment splitting algorithm, respectively. The results are aggregated by attribute, total privacy loss budget, NAICS level, and algorithmic configuration. The green and red dashed lines mark the 5% and 20% ARE thresholds, respectively, representing example fitness-for-use goals. First, we expect the relative errors for counting attributes (*i.e.,* ESTAB and FIRM) to

Table 8: Establishment splitting thresholds for three different splitting schemes and their associated percentiles of score in the simulated CBP data.

SchemeName	SplitAttribute	SplitThreshold	PctScore
Conservative	PAYANN	10000	99%
	PAYQTR1	2500	99%
	EMP	100	97%
Moderate	PAYANN	500	79%
	PAYQTR1	125	80%
	EMP	5	66%
Median	PAYANN	104	50%
	PAYQTR1	24	50%
	EMP	2	47%

have the same distribution for either algorithm, as they are unaffected by establishment splitting. However, when we look at the magnitude attributes (EMP, PAYANN, and PAYQTR1), none of the top-coding schemes on the left come close to providing reasonable AREs, since the majority of the box plot masses for these queries are above the dashed green line. Alternatively, on the right, we see that establishment splitting provides far smaller relative errors, even for more granular queries at finer NAICS levels (although the relative errors increase as the NAICS level increases, as expected). Like in Figure 2 we see that the moderate splitting scheme, which has larger splitting thresholds, results in larger relative error than the median splitting scheme. While the moderate splitting scheme only satisfies the fitness for use goals for the ESTAB and EMP queries (at NAICS level 2) the median splitting scheme satisfies the fitness for use in all cases.

Similarly, in Figure 5, we plot the policy loss CDFs for each splitting scheme, which can be interpreted similarly to the policy loss CDFs in Figure 2 with a few differences worth highlighting. First, by PRzCDP's group composition properties, we can extend the results from the left column of establishment subplots to the right column of firm subplots. Since the majority of establishments in the simulated data have a unique ID, the plots look very similar; however, the firm-level CDFs sit slightly below the establishment level CDFs. This demonstrates how establishment-level guarantees are conferred to firms. Second, we additionally calculate the *realized* privacy losses for each establishment and firm. Specifically, this calculates the log-max divergence between the establishment splitting outputs using the specific simulated CBP dataset with and without the establishment (or firm) of interest. By construction, the realized loss is always less than the policy loss, so the blue CDF line will always be to the left of the orange policy loss line. First, we observe that for the majority of establishments, the realized privacy loss is significantly lower than the policy upper bound. Moreover, because this gap depends on the number of queries answered in the workload, we can reasonably expect this gap to be larger when considering the entire CBP workload, not just county-level queries. Next, we observe that, as the splitting thresholds decrease, the gap between the realized privacy loss and the policy upper bound decreases.

Finally, instead of considering the realized relative errors, we ask: what is the *smallest* possible policy function which ensures we

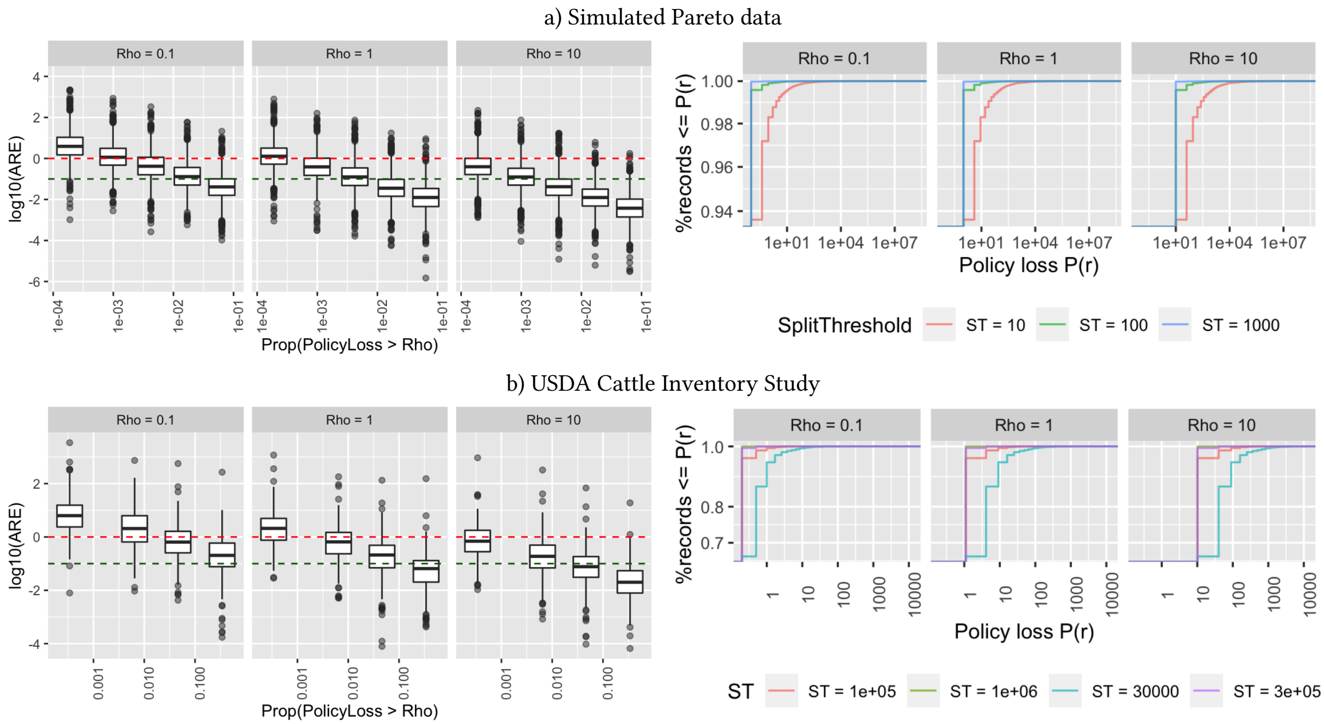

Figure 2: (Left) distribution of ARE over workload queries (y-axis) by proportion of records with policy loss greater than ρ (x-axis) *i.e.,* as the splitting threshold decreases. (Right) Empirical CDFs of policy loss, *i.e.,* proportion of observed records (y-axis) with policy loss bounded by $P(r)$ (x-axis). Columnar subplots show different levels of minimum policy loss ρ. Red line represents 100% ARE and green line represents 10% ARE.

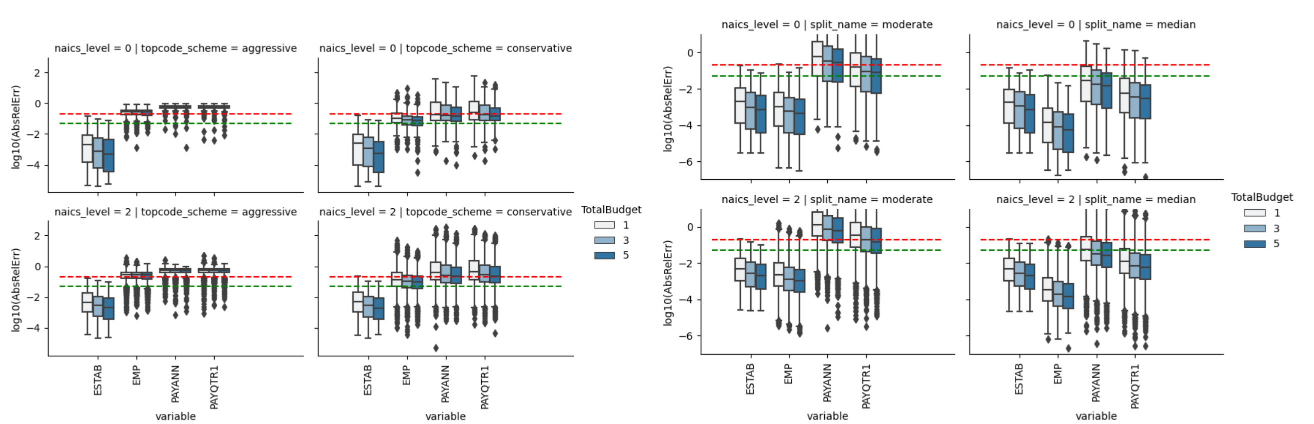

Figure 3: AREs for the CBP query workload using topcoding and zCDP (left) versus using unit splitting (right) for different NAICS levels (rows) and splitting schemes (columns). Red line represents 20% ARE and green line represents 5% ARE.

reach a particular fitness-for-use goal? Specifically, we calculate the smallest policy loss function for each entity where we assume that for each query in the county workload, we have at theoretical query relative error of less than δ with probability at least 95%. We plot the implied policy loss CDFs in Figure 4. As expected, we require smaller policy losses for the majority of establishments as the splitting thresholds get smaller and smaller, since we are incurring larger privacy losses for larger establishments. Additionally, as expected, when δ increases, the distribution of the minimum policy loss subsequently decreases (*i.e.,* the CDFs are shifted to the left). All this demonstrates that with these splitting schemes, fitness-for-use goals are more feasible than under the traditional top-coding assessment.

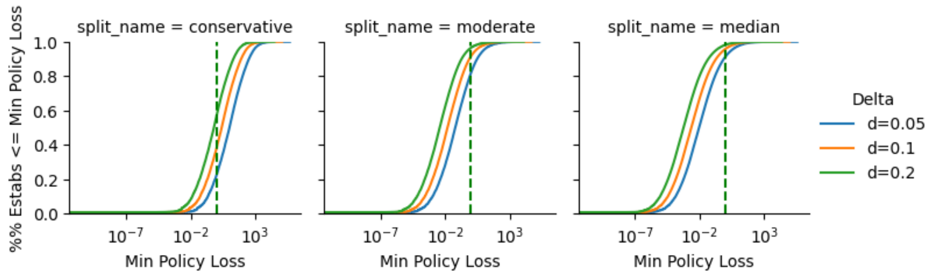

Figure 4: Theoretical minimum policy function CDFs to achieve different fitness-for-use goals on 95% of the COUNTY by NAICS code query workload. The green dashed line represents the total unsplit privacy loss budget of 1.

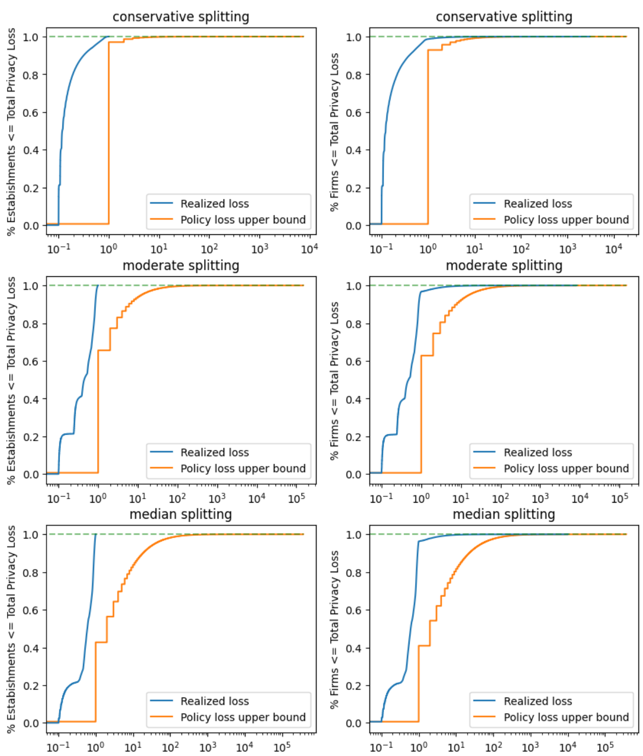

Figure 5: CBP policy losses grouped by establishment (left) and firm (right)

7 CONCLUSION

To summarize, we introduced PRzCDP to transparently encode dependencies between per-record privacy loss and confidential records. This relaxation of traditional DP notions helps answer SQL-style queries over skewed data, where approaches like zCDP may fail to offer reasonable privacy-utility trade-offs. By making the policy function public, we offer a new way of describing privacy loss in cases where a small minority of records pose exorbitant privacy risks that aren't representative of the entire dataset. Such policy functions are particularly useful when the unit of privacy analysis is not an individual person, but a group of people in a business establishment or other organization. We additionally offer a way of indirectly setting the policy function through unit splitting, a pre-processing step that composes with DP algorithms to provide PRzCDP guarantees by construction. Our experiments applying this technique to simulated and real data demonstrate how PRzCDP can better answer realistic SQL-style query workloads on skewed data without relying on zCDP's worst-case analysis.

Note that choosing how to implement PRzCDP using our algorithms requires subjective policy choices about how to set privacy loss, just like in standard DP. Because our proposed unit splitting methods involve two such choices, one for the baseline privacy loss and one for how to implement splitting, there is no one "optimal" policy function for a particular data utility goal. That is, fixed data utility goals for our queries can be achieved by either increasing the baseline privacy loss or by more aggressively splitting records, producing policy loss functions that necessarily differ on smaller or larger records.

Future work beyond the scope of the article could more formally characterize the semantic guarantees offered by PRzCDP. Techniques like unit splitting intrinsically leak more information about confidential records when records are split with finer granularity. Understanding the kinds of queries that could be leveraged to learn confidential information via the policy function require further investigation.

Future work could explore different techniques for choosing how privacy loss scales with record values. While we considered quadratic dependence between record values and policy loss, using ϵ-DP style semantics could yield linear dependence instead.

ACKNOWLEDGMENTS

We would like to thank Margaret Beckom, Anthony Caruso, William Davie Jr, Ian Schmutte, and Brian Finley from the U.S. Census Bureau and Zach Terner from MITRE for their valuable insight and feedback.

REFERENCES

[1] 83rd United States Congress. 1954. Title 13.
[2] 99th United States Congress. 1986. Title 26 / Internal Revenue Code of 1986.
[3] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi (Eds.). ACM, 308–318. https://doi.org/10.1145/2976749.2978318
[4] Mohammad Alaggan, Sébastien Gambs, and Anne-Marie Kermarrec. 2016. Heterogeneous Differential Privacy. *Journal of Privacy and Confidentiality* 7, 2 (2016),

127–158.
[5] Hilal Asi, Jonathan Ullman, and Lydia Zakynthinou. 2023. From Robustness to Privacy and Back. arXiv:2302.01855 [cs.LG]
[6] Skye Berghel, Philip Bohannon, Damien Desfontaines, Charles Estes, Sam Haney, Luke Hartman, Michael Hay, Ashwin Machanavajjhala, Tom Magerlein, Gerome Miklau, Amritha Pai, William Sexton, and Ruchit Shrestha. 2022. Tumult Analytics: a robust, easy-to-use, scalable, and expressive framework for differential privacy. arXiv:2212.04133 [cs.CR]
[7] Mark Bun and Thomas Steinke. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*. Springer, 635–658.
[8] U.S. Census Bureau. 2023. County Business Patterns. https://www.census.gov/programs-surveys/cbp.html
[9] U. S. Census Bureau. 2023. US Census Bureau, County Business Patterns: Demonstration Tables for New Differential Privacy Methodology for Disclosure Avoidance. https://www.census.gov/topics/business-economy/disclosure/data/tables/cbp-privacy-demonstration-tables.html
[10] Damien Desfontaines and Balázs Pejó. 2020. Sok: differential privacies. *Proceedings on privacy enhancing technologies* 2020, 2 (2020), 288–313.
[11] Damien Desfontaines, James Voss, Bryant Gipson, and Chinmoy Mandayam. 2022. Differentially private partition selection. *Proceedings on Privacy Enhancing Technologies* 1 (2022), 339–352.
[12] Cynthia Dwork and Jing Lei. 2009. Differential Privacy and Robust Statistics. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing* (Bethesda, MD, USA) *(STOC '09)*. Association for Computing Machinery, New York, NY, USA, 371–380. https://doi.org/10.1145/1536414.1536466
[13] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*. Springer, 265–284.
[14] Cynthia Dwork and Aaron Roth. 2013. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3-4 (2013), 211–407. https://doi.org/10.1561/0400000042
[15] Hamid Ebadi, David Sands, and Gerardo Schneider. 2015. Differential privacy: Now it's getting personal. *Acm Sigplan Notices* 50, 1 (2015), 69–81. Publisher: ACM New York, NY, USA.
[16] Timothy Evans, Laura Zayatz, and John Slanta. 1996. Using noise for disclosure limitation of establishment tabular data. In *Proceedings of the Annual Research Conference, US Bureau of the Census, Washington, DC*, Vol. 20233. 65–86.
[17] Vitaly Feldman and Tijana Zrnic. 2021. Individual privacy accounting via a renyi filter. *Advances in Neural Information Processing Systems* 34 (2021).
[18] Arpita Ghosh and Aaron Roth. 2015. Selling privacy at auction. *Games and Economic Behavior* 91 (2015), 334–346. Publisher: Elsevier.
[19] Statistics of Income (SOI) Division Internal Revenue Service. 2023. SOI Tax Stats - Products, Publications, and Papers. https://www.irs.gov/statistics/soi-tax-stats-products-publications-and-papers
[20] Zach Jorgensen, Ting Yu, and Graham Cormode. 2015. Conservative or liberal? Personalized differential privacy. In *2015 IEEE 31St international conference on data engineering*. IEEE, 1023–1034.
[21] Ios Kotsogiannis, Stelios Doudalis, Sam Haney, Ashwin Machanavajjhala, and Sharad Mehrotra. 2020. One-sided Differential Privacy. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. 493–504. https://doi.org/10.1109/ICDE48307.2020.00049
[22] Chao Li, Gerome Miklau, Michael Hay, Andrew McGregor, and Vibhor Rastogi. 2015. The matrix mechanism: optimizing linear counting queries under differential privacy. *The VLDB journal* 24, 6 (2015), 757–781. Publisher: Springer.
[23] Xiyang Liu, Weihao Kong, and Sewoong Oh. 2021. Differential privacy and robust statistics in high dimensions. arXiv:2111.06578 [math.ST]
[24] Meghan O'Malley and Lawrence R. Ernst. 2007. Practical Considerations in Applying the pq-Rule for Primary Disclosure Suppressions December.
[25] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908* (2018).
[26] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable Private Learning with PATE. *arXiv:1802.08908 [cs, stat]* (Feb. 2018). http://arxiv.org/abs/1802.08908 arXiv: 1802.08908.
[27] Rachel Redberg and Yu-Xiang Wang. 2021. Privately publishable per-instance privacy. *Advances in Neural Information Processing Systems* 34 (2021), 17335–17346.
[28] National Agricultural Statistics Service. 2023. Census of Agriculture. https://www.nass.usda.gov/AgCensus/
[29] Yu-Xiang Wang. 2019. Per-instance differential privacy. *Journal of Privacy and Confidentiality* 9, 1 (2019).
[30] Justin Whitehouse, Aaditya Ramdas, Ryan Rogers, and Zhiwei Steven Wu. 2022. Fully adaptive composition in differential privacy. *arXiv preprint arXiv:2203.05481* (2022).