Differentially Private Explanations for Clusters

Amir Gilad
Hebrew University

Tova Milo
Tel Aviv University

Kathy Razmadze
Tel Aviv University

Ron Zadicario
Tel Aviv University

Abstract

The dire need to protect sensitive data has led to various flavors of privacy definitions. Among these, Differential privacy (DP) is considered one of the most rigorous and secure notions of privacy, enabling data analysis while preserving the privacy of data contributors. One of the fundamental tasks of data analysis is clustering , which is meant to unravel hidden patterns within complex datasets. However, interpreting clustering results poses significant challenges, and often necessitates an extensive analytical process. Interpreting clustering results under DP is even more challenging, as analysts are provided with noisy responses to queries, and longer, manual exploration sessions require additional noise to meet privacy constraints. While increasing attention has been given to clustering explanation frameworks that aim at assisting analysts by automatically uncovering the characteristics of each cluster, such frameworks may also disclose sensitive information within the dataset, leading to a breach in privacy.

To address these challenges, we present DPClustX, a framework that provides explanations for black-box clustering results while satisfying DP. DPClustX takes as input the sensitive dataset alongside privately computed clustering labels, and outputs a global explanation, emphasizing prominent characteristics of each cluster while guaranteeing DP. We perform an extensive experimental analysis of DPClustX on real data, showing that it provides insightful and accurate explanations even under tight privacy constraints.

1 Introduction

Sensitive data collection has never been more prevalent, from fitness trackers [22] and financial institutions [57], to healthcare and insurance facilities [2]. Such data require special treatment to allow for its safe and secure usage without exposing individuals to harm due to the presence of their information in the data. Differential Privacy (DP) [16, 19] has emerged as the gold standard for handling such issues. The crux of DP is to allow sensitive data to be used while bounding the privacy leakage and to offer utility bounds on the obtained results. Indeed, DP has been adopted by multiple companies [3, 17] and governmental organizations [12, 20, 67].

Among the many operations in data analysis, *clustering* plays a pivotal role in uncovering hidden patterns and providing actionable insights from data. To ensure privacy, researchers have worked extensively to adapt clustering techniques to comply with DP [25, 27, 62, 64]. Under DP, the true clusters must be obfuscated and slightly distorted to prevent individual information leakage, often at the cost of accuracy.

Clustering algorithms often operate as black boxes, offering little insight into the reasoning behind their results. Hence, it is challenging for end users to comprehend this reasoning, or draw meaningful conclusions from the results based on domain knowledge [32]. The additional requirement to adhere to strict DP standards further amplifies this complexity. To account for this, previous work has focused on *explaining non-private clustering algorithms* [21, 23, 45, 51, 69] and aimed to provide succinct and interpretable summaries of the properties of each cluster by showing how it varies from the other clusters. When considering privacy, it is likely that lack of access to the data is accompanied by lack of access to the clustering algorithm, requiring an approach that is suitable for black-box clustering algorithms.

Works that focus on providing explanations for black-box clustering results are often histogram-based approaches [8, 11], which is a popular form of explanation in other settings as well, including visualization recommendations [38, 42, 72] and automated insight extraction from data [5, 66]. With these approaches, users get a *histogram for each cluster* that focuses on a specific attribute and graphically shows how the data associated with the cluster differs from the rest of the data. Yet, the approaches that generate such explanations cannot be directly used in the DP setting.

First, these approaches choose histograms based on quality functions, such as interestingness [8, 11, 30, 61], sufficiency [8, 10], and diversity [8, 75], which require significant distortion under DP. That is, the required noise scale is large relative to their range, making it impractical to obtain a reliable explanation, as the ranking of explanations by quality is unlikely to be preserved after adding noise. Second, existing methods generate all explanation options before choosing the ones with the highest scores. However, in the DP setting, this strategy quickly becomes infeasible because it requires producing private histograms for every attribute and cluster, necessitating excessive distortion to ensure DP compliance. Third, evaluating the quality of the explanation based on noisy histograms introduces excessive noise, as each bin is injected with independent noise, which accumulates and leads to an inaccurate quality assessment.

In light of this, *we propose* DPClustX, *a novel framework for generating histogram-based explanations of black-box clustering results under DP.* DPClustX is inspired by previous work [8, 11] and addresses the above limitations as follows. (1) We first prove that previous approaches cannot be applied directly, as existing quality functions for histogram-based explanations would have to be significantly distorted to satisfy DP. Then, we develop DP tailored variants that enable the generation of high-quality and privacy-preserving explanations. (2) To minimize privacy costs in the explanation selection process, we evaluate the *attribute* quality directly, privately selecting high-scoring explanation attributes based on the sensitive dataset with our novel quality function. We then generate noisy histograms exclusively for the selected attributes, leveraging previous work on DP histograms [13, 18]. However, each explanation corresponds to an assignment of histograms to clusters, and its quality is evaluated globally across all clusters. Therefore, the search space

age	gender	lab_proc	diag_1	...	cluster
[60, 70)	Female	[40, 50)	Circulatory	...	1
[60, 70)	Female	[0, 10)	Diabetes	...	2
[70, 80)	Male	[40, 50)	Injury	...	2
...	...	...	...	...	...

Figure 1: Example of the Diabetes dataset.

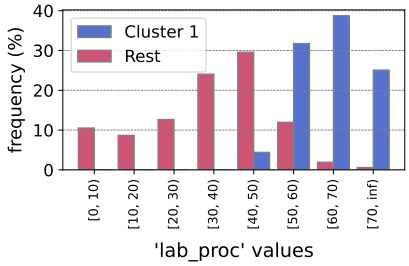
for the best clustering explanation, which encompasses all possible assignments, is considerably large.

Hence, (3) we adapt techniques from prior work [8] and develop a DP mechanism that constructs a smaller *candidate set* for each cluster, from which the clustering explanation is privately derived. To this end, we adapt the idea of a single-cluster score function which is used to rank the attributes for each cluster by their explanation quality for that cluster [8] and tailor it to the DP setting. This function is adapted to DP so that the noise added to it will still render the results useful in filtering attributes. However, to naively select the top explanation attributes for each cluster, one would need to apply a DP mechanism for privately selecting a single high quality attribute multiple times, with each iteration requiring a re-calculation of noisy scores for all remaining candidates. Instead, we utilize the One-shot Top-k mechanism [15], which computes the noisy scores **once** and then releases the top k candidates. This further reduces execution times, thereby enhancing the interactive user experience. An illustration of the DPClustX framework, summarizing these steps, is given in Figure 3. Our experimental study confirms that DPClustX generates insightful and accurate explanations even under tight privacy constraints, demonstrating robustness to attribute correlations, variations in the number of clusters, and maintaining reasonable execution times

Example 1.1. Consider an analyst working with the Diabetes dataset [7] (a subset of the columns is illustrated in Figure 1), aiming to identify groups of patients with similar medical records using the DP-k-means algorithm [64]. While DP-k-means privately provides cluster centers (see the last column in Figure 1 associating each tuple with a cluster), it does not offer additional insights about the clusters. Instead of exhausting the privacy budget through a manual EDA session, the analyst employs DPClustX to generate histogram-based explanations. These explanations reveal significant patterns, such as the fact that Cluster 1 consists primarily of individuals who underwent a higher number of lab procedures, as shown in Figure 2a. Comparing the cluster distribution of `lab_proc` with the remaining data, we see that values outside Cluster 1 are concentrated in the middle and lower ranges, peaking at $[40, 50)$. In contrast, Cluster 1 values are concentrated in the upper range, peaking at $[60, 70)$. This suggests that the clustering algorithm groups individuals with a higher number of lab procedures in Cluster 1. For simplicity, we have attached an LLM generated textual description of the histogram in Figure 2b.

Our contributions. Our contributions are summarized as follows:

- We formulate the problem of finding a high scoring privacy-preserving histogram-based explanation for black-box clustering methods, with the main challenge being the private selection of high-quality attributes (Section 3).
- We design quality functions for histogram-based explanations inspired by the notions of interestingness, sufficiency, and diversity that are adapted for the DP setting (Section 4).



(a) Part of a histogram-based explanation for the Diabetes dataset. The selected attribute, `lab_proc`, specifies the number of lab procedures an individual underwent.

Textual description: The '`lab_proc`' column values differ significantly. Values outside Cluster 1 are concentrated in the lower and mid-range (**85**% below **50**), while Cluster 1 contains mainly higher values (**95**% above **50**).

(b) Textual description of the histogram explanation in Figure 2a.

Figure 2: Cluster explanation with its textual description.

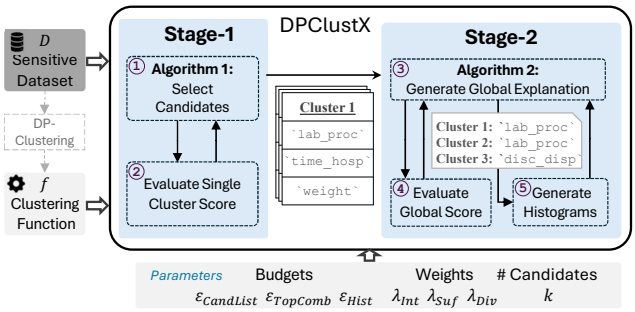


Figure 3: System architecture of DPClustX. Algorithm 1 selects candidate attributes for each cluster (1) using the single-cluster score function (2). Algorithm 2 selects a high-quality attribute combination from these candidates (3) using the global score function (4), and generates noisy histograms only for the selected attributes (5).

- We develop DPClustX, the first framework designed to generate histogram-based explanations for clustering results under DP (Section 5), equipped with formal guarantees.
- We provide a comprehensive experimental study demonstrating the effectiveness of DPClustX, showing that its explanations align closely with non-private explanations even under a strict privacy budget (Section 6).

2 Preliminaries

In this section, we introduce our notations, and review existing concepts from DP used in this work.

Data. Given a single table schema $R(A_1, \ldots, A_d)$, R is a relation name and $\mathcal{A} = \{A_1, \ldots, A_d\}$ denotes the set of attributes. Each attribute A_i has discrete, finite, and data-independent domain $\text{dom}(A_i)$. The full domain of R is $\text{dom}(R) = \text{dom}(A_1) \times \cdots \times \text{dom}(A_d)$. An instance (dataset) D of a relation R is a bag

(multiset) whose elements are tuples in $\mathrm{dom}(R)$. We let $\mathcal{D}$ denote the set of all datasets of the relation R, i.e., $\mathcal{D} = \{D \mid D \subseteq \mathrm{dom}(A_1) \times \cdots \times \mathrm{dom}(A_d), |D| < \infty\}$. The number of tuples in D is denoted as $|D|$. $\pi_A(D)$ is the projection of D onto the attribute A. For a dataset D and an attribute $A \in \mathcal{A}$, we let $\mathrm{dom}_D(A)$ denote the *active domain* of A with respect to D, i.e, the subset of $\mathrm{dom}(A)$ of values appearing in $\pi_A(D)$ at least once. we denote by $cnt_{A=a}(D)$ the number of occurrences of value a in $\pi_A(D)$, and by $h_A(D)$ the histogram of the dataset D over attribute A. That is, $h_A(D)[a] = cnt_{A=a}(D)$ for any $a \in \mathrm{dom}(A)$. For visualizations, we use normalized histograms, where each count is replaced by its proportion.

Example 2.1. Consider the Diabetes dataset, illustrated in Figure 1, which contains 47 attributes, including `age`, `gender`, `lab_proc`, and `diag_1`. The domain $\mathrm{dom}(\texttt{lab\_proc})$ comprises 8 values, each representing a range of lab procedure counts. Figure 2a illustrates two histograms derived from the Diabetes dataset. For instance, the blue bars represent the histogram $h_{\texttt{lab\_proc}}(D_1)$, where D_1 is the cluster explained. Here, $\mathrm{dom}_{D_1}(\texttt{lab\_proc})$ comprises 4 values, as no tuple in D_1 has `lab_proc` < 40.

Histogram-based explanation (HBE). A clustering of D is a partition into disjoint subsets $\{D_c \subseteq D \mid c \in C\}$, each assigned with a cluster label $c \in C$. That is, $D_c \cap D_{c'} = \emptyset$ for $c \neq c'$ and $\bigcup_{c \in C} D_c = D$, An single-cluster HBE candidate consists of two histograms on a specified attribute: the histogram of the cluster values, and the histogram of the values outside the cluster (illustrated in Figure 2a). A global HBE candidate is a set of single-cluster HBE candidates, with a candidate for each cluster. Formally,

Definition 2.2 (Single cluster HBE candidate [8, 11]). For a dataset D and a cluster label $c \in C$, a single-cluster HBE candidate e_c is a tuple $(c, A, h_A(D \setminus D_c), h_A(D_c))$ where $A \in \mathcal{A}$.

Example 2.3. Figure 2a illustrates a single cluster HBE candidate for Cluster 1 in the Diabetes dataset [7], using the `lab_proc` attribute. The (normalized) histogram of the cluster values is shown in blue, and for the remaining data in red.

Definition 2.4 (Global HBE candidate [8]). Given D, clustered into $\{D_c \mid c \in C\}$, a global HBE candidate is a set $\{e_c \mid c \in C\}$, where e_c is a single-cluster HBE candidate for D_c.

An HBE should capture unique patterns that characterize each cluster, and distinguish it from other clusters. Building upon [10, 11, 30, 75], the work of [8] proposed the aspects of *interestingness*, *sufficiency* and *diversity* for evaluating the quality of an HBE. Briefly, interestingness is quantified as the distributional shift between the inner cluster distribution and the full dataset in a given attribute. Sufficiency represents the extent to which the HBE captures the patterns of the underlying clustering, and diversity measures the overall distinctiveness among explanations. We discuss these measures further in Section 4.

2.1 Differential Privacy

We next review preliminary background from the DP literature used in this work. DP ensures that the distribution of outputs does not significantly change when the algorithm is applied to any two *neighboring databases*.

Definition 2.5 (Neighboring Datasets [16]). Two datasets D and D' are called *neighboring* (denoted $D \sim D'$) if D' can be obtained from D by removing or adding one tuple.

Definition 2.6 (ε-Differential Privacy (DP) [16, 19]). A randomized mechanism $\mathcal{M}$ is said to satisfy ε-DP if for any neighboring datasets $D \sim D'$ and any set of possible outputs $S \subseteq Range(\mathcal{M})$,

$$\mathbf{Pr}[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \cdot \mathbf{Pr}[\mathcal{M}(D') \in S],$$

where the probability is over the randomness of $\mathcal{M}$.

Proposition 2.7. *The following holds for DP [18, 19]:*

- *Sequential Composition: Let $\mathcal{M}_1 : \mathcal{D} \to \mathcal{Y}$ and $\mathcal{M}_2 : \mathcal{D} \times \mathcal{Y} \to \mathcal{Z}$. Suppose $\mathcal{M}_1$ satisfies ε_1-DP and for every $y \in \mathcal{Y}$, $\mathcal{M}_2(\cdot, y)$ satisfies ε_2-DP (as a function of its first input). Define $\mathcal{M}_3 : \mathcal{D} \to \mathcal{Z}$ by $\mathcal{M}_3(D) = \mathcal{M}_2(D, \mathcal{M}_1(D))$. Then, $\mathcal{M}_3$ satisfies $(\varepsilon_1 + \varepsilon_2)$-DP*
- *Parallel Composition: Suppose $\mathcal{M}_1 : \mathcal{D} \to \mathcal{Y}$ satisfies ε_1-DP and $\mathcal{M}_2 : \mathcal{D} \to \mathcal{Z}$ satisfies ε_2-DP. Let $\mathcal{D}_1, \mathcal{D}_2 \subseteq \mathrm{dom}(R)$ be disjoint subsets of the input domain. Define $\mathcal{M}' : \mathcal{D} \to \mathcal{Y} \times \mathcal{Z}$ by $\mathcal{M}'(D) = (\mathcal{M}_1(D \cap \mathcal{D}_1), \mathcal{M}_2(D \cap \mathcal{D}_2))$. Then, $\mathcal{M}'$ satisfies $\max\{\varepsilon_1, \varepsilon_2\}$-DP.*
- *Post-processing: Suppose $\mathcal{M} : \mathcal{D} \to \mathcal{Y}$ satisfies ε-DP. Then, for any function $g : \mathcal{Y} \to \mathcal{Z}$ (deterministic or randomized), The mechanism defined by $g(\mathcal{M}(D))$ satisfies ε-DP.*

To quantify the noise that has to be injected in order for mechanisms to satisfy DP, we first define then notion of sensitivity,

Definition 2.8 (Sensitivity [19]). For a set of candidates $\mathcal{R}$, Let $q : \mathcal{D} \times \mathcal{R} \to \mathbb{R}$ be a quality function that given $D \in \mathcal{D}$, defines a score for every $r \in \mathcal{R}$. The sensitivity of q is

$$\Delta_q = \sup_{r \in \mathcal{R}} \sup_{D \sim D'} \left| q(D) - q(D') \right|$$

The exponential mechanism [47] is a DP primitive for privately releasing the top item from a set of candidates with respect to a quality function that depends on the sensitive dataset. The mechanism injects noise to the selection process, and outputs each candidate with probability proportional to its score.

Definition 2.9 (The Exponential Mechanism (EM) [47]). Given $D \in \mathcal{D}$, a set of candidates $\mathcal{R}$, a quality function $q : \mathcal{D} \times \mathcal{R} \to \mathbb{R}$, and a privacy parameter ε, the exponential mechanism $\mathcal{M}_E$ selects and outputs $r \in \mathcal{R}$ with probability proportional to $\exp\left(\frac{\varepsilon \cdot q(D,r)}{2\Delta_q}\right)$

Theorem 2.10 ([47]). *The exponential mechanism satisfies ε-DP. Moreover,*

$$\mathbf{Pr}\left[\mathcal{M}_E(D, \mathcal{R}, q, \varepsilon) \leq \max_{r \in \mathcal{R}} q(D, r) - \frac{2\Delta_q}{\varepsilon}(\ln|\mathcal{R}| + t)\right] \leq e^{-t}.$$

The One-shot Top-k Mechanism. In DPClustX, we utilize the *One-shot Top-k mechanism* [15] to privately select top-k explanation attributes for each cluster. This mechanism adds independent Gumbel noise[1] to each of the true scores with scale $\sigma = 2\Delta_{\mathrm{Score}} \cdot k/\varepsilon$, where Δ_{Score} is the sensitivity of the score function (Definition 2.8). Then, it reorders all the candidates in a descending order by their noisy scores, and outputs the first k candidates. It satisfies ε-DP since it is identical to iteratively applying the EM k times [15],

[1]The CDF of the Gumbel distribution Gumbel (σ) is $F(z) = \exp(-\exp(-z/\sigma))$.

where each satisfies ε/k-DP. Therefore, by sequential composition (Proposition 2.7) it satisfies overall ε-DP.

Differentially private histograms. DP mechanisms for computing private histograms are well-studied (e.g., [4, 29, 40, 59, 74]). As DPClustX can be instantiated with any DP histogram generation mechanism, we denote it as $\mathcal{M}_{hist}(\pi_A(D), \varepsilon_{hist})$. It takes the column of interest $\pi_A(D)$ and a privacy budget ε_{hist}, and outputs a histogram of noisy counts $\widetilde{h_A}(D)$ over $\text{dom}(A)$, while satisfying ε_{hist}-DP. Such mechanisms are accompanied by utility bounds, enabling accuracy control by translating accuracy requirements into the required privacy budget.

Differentially private clustering. DP clustering has been extensively studied in the DP literature, with prominent approaches aiming to release cluster centers from a sensitive dataset D while preserving DP (e.g., [25, 27, 62, 64]). In the non-private black-box clustering explanation setting (e.g. [69]), a clustering is typically modeled by a function $f : D \to C$, assigning each tuple in D a cluster label $c \in C$. However, this modeling is inherently unsuitable for the output of DP clustering, which requires any possible output (i.e., a clustering function) to occur with similar probability for any two different but neighboring datasets. Instead, we model the output of a DP clustering algorithm as function $f : \text{dom}(R) \to C$. This definition encompasses DP mechanisms that release centers, as the fixed centers define a cluster assignment for any tuple in $\text{dom}(R)$, while also accommodating other approaches, such as user-defined predicates or future clustering techniques.

3 Problem Statement

We begin by describing the privacy setup. Subsequently, we present the formal problem definitions.

An HBE mechanism (Definition 3.1) preserves privacy if for any clustering function $f : \text{dom}(R) \to C$, its distribution of outputs does not change much when we add a single tuple to the dataset. Note that we assume only *black-box* access to f, and therefore make no assumptions regarding the formation or structure of the clusters.

Definition 3.1. An HBE mechanism $\text{Exp}(D, f)$ is an algorithm that takes a dataset D and a clustering function $f : \text{dom}(R) \to C$, and outputs a global HBE (Definition 2.4) for the clustering of D defined by f. We say that Exp satisfies ε_{exp}-DP if for any clustering function f, for any pair of neighboring datasets $D \sim D'$, and any set of explanations $S \subseteq Range(\text{Exp})$, it holds $\mathbf{Pr}[\text{Exp}(D, f) \in S] \leq e^{\varepsilon_{exp}} \cdot \mathbf{Pr}[\text{Exp}(D', f) \in S]$.

Note that a clustering algorithm may output different clustering functions for two neighboring datasets. However, Definition 3.1 is motivated by the sequential composition theorem (Proposition 2.7) as our approach is aimed at a privately computed f. To formally argue that the entire process of DP clustering and explanation satisfies DP by applying sequential composition, it suffices to show that if we fix the output of the clustering algorithm (i.e., a clustering function), the distribution of outputs of $\text{Exp}(\cdot, f)$ only changes slightly under two neighboring input datasets, *with the fixed clustering function* f. A similar setting has also been used for DP classifier explanations [58]. The resulting overall privacy guarantee is as follows. Let $M_{clust}(D, C)$ be a DP clustering mechanism, i.e, an algorithm that outputs a clustering function f while satisfying ε_{clust}-DP (e.g, DP k-means [64], where f is defined by the centers). By Proposition 2.7 (sequential composition), the sequential, adaptive composition $\text{Exp}(D, M_{clust}(D, C))$ satisfies $(\varepsilon_{exp} + \varepsilon_{clust})$-DP.

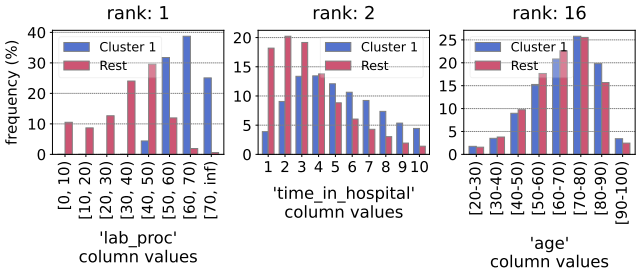


Figure 4: Ranked explanation candidates for Cluster 1 of the Diabetes dataset.

One challenge with existing approaches is the high sensitivity of previously proposed HBE score functions, which cannot be used directly in the DP setting (e.g., Proposition 4.1). Moreover, these functions are applied to pre-computed explanation candidates (Definition 2.2), and *privately* computing all candidates incurs a significant waste of privacy budget. Instead, one would hope to assess *attribute* quality directly, and produce DP histograms only for attributes selected for the output.

An *attribute combination* $\mathcal{AC} : C \to \mathcal{A}$ maps each cluster label to an attribute. Thus, our goal is to find a high-quality attribute combination such that the histograms of the corresponding attributes form a high-quality HBE[2]. This leads to our first goal: devising *candidate attribute* quality functions with low sensitivity while preserving utility, where sensitivity is as defined in Definition 2.8. Then, our next goal is to develop a privacy-preserving algorithm that leverages the low-sensitivity score to produce high-quality explanations.

We can now summarize the challenges addressed in this work:

Problem 1 (Low Sensitivity Quality Functions)**.** *Find a low-sensitivity, global quality function* GlScore *that maps a sensitive dataset D, a clustering function $f : \text{dom}(R) \to C$, and an attribute combination $\mathcal{AC}$ to a real number.*

Once we have a low-sensitivity score function, we can rank the explanations and return the highest scoring ones. Nevertheless, we still need a private mechanism to allow us to do so with low error.

Problem 2 (Select Top explanation attributes)**.** *Given a sensitive dataset D, a clustering function $f : \text{dom}(R) \to C$, and a privacy budget ε, find a high-scoring attribute combination $\mathcal{AC}$ according to the global score function* GlScore *and output the corresponding global explanation while satisfying ε-DP.*

Example 3.2. Figure 4 gives an example of three explanation candidates for Cluster 1 of the Diabetes dataset. In this example, the top-ranked candidate is also selected to explain Cluster 1 in the final output, showcased in Example 1.1.

[2]While the global explanation uses one histogram per cluster, as in prior work [8], our framework can be extended to output multiple histograms per cluster. However, this comes at the cost of increased complexity, as further discussed in Appendix B.

In the next section, we describe our solution to Problem 1 and then describe our approach for obtaining high-scoring explanations (Definition 2.4) while satisfying ε-DP to solve Problem 2.

4 Low Sensitivity Quality Functions

In this section, we address Problem 1 by building upon prior work [8, 10, 11, 30, 75] and focusing on three prominent quality aspects of HBEs: Sufficiency [8, 10], interestingness [30, 61], and diversity [8, 75]. We prove that these measures are highly sensitive, making them unsuitable for a DP algorithm. These results then motivate us to devise low-sensitivity variants.

4.1 Interestingness

The interestingness of an HBE is quantified as the distributional shift of the given attribute values between the cluster and the full-dataset [8, 11, 30]. There are many ways to quantify distance between probability distributions. Among them, the *total variation distance* (TVD) has been shown to be effective in quantifying the interestingness of HBEs [8].

For a dataset D and a cluster $D_c \subseteq D$, the TVD between the distributions of values in $\pi_A(D)$ and $\pi_A(D_c)$, and thus the *interestingness* of an attribute A with respect to the cluster D_c, is defined as

$$\mathrm{TVD}(\pi_A(D), \pi_A(D_c)) = \frac{1}{2} \sum_{a \in \mathrm{dom}(A)} \left| \frac{cnt_{A=a}(D)}{|D|} - \frac{cnt_{A=a}(D_c)}{|D_c|} \right| \quad (1)$$

The global interestingness of an attribute combination $\mathcal{AC}$, is defined as the average of all single-cluster interestingness scores [8], which we denote by $\mathrm{Int}(D, f, \mathcal{AC})$

In the context of DP, *we cannot make any assumptions about the input dataset D or the clustering function*, and privacy must be guaranteed for any input, regardless of the cluster size. In particular, when a cluster is small, the removal of a single tuple from it can significantly change the cluster's internal distribution, which may lead to a large change in the distribution distance, as reflected in Equation (1). Given that sensitivity analysis must account for such cases, the sensitivity of this function is high.

Proposition 4.1. *The sensitivity of* TVD *is at least* $\frac{1}{2}$ *and its range is* $[0, 1]$.

Intuitively, since this function outputs a number between 0 and 1, its sensitivity is relatively high. While the proof of Proposition 4.1 is provided in Appendix A.1, we illustrate the issue in the following example.[3]

Example 4.2 (The Issue with Interestingness sensitivity). Suppose D is a dataset of size 100,000 with a binary attribute A. Suppose further that 95% of individuals in the dataset we have $A = 1$. Assume that the clustering is imbalanced, with a cluster D_c that contains only 1 tuple t, and $t[A] = 0$. In this case, from Equation (1), the interestingness score of attribute A for explaining cluster D_c is $\frac{1}{2}(|0.95 - 0| + |0.05 - 1|) = 0.95$. however, suppose a new tuple t' that satisfies $t'[A] = 1$ is added to cluster D_c. Now the interestingness in Equation (1) becomes $\frac{1}{2}(|95,001/100,001 - 0.5| + |5000/100,001 - 0.5|) \approx 0.45$. Note that while we only added a single tuple, it led to a change of $\approx 0.95 - 0.45 = 0.5$ in the interestingness function.

Low-Sensitivity Interestingness. We propose a new interestingness function that is inspired by the interestingness of [8], but has lower sensitivity.

Definition 4.3 (Low Sensitivity Interestingness). For a dataset D and a clustering function $f : \mathrm{dom}(R) \to C$, define the interestingness of an attribute A for explaining the cluster $D_c = \{t \in D \mid f(t) = c\}$ as

$$\mathrm{Int}_p(D, f, c, A) = \frac{1}{2} \sum_{a \in \mathrm{dom}(A)} \left| cnt_{A=a}(D_c) - \frac{|D_c|}{|D|} \cdot cnt_{A=a}(D) \right|.$$

To gain insight into the low sensitivity formulation, note that $\mathrm{Int}_p(D, f, c, A) = |D_c| \cdot \mathrm{TVD}(\pi_A(D), \pi_A(D_c))$, where TVD is the total variation distance (Equation (1)). Hence, for a given cluster, the ranking of attributes by low-sensitivity interestingness (Definition 4.3) is identical to their ranking by TVD deviation, which is the sensitive interestingness [8]. However, intuitively, the effect of small clusters on the sensitivity is mitigated by the multiplication by $|D_c|$, as now the sensitivity is 1, but the function takes values in $[0, |D_c|]$. This new interestingness function gives us more leeway for adding the necessary noise to satisfy DP, while not distorting the attribute ranking too much when the cluster is sufficiently large. (see Appendix A.1 for proof).

Proposition 4.4. *The sensitivity of* $\mathrm{Int}_p(D, f, c, A)$ *is* 1 *and its range is* $[0, |D_c|]$.

4.2 Sufficiency

We begin by revisiting the notion of sufficiency from previous work. Recognizing the need to ensure that a given explanation is relevant only to its associated class, Dasgupta et al. [10] introduced an abstract definition of sufficiency (faithfulness) of *classifier explanations*. Informally, the sufficiency of an explanation at $t \in D$ is defined as the fraction of tuples assigned the same prediction as t, out of all tuples for which the explanation of t "holds". In other words, if t is assigned an explanation that holds for another tuple, then that tuple should have the same classification as t. Then, a global sufficiency score is obtained by averaging the sufficiency value at all tuples. However, their framework assumes the existence of a binary relation indicating whether an explanation holds for a given tuple, and extending this idea to HBEs is not straightforward. Copul et al. [8] proposed an adaptation of the formula from [10], replacing the classifier prediction with cluster assignment, and the binary relation with a probability value derived from the HBE. Specifically, they quantify the extent to which an HBE, defined by an attribute combination $\mathcal{AC}$, "holds for" a tuple t that belongs to a cluster D_c as the probability that uniformly random tuple sampled from D conditioned on having the same value in $\mathcal{AC}(c)$ as t, belongs to the same cluster as t. Unfortunately, this function, which we denote $\mathrm{Suf}(D, f, \mathcal{AC})$, is too sensitive to be of use. [4]

Proposition 4.5. *The sensitivity of* $\mathrm{Suf}(D, f, \mathcal{AC})$ *is at least* $\frac{1}{2}$ *and its range is* $[0, 1]$.

[3] Previous work has also considered the Jensen-Shannon distance [41] as an interestingness measure. We show that it is highly sensitive as well, making it unsuitable for the privacy setting. The proposition and proof appear in Appendix A.1.

[4] See further discussion in Appendix A.2.

Low Sensitivity Sufficiency. Next, we introduce an alternative formulation of the sufficiency measure for the following reasons. First, the original function is too sensitive to be of use under DP. Second, the alternative formulation shows that single-attribute sufficiency can, in fact, be measured for each individual cluster, with the global sufficiency being the average of these single-cluster sufficiency functions, which simplifies the sensitivity analysis. Third, it ensures that the range of each single-cluster sufficiency matches that of the interestingness measure, $[0, |D_c|]$ for a cluster D_c, and that the sensitivity of the modified sufficiency is also 1, making the two directly comparable.

Definition 4.6 (Low Sensitivity Sufficiency). For a dataset D and a clustering function $f : \text{dom}(R) \to C$, define the sufficiency of an attribute A for explaining the cluster D_c as

$$\text{Suf}_p(D, f, c, A) = \sum_{a \in \text{dom}_{D_c}(A)} \frac{cnt_{A=a}(D_c)^2}{cnt_{A=a}(D)}.$$

Note that we do not divide by zero, since the sum is only over values appearing at least once in D_c, and hence also in D. Intuitively, from Definition 4.6, we see that Suf_p is maximized when values of $\text{dom}(A)$ appearing in the cluster D_c occur *only* within it, reflecting maximal sufficiency of A, since observing a tuple's value in that attribute "suffices" to determine its membership to D_c. Conversely, the function decreases when such values also appear frequently outside the cluster, indicating that A is insufficient to explain D_c.

Our modified sufficiency is consistent with the definition in previous work, as the equality in item *(1)* of Proposition 4.7 ensures that both measures induce the same ranking of attribute combinations when the dataset and clustering are fixed. We prove:

Proposition 4.7. *For an attribute combination $\mathcal{AC}$,*

(1) The following equality holds:

$$|D| \cdot \text{Suf}(D, f, \mathcal{AC}) = \sum_{c \in C} \text{Suf}_p(D, f, c, \mathcal{AC}(c))$$

(2) The sensitivity of $\text{Suf}_p(D, f, c, A)$ is 1 and its range is $[0, |D_c|]$.

4.3 Diversity

The diversity measure is designed to quantify the overall distinctiveness among explanations. We provide an intuitive introduction of the diversity measure, denoted Div, outline its inadequacies in the context of DP, and discuss its relation to our new measure.[5]

Diversity is initially defined for a pair of single-cluster explanations, and generalized to global explanations by averaging all pairwise diversities. When two single-cluster explanations utilize different attributes, the diversity for that pair attains its maximum value of 1. If the explanations use the same attribute, the diversity is measured as the distance between the two distributions, quantifying the new knowledge gained from the additional explanation on that attribute. However, common metrics used in previous work, such as total-variation distance and Jensen-Shannon distance, have high sensitivity (Proposition 4.1, Footnote 3), which intuitively means they are unsuitable for the DP setting.

Inspired by the diversity measure of [8], we introduce the following low-sensitivity pairwise diversity function:

[5]The reader is referred to [8] for the original definition, and to Appendix A for our sensitivity analysis of this function.

Definition 4.8 (Pair Diversity). For a dataset D and a clustering function f, the diversity score of a pair of attributes $A_c, A_{c'} \in \mathcal{A}$, where A_c (respectively $A_{c'}$) is a candidate attribute for explaining $D_c = \{t \mid f(t) = c\}$ (respectively $D_{c'}$), is

$$d(D, f, c, c', A_c, A_{c'}) = \min\{|D_c|, |D_{c'}|\} \times \begin{cases} 1 & A_c \neq A_c \\ \frac{1}{2} \sum_{a \in \text{dom}(A)} \left| \frac{cnt_{A=a}(D_c)}{\max\{|D_c|,1\}} - \frac{cnt_{A=a}(D_{c'})}{\max\{|D_{c'}|,1\}} \right| & A_c = A_{c'} \end{cases}$$

To interpret Definition 4.8, note that when both clusters are not empty and are explained by the same attribute A, we have $d(D, c, c', A, A) = \min\{|D_c|, |D_{c'}|\} \cdot \text{TVD}(\pi_A(D_c), \pi_A(D_{c'}))$, where TVD is defined in Equation (1). Thus, for a given pair of clusters, the low-sensitivity pairwise diversity ranks attributes identically to the sensitive TVD deviation, and is maximized when the clusters are explained by different attributes.

Following previous work on result diversification (e.g., [6, 70]), we define global diversity as the average of all pairwise diversities. Note that achieving low sensitivity requires that pairs from smaller clusters have a reduced impact on the global diversity function, as is evident in Definition 4.9. The sensitivity bound for this function is provided in Proposition 4.10, leveraging the fact that a convex combination of sensitivity-1 functions has a sensitivity bounded by 1 (see Lemma A.3).

Definition 4.9 (Global Diversity). For a dataset D and a clustering function $f : \text{dom}(R) \to C$, define the diversity score of an attribute combination $\mathcal{AC}$ as

$$\text{Div}_p(D, f, \mathcal{AC}) = \frac{1}{\binom{|C|}{2}} \sum_{\{c,c'\} \subseteq C} d(D, f, c, c', \mathcal{AC}(c), \mathcal{AC}(c'))$$

where the sum is over all distinct pairs of cluster labels.

Proposition 4.10. *The sensitivity of Div_p is bounded by 1. Moreover, its range is $[0, R_{\text{Div}}]$ where $R_{\text{Div}} = \frac{1}{\binom{|C|}{2}} \sum_{i=1}^{|C|} (|C| - i) \left|D_{c_i}\right|$ is a weighted average of the cluster sizes, and $|D_{c_i}| \leq |D_{c_{i+1}}|$.*

4.4 Combining All Quality Functions

Following prior work on explainability [8, 37, 43, 52, 60], we combine the different measures into a weighted sum, with weights that may be user-defined or preference-driven. We first define the single-cluster score function, which assesses the quality of an attribute A in explaining a given cluster D_c.

Definition 4.11 (Single-Cluster Score). Let $\gamma = (\gamma_{\text{Int}}, \gamma_{\text{Suf}})$ be a pair of non-negative parameters that sum to 1. For a dataset D and a clustering function $f : \text{dom}(R) \to C$, define the quality score $A \in \mathcal{A}$ as a candidate attribute for a cluster $c \in C$, as

$$\text{Score}_\gamma(D, f, c, A) = \gamma_{\text{Int}} \cdot \text{Int}_p(D, f, c, A) + \gamma_{\text{Suf}} \cdot \text{Suf}_p(D, f, c, A).$$

Importantly, we can bound the sensitivity of the score function.

Proposition 4.12. *$\text{Score}_\gamma(D, f, c, A)$ has sensitivity bounded by 1 and its range is $[0, |D_c|]$.*

The global score combines individual measures to assess the quality, with smaller clusters contributing less than larger ones for the same distribution distance, addressing the high sensitivity of the original global score. As shown in Proposition 4.1, removing a

single point from a small cluster can significantly alter its column distribution, leading to a large score change. In Appendix A.4, we prove the bound Proposition 4.14.

Definition 4.13 (Global Score). Let $\lambda = (\lambda_{\text{Div}}, \lambda_{\text{Int}}, \lambda_{\text{Suf}})$ be non-negative parameters that sum to 1. For a dataset D and a clustering function f, define the overall quality score of an attribute combination $\mathcal{AC}$ as

$$\text{GlScore}_\lambda(D, f, \mathcal{AC}) = \lambda_{\text{Int}} \cdot \text{Int}_p(D, f, \mathcal{AC}) + \lambda_{\text{Suf}} \cdot \text{Suf}_p(D, f, \mathcal{AC}) + \lambda_{\text{Div}} \cdot \text{Div}_p(D, f, \mathcal{AC})$$

where we extend $\text{Int}_p(D, f, \mathcal{AC}) = \frac{1}{|C|} \sum_{c \in C} \text{Int}_p(D, f, c, \mathcal{AC}(c))$ and $\text{Suf}_p(D, f, \mathcal{AC}) = \frac{1}{|C|} \sum_{c \in C} \text{Suf}_p(D, f, c, \mathcal{AC}(c))$.

Proposition 4.14. *GlScore_λ has sensitivity bounded by 1. Moreover, its range is $[0, R_{\text{GlScore}_\lambda}]$ where*

$$R_{\text{GlScore}_\lambda} = (\lambda_{\text{Int}} + \lambda_{\text{Suf}}) \cdot \frac{1}{|C|} \sum_{c \in C} |D_c| + \lambda_{\text{Div}} \cdot R_{\text{Div}}$$

is a weighted average of the cluster sizes, and R_{Div} is defined as in Proposition 4.10.

Selecting the weight parameters. Previous work on HBEs in the non-private setting [8] has shown through user studies and quantitative evaluations that equal weight distribution $\lambda_{\text{Int}} = \lambda_{\text{Suf}} = \lambda_{\text{Div}} = 1/3$ produces high quality and informative explanations. We adopt this default parameter setting in our framework. However, weights can be adjusted based on preference. Our experiments show that DPClustX maintains high explanation quality compared to the non-private baseline across different weight distributions.

5 The DPClustX Framework

In this section, we introduce our algorithms for computing explanations (depicted in Figure 3), addressing Problem 2. Since the space of all possible attribute combinations is generally too large to analyze, we propose a novel DP-tailored manner of pruning the search space. Following previous work [8], we construct a small set of high-quality candidate *attributes* for each cluster (Stage-1), which serve as the candidate pool for a *global* explanation of all clusters (Stage-2). To optimize our privacy budget usage, we avoid generating noisy histograms at Stage-1.

5.1 Stage-1: Construct Private Candidate Sets

We now present our candidate set construction algorithm (Figure 3).

In DPClustX, to satisfy DP, we privately select top-k candidate *attributes* for each cluster based on our single-cluster score function. Since doing so by iteratively applying the exponential mechanism k times can be computationally expensive, we adopt the *One-shot Top-k mechanism* (Section 2.1) to privately select top-k.

The pseudo code in Algorithm 1 provides the candidate selection procedure. we first set the privacy budget $\varepsilon_{Topk} = \varepsilon_{CandSet}/|C|$ allocated for each Top-k selection, and in Line 2 we set the noise scale parameter $\sigma = 2k/\varepsilon_{Topk}$. Then, we run the Top-$k$ attributes selection procedure for each cluster $c \in C$. Specifically, in Line 5 we compute the noisy single-cluster score (Definition 4.11) for each cluster and attribute. In Line 7, we sort the attributes based on the noisy scores for each cluster. In Line 9, we define the set S_c. The output returned in Line 11 consists of the sets S_c for every $c \in C$.

Note that $1/|C|$-fraction of the total privacy budget $\varepsilon_{CandSet}$ is allocated for selecting top candidate attributes for each cluster. While one might have hoped to use parallel composition, this is typically not possible. The quality score of an attribute for a given cluster depends on the entire dataset, not just the cluster itself. A high-scoring attribute exhibits large distributional deviation between the cluster and the full dataset, requiring consideration of tuples outside the cluster.

Algorithm 1 Select-Candidates: Generate all single-cluster top-k candidate attributes

Input: Dataset D, clustering function $f : \text{dom}(R) \to C$, hyperparameters $\gamma = (\gamma_{\text{Int}}, \gamma_{\text{Suf}})$ Attribute set $\mathcal{A}$, privacy parameter $\varepsilon_{CandSet}$, cardinality k.

Output: Sets $S_{c_1} \ldots, S_{c_{|C|}}$ where S_{c_i} contains noisy Top-k candidate explanation attributes for D_{c_i}.

1: $\varepsilon_{Topk} \leftarrow \varepsilon_{CandSet}/|C|$
2: Set $\sigma \leftarrow 2k/\varepsilon_{Topk}$
3: **for** $c \in C$ **do**
4: **for** $A \in \mathcal{A}$ **do**
5: $s_A \leftarrow \text{Score}_\gamma(D, f, c, A) + \text{Gumbel}(\sigma)$
6: **end for**
7: Sort $\{s_A \mid A \in \mathcal{A}\}$ in descending order.
8: Let $A_1, \ldots, A_k$ be the attributes corresponding to the top-k noisy scores.
9: Set $S_c \leftarrow \{A_1, \ldots, A_k\}$
10: **end for**
11: **return** Candidate sets $S_{c_1} \ldots, S_{c_{|C|}}$.

The privacy and utility guarantees of Algorithm 1 are given by the following proposition:

Proposition 5.1. *Given a clustering function f, a set of attributes $\mathcal{A}$, a privacy parameter $\varepsilon_{CandSet}$, non-negative hyperparameters $\gamma_{\text{Int}}, \gamma_{\text{Suf}}$ that sum to 1, and a size parameter k, the following holds:*

(1) Algorithm 1 satisfies $\varepsilon_{CandSet}$-DP.

(2) For $c \in C$, denote by $\text{OPT}_c^{(\ell)}$ the ℓ-th highest (true) score, and by $A_c^{(\ell)}$ the ℓ-th explanation attribute selected by Algorithm 1 to S_c. For all c and $\ell = 1, 2, \ldots, k$, we have

$$\Pr\left[\text{Score}(c, A_c^{(\ell)}) \le \text{OPT}_c^{(\ell)} - \frac{2|C| \cdot k}{\varepsilon_{CandSet}} (\ln|\mathcal{A}| + t)\right] \le e^{-t}.$$

where we denote $\text{Score}(c, A_c^{(\ell)}) = \text{Score}(D, f, c, A_c^{(\ell)})$

The proof of item 1 follows from Proposition 2.7 and the analysis of the one-shot Top-k mechanism [14, 15]. The proof of item 2 is based on the utility proposition of EM (Theorem 3.11 in [19]). Both proofs can be found in Appendix A.

Example 5.2. Reconsider Example 1.1. Algorithm 1 outputs the set $S_1 = \{\texttt{lab\_proc}, \texttt{time\_in\_hospital}, \texttt{num\_medications}\}$, which indeed comprises the top-3 attributes for Cluster 1 obtained from the ranked list partly presented in Figure 4. Algorithm 1 outputs one such set for each cluster.

5.2 Stage-2: A Private Global Explanation

We now present a privacy-preserving mechanism that, given a dataset D and a clustering function f, outputs an HBE for the given clustering while selecting the explanation attribute for each cluster D_c from its corresponding set S_c (Section 5.1), as depicted in Figure 3. By restricting the search space to the candidate sets, DPClustX evaluates only $k^{|C|}$ attribute combinations, instead of the full space of $|\mathcal{A}|^{|C|}$ combinations.

As in Stage-1, pre-computing noisy histograms and selecting the best candidate combination afterward incurs a waste of privacy budget, and evaluating the global score based on noisy histograms introduces more noise than necessary due to the injection of noise into *each count* of the private histograms. To address these issues, we propose the following approach.

We begin by computing the candidate sets S_c using Algorithm 1 with privacy budget $\varepsilon_{CandSet}$. Then, we apply the EM (Definition 2.9) to select the noisy-best *attribute combination* drawn from each cluster's candidate set using our global score function (Definition 4.13). Subsequently, we generate the noisy histograms *only for the selected attribute combination*, employing a black-box mechanism for histogram generation while leveraging the parallel composition property for efficient privacy budget allocation.

The pseudo code describing our approach can be found in Algorithm 2. We first set the marginal weights for interestingness and sufficiency in the single-cluster score function. In Line 3, Algorithm 1 is invoked to obtain candidate sets S_c for each cluster. In Line 6, the EM is run using the global score function (Definition 4.13), privacy parameter $\varepsilon_{TopComb}$, and the set of possible candidate attribute combinations, assigning each cluster c an attribute from S_c. Line 8 extracts all attributes selected at least once. In Line 9, we allocate $\varepsilon_{hist,all} = \varepsilon_{Hist}/(2|\mathcal{A}'|)$ and $\varepsilon_{hist,cluster} = \varepsilon_{Hist}/2$ as privacy budgets for full-dataset and cluster histograms, respectively. Lines 10–12 compute full-dataset histograms. Line 16 computes cluster histograms. In Line 17, we derive out-of-cluster histograms by subtracting cluster from full-dataset histograms, replacing negatives with 0. Line 18 constructs explanations e_c using these histograms, and Line 20 outputs the global explanation $\{e_c \mid c \in C\}$.

The following theorem states the privacy guarantee of DPClustX.

Theorem 5.3. *Given a clustering function $f : \mathrm{dom}(R) \to C$, number of candidates k, privacy parameters $\varepsilon_{CandSet}, \varepsilon_{TopComb}, \varepsilon_{Hist}$, Algorithm 2 is $(\varepsilon_{CandSet} + \varepsilon_{TopComb} + \varepsilon_{Hist})$-DP.*

Proof Sketch. The execution of Algorithm 1 satisfies $\varepsilon_{CandSet}$-DP by Proposition 5.1. Since the function $\mathrm{GlScore}_\lambda$ has sensitivity 1 (Proposition 4.14), the execution of the exponential mechanism in Line 6 guarantees $\varepsilon_{TopComb}$-DP, applying Theorem 2.10. We assume that each execution of $\mathcal{M}_{hist}(\cdot, \varepsilon)$ satisfies ε-DP. Therefore, by sequential composition, the computation of $\widetilde{h}_A$ for all $A \in \mathcal{A}'$ satisfies overall $\varepsilon_{Hist}/2$-DP. Since the clusters are disjoint, the computation of all noisy histograms $\widetilde{h}^c$ satisfies overall $\varepsilon_{Hist}/2$-DP by parallel composition. Note that the computation of each $\widetilde{h^{-c}}$ in Line 17 is post-processing, and therefore incurs no additional privacy loss. Overall, by sequential composition, the entire algorithm guarantees $(\varepsilon_{CandSet} + \varepsilon_{TopComb} + \varepsilon_{Hist})$-DP. □

Algorithm 2 DPClustX Algorithm - Generate Global Explanation

Input: Dataset D, clustering function f, number of candidates k, privacy parameters $\varepsilon_{CandSet}, \varepsilon_{TopComb}, \varepsilon_{Hist}$, hyperparameters $\lambda_{\text{Div}}, \lambda_{\text{Int}}, \lambda_{\text{Suf}}$

Output: Global explanation $\{e_c \mid c \in C\}$ (Definition 2.4)

1: Let $\gamma_{\text{Suf}} \leftarrow \lambda_{\text{Suf}}/(\lambda_{\text{Suf}} + \lambda_{\text{Int}})$ and $\gamma_{\text{Int}} \leftarrow \lambda_{\text{Int}}/(\lambda_{\text{Suf}} + \lambda_{\text{Int}})$
2: *// Invoke Algorithm 1*
3: Let $S_{c_1}, \ldots S_{c_{|C|}} \leftarrow$ Select-Candidates$(D, f, \gamma, \mathcal{A}, \varepsilon_{CandSet}, k)$
4: *// Select attribute combination*
5: Let $\mathcal{R} \leftarrow \{\mathcal{AC} \mid \forall c.\ \mathcal{AC}(c) \in S_c\}$ be the set of candidate combinations, mapping each $c \in C$ to an attribute in S_c.
6: $\mathcal{AC} \leftarrow \mathcal{M}_E(D, \mathrm{GlScore}_\lambda, \Delta_{\mathrm{GlScore}}, \mathcal{R}, \varepsilon_{TopComb})$
7: *// Generate full-dataset histograms*
8: Let $\mathcal{A}' \leftarrow \{\mathcal{AC}(c) \mid c \in C\}$ be the set of attributes appearing in at least once in the combination.
9: $\varepsilon_{hist,all} \leftarrow \varepsilon_{Hist}/(2\,|\mathcal{A}'|)$, $\varepsilon_{hist,cluster} \leftarrow \varepsilon_{Hist}/2$
10: **for** every $A \in \mathcal{A}'$ **do**
11: $\quad \widetilde{h}_A \leftarrow \mathcal{M}_{hist}(\pi_A(D), \varepsilon_{hist,all})$
12: **end for**
13: *// Compute single-cluster explanations*
14: **for** every $c \in C$ **do**
15: $\quad$ Let $A_c \leftarrow \mathcal{AC}(c)$ be the attribute selected for D_c.
16: $\quad \widetilde{h}^c \leftarrow \mathcal{M}_{hist}(\pi_{A_c}(D_c), \varepsilon_{hist,cluster})$
17: $\quad \widetilde{h}^{-c} \leftarrow \max(\widetilde{h}_{A_c} - \widetilde{h}^c, 0)$
18: $\quad e_c \leftarrow (c, A_c, \widetilde{h}^{-c}, \widetilde{h}^c)$
19: **end for**
20: **return** global explanation $\{e_c \mid c \in C\}$.

Example 5.4. Consider the setting in Example 1.1, and suppose the input contains 3 clusters of the Diabetes dataset. First, a candidate set is obtained for each cluster, with the set for Cluster 1 shown in Example 5.2. In Line 6, the selected attribute combination is: Cluster 1: `lab_proc`, Cluster 2: `lab_proc`, and Cluster 3: `discharge_disp`. DP noisy histograms are generated for these attributes. Figure 2a depicts a part of the output , showcasing the explanation for Cluster 1. The full output of Algorithm 2 contains such an explanation for each cluster.

Time complexity. The time complexity DPClustX is proportional to $O(|\mathcal{A}| \cdot |C| + k^{|C|})$, where the first term is due to Stage-1 (Algorithm 1) and the second to Stage-2 (Algorithm 2). Stage-1 constructs a candidate set for each cluster, requiring $O(|\mathcal{A}| \cdot |C|)$ evaluations of the single-cluster score function, each involving two count group-by queries. The noisy scores are computed only once using the one-shot top-k mechanism, instead of k times using repeated applications of the EM that require overall $O(k\,|\mathcal{A}|\,|C|)$ noisy scores evaluations. Stage-2 performs $O(k^{|C|})$ evaluations of the global score function, corresponding to the number of global explanation candidates. The complexity of each global score evaluation is as follows. The average interestingness and sufficiency across all clusters require $O(|C|)$ count group-by queries. Computing global diversity requires $O(|C|^2)$ count group-by queries, as it is defined as average of pairwise diversities, with each pair requiring two count queries.

6 Experiments

In this section, we evaluate the quality and efficiency of the explanations generated by DPClustX with the following questions:

(1) With respect to the quality measures for HBEs, how does DPClustX perform compared to the non private method and a naive approach which computes all histograms in advance?
(2) How close is the attribute combination selected by DPClustX to that of the non-private baseline?
(3) How efficient is DPClustX in computing the explanations?

Summary of our findings. With a total privacy budget of $\varepsilon = 0.1$, DPClustX selects attributes of comparable quality to those chosen by the non-private baseline in all datasets and clustering methods. Moreover, at $\varepsilon = 1$, DPClustX consistently selects the same attributes as the non-private baseline across all runs and clustering methods for the Diabetes dataset. The execution time of DPClustX for generating explanations averages under 6.6 seconds across all datasets and clustering methods with at most 9 clusters.

6.1 Experimental Settings

We next present our settings for the experiments. We have implemented DPClustX in Python 3.9.19 using the Pandas and NumPy libraries. All experiments were run on an Intel Xeon CPU-based server with 24 cores and 96 GB of RAM. We use the Geometric mechanism [26] for DP histogram generation, implemented by DiffPrivLib [31]. The source code is available in [1].

Default parameters. Unless mentioned otherwise, the following default parameters are used. We set $\varepsilon_{CandSet} = \varepsilon_{TopComb} = \varepsilon_{Hist} = 0.1$. Thus, the combined privacy budget for attribute selection is $\varepsilon = 0.2$, which we vary from 0.001 to 1 when evaluating its effect on the selected attributes. The number of candidate attributes per cluster selected at Stage-1 (denoted by k in Section 5.1) is set to 3, a choice supported by ablation studies presented in Figure 7, where it is varied from 1 to 5. Following the discussion in Section 4.4, we set the default values $\lambda_{\text{Int}} = \lambda_{\text{Suf}} = \lambda_{\text{Div}} = 1/3$. We also evaluate alternative weight distributions, where we set one weight to zero and each of the remaining two to 1/2. Unless otherwise stated, we use 5 clusters for evaluation.

Datasets. We evaluate DPClustX on the following two datasets:

- **US Census Data** [49]: 2,458,285 tuples and 68 attributes. This dataset contains a one percent sample of the Public Use Microdata Samples (PUMS) person records drawn from the 1990 census.
- **Diabetes** [7]: 101,766 tuples and 47 attributes. This dataset comprises ten years (1999–2008) of clinical care data. Each tuple represents a hospital record of a diabetic patient. Numerical and large-domain categorical attributes are binned in accordance with [63] to ensure interpretable histograms, following prior work on HBEs [8, 11]. Domain sizes vary from 2 to 39. Further details can be found in Appendix C.
- **Stack Overflow Developer Survey** [56]: 98,855 tuples and 60 attributes. This dataset is obtained from the 2018 Stack Overflow Developer Survey. We consider 60 attributes, which include demographic information, professional background, and work habits of the respondents. Numerical and large-domain categorical attributes are binned. Domain sizes vary from 2 to 22. The preprocessing of this dataset is detailed in Appendix C.

Clustering methods. To demonstrate the effectiveness and versatility of DPClustX, we evaluate it across diverse scenarios, using both private and non-private clustering methods, following prior work on DP classifier explanations [28, 50, 58]. In real-world deployment, to protect data privacy, the clustering function must be either privately computed or data-independent. This evaluation highlights the robustness of our method in providing meaningful explanations for different clustering tasks, including: *(i)* k-means, *(ii)* DP-k-means [64] implemented by DiffPrivLib [31], *(iii)* k-modes, *(iv)* Agglomerative Clustering, *(v)* Gaussian Mixture Models (GMMs). The budget for DP-k-means is set to $\varepsilon = 1$, as commonly used for clustering in experimental settings [53, 54, 64, 65]. Categorical attributes are transformed into equivalent numerical data by mapping each domain value to a unique integer. Due to its scalability limitations, Agglomerative clustering is not applied to the Census dataset.

Baselines. Despite the importance of explainability, to the best of our knowledge, no attempts have been made to develop explanations for clustering results under DP. Nevertheless, we consider one non-private algorithm and two devised DP adaptations of that baseline for comparative evaluation. We compare the performance of DPClustX with the following approaches for generating explanations:

- **TabEE (non-private)**: The non-private algorithm in [8] for finding a high-scoring attribute combination. The algorithm selects the top attribute combination from a pre-constructed candidate pool based on the original, sensitive definition of the quality functions, as we discussed in Section 4.
- **DP-TabEE**: We implement a direct adaptation of the TabEE algorithm to satisfy DP. This baseline uses the original, sensitive quality functions for attribute selection, but injects the required noise to satisfy DP, according to Theorem 2.10 and the sensitivity of the quality functions (i.e., Propositions 4.1 and 4.5).
- **DP-Naive**: In Section 5 we discussed a naive approach for computing HBEs under DP. We implement this simple DP baseline as follows. Given a privacy budget ε, we compute each of the full-dataset histograms using a budget $\varepsilon/(2\,|\mathcal{A}|)$ for each attribute. We compute the histogram of each cluster for each attribute using a budget of $\varepsilon/(2\,|\mathcal{A}|)$ per cluster. Then, as a post-processing step, we run the TabEE-based algorithm on the noisy histograms. By the composition and post-processing theorems, the entire algorithm satisfies ε-DP.

Evaluation measures. While in this work we introduce low-sensitivity variants of *interestingness*, *sufficiency* and *diversity*, which are incorporated into our algorithm, the *original*, sensitive variants of these functions can still be used for evaluating the quality of the selected attribute combination. Hence, we let $Quality = \lambda_{\text{Int}} \cdot \text{Int} + \lambda_{\text{Suf}} \cdot \text{Suf} + \lambda_{\text{Div}} \cdot \text{Div}$. be the global score function from [8], using the sensitive quality functions Int, Suf, Div (Section 4).

We also evaluate the similarity between the attribute combination selected by non-private baseline, denoted $\mathcal{AC}^*$, and that of DPClustX or the DP-Naive baseline. To this end, we adapt the mean absolute error (MAE) [71] to our discrete setting. The MAE score for a combination $\mathcal{AC}$ is given by $\text{MAE}(E) = \frac{1}{|C|} \sum_{c \in C} \mathbf{1}_{\{\mathcal{AC}(c) \neq \mathcal{AC}^*(c)\}}$.

6.2 Quality Analysis

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

Selected attributes quality score. In this experiment (Figure 5), we evaluate the *Quality* score of the selected attribute combination with default parameters for different privacy budgets ε, where $\varepsilon_{CandSet} = \varepsilon_{TopComb} = \varepsilon/2$. Note that this experiment examines the attribute choice, hence histogram generation is not needed. Results for additional cluster numbers are provided in Appendix C, showing similar trends. We find that increasing the total privacy budget ε improves the quality scores achieved by DPClustX in all cases, and that it consistently outperforms the other DP baselines. Moreover, DP-TabEE failed to improve in the examined range. For the Diabetes dataset, at $\varepsilon = 0.1$, DPClustX scores are only 0.66% lower than TabEE on average, while DP-Naive scores 20.22% lower. At $\varepsilon = 1$, DPClustX matches TabEE across all methods. For the Census dataset, at $\varepsilon = 10^{-2.5}$, DPClustX attains scores 1.6% lower on average, while DP-Naive scores are 9.8% lower. At $\varepsilon = 0.1$, the difference for DPClustX is only 0.003%. For the Stack Overflow dataset, the scores of DPClustX are only 1.3% lower than TabEE on average at $\varepsilon = 0.1$, while DP-Naive scores 9.8% lower. The scores of DP-TabEE are lower by 19% even at $\varepsilon = 1$.

Selected attributes error. This experiment (Figure 6) examines the MAE of the selected attribute combination with varying privacy budgets ε, where $\varepsilon_{CandSet} = \varepsilon_{TopComb} = \varepsilon/2$. Note that all attributes are considered distinct, despite possible correlations. Hence, an MAE score of 0 implies identical attribute choice to that of the non-private TabEE baseline. The results show that DPClustX outperforms the DP baselines in all cases. For the Diabetes dataset, DPClustX selects the same combination as TabEE for all methods at $\varepsilon = 1$, and for most methods at $\varepsilon = 0.1$, with an MAE of 0.04 for GMMs and 0.36 for k-modes. For the Census dataset, the MAE values are below 0.25 for all methods at $\varepsilon = 0.1$ and below 0.18 at $\varepsilon = 1$. For the Stack Overflow dataset at $\varepsilon = 1$, DPClustX obtains average MAE scores below 0.4 for all methods, and below 0.12 for GMMs and k-modes. Notably, with the Stack Overflow and Census datasets, multiple choices achieve nearly optimal scores due to attribute correlations. Consequently, under DP randomization, it is expected that the selection will not always favor the top solution, leading to relatively higher MAE for all DP methods, despite the nearly-optimal quality scores (Figure 5).

Quality for different candidate set sizes. In this experiment (Figure 7) We evaluate the score of the selected attribute combination with varying candidate set sizes from Stage-1 of our algorithm. We focus on the Census and Diabetes datasets, as the Stack Overflow dataset exhibited similar trends. For the Diabetes dataset and all methods except k-modes, we find that increasing the candidate size from 1 to 5 had no effect, with the same attributes selected in all runs. For k-modes, the score increases by 8% between 1 and 3 candidates, and stabilizes. For the Census dataset, a positive trend is observed for all methods between 1 and 3, peaking at 3 and stabilizing. GMMs shows a 40% score improvement between 1 and 2 candidates, with the same attributes selected for 3. We set the default size to 3, as further increase did not improve quality in our experiments, but considerably increased the running time, as shown in Figure 9b.

Quality for different choices of weights. We measure the *Quality* scores of the selected attribute combination for different weight parameters λ_{Int}, λ_{Suf}, and λ_{Div}, setting one to zero and the remaining two to 1/2. The results, detailed in Table 1, show zero or negligible score difference on the Census dataset for 3 clusters. For the Diabetes dataset, scores are lower by merely 0.11% on average across all clustering methods and weight configurations. For 5 clusters, the results show a minor difference of 0.06% on average on the Diabetes dataset, and of 0.13% on the Census dataset. For 7 clusters, we find that the scores are lower by 0.4% on average on the Diabetes dataset. and by only 0.08% for Census dataset. These minor differences indicate that DPClustX selects an attribute combination with a quality comparable to that of the non-private baseline across different weights configurations, offering the same flexibility in parameter selection.

Quality for different numbers of clusters. We examine the impact of varying the number of clusters on the quality of explanations generated by DPClustX compared to the baselines. Figure 8a presents the results for the Census and Diabetes datasets using k-means clustering, with other methods exhibiting similar trends. The results indicate that explanation quality decreases as the number of clusters increases, even without privacy constraints. In all cases, DPClustX outperforms the DP baselines. For Census, both DPClustX and DP-Naive achieve scores comparable to TabEE, whereas DP-TabEE obtains significantly lower scores (36.7% difference on average). For Diabetes, DPClustX outperforms the DP baselines, and maintains a score close to TabEE (lower by 2% on average, and by 0.4% when the number of clusters is at most 9). Note that the presence of small clusters, which tends to occur with a larger number of clusters, inevitably leads to some degradation in utility of DP methods. Intuitively, the small count differences are masked by DP noise, leading to inaccurate quality evaluations of the histograms. However, this effect is not observed in Figure 8a for the Census dataset as it is larger.

Quality for different cluster sizes. We study the impact of varying the average cluster size on the quality scores. (Figure 8b). A subset comprising η fraction of the tuples in each cluster is sampled, where η ranges from 10^{-3} to 1 (full dataset), and an explanation is generated for the sampled data. Figure 8b shows the results for the Census and Diabetes datasets under k-means clustering, while for other cases the results exhibited similar trends. We find that the non-private TabEE baseline maintains a stable explanation quality, while the DP methods exhibit a decrease in quality with smaller cluster sizes. For the Diabetes dataset, at $\eta = 10^{-0.5}$, the average cluster size is 6436, and DPClustX performs comparably to TabEE, with a minor difference of 0.2%, while DP-Naive scores 15% lower, and DP-TabEE consistently performs poorly with a 42% difference. At $\eta = 10^{-1}$, the average cluster size is 2035, and DPClustX's score decreases by 20%. For the Census dataset, DPClustX performs comparably to the non-private TabEE, with an average difference of 0.09% at $\eta \geq 10^{-2}$ and an average cluster size of ≥ 4917. At $\eta = 10^{-2.5}$, the average cluster size is 1555, and the score decreases by 12%. In all cases, DPClustX consistently outperforms or matches the two DP baselines with smaller cluster sizes.

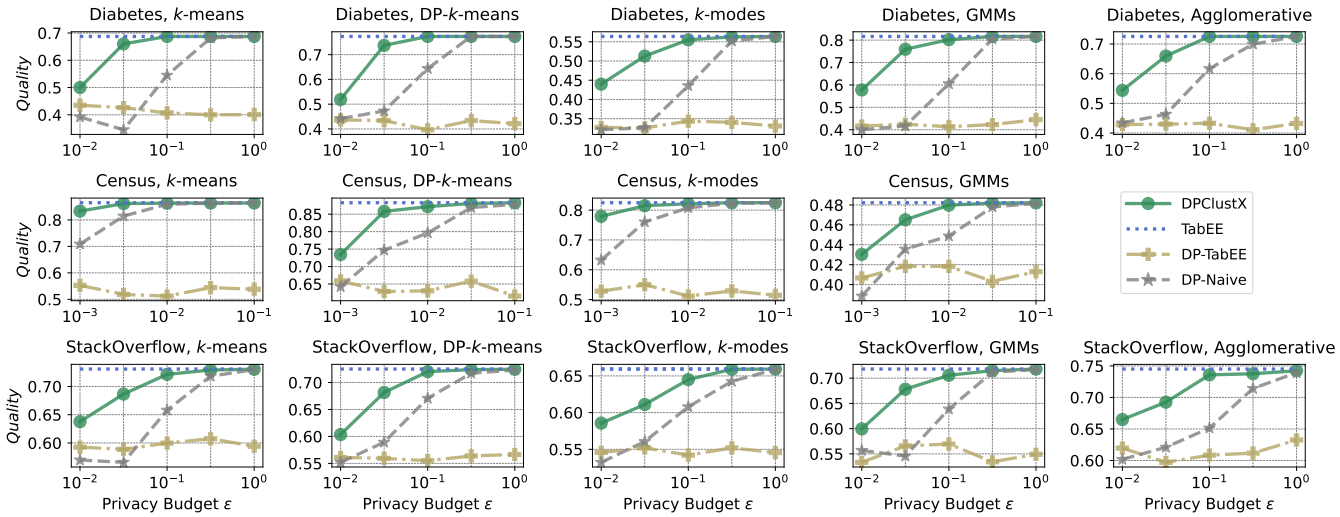


Figure 5: *Quality* values of the selected attribute combination as the total privacy budget ε varies. Note that the range of the *Quality* axis differs across methods, reflecting the substantial variation in explanation quality between clustering approaches.

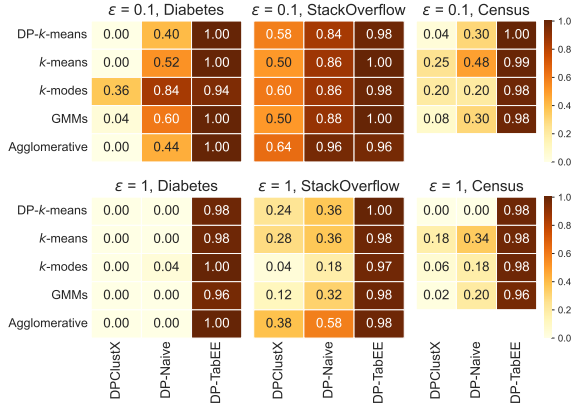


Figure 6: Mean Absolute Error (MAE) values of the selected attribute combination compared to the non-private TabEE baseline [8], as the total privacy budget ε varies.

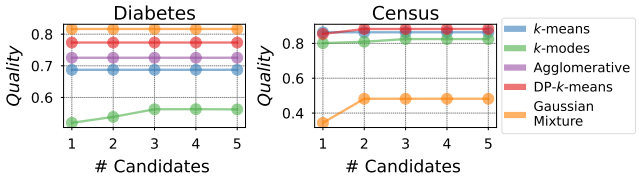
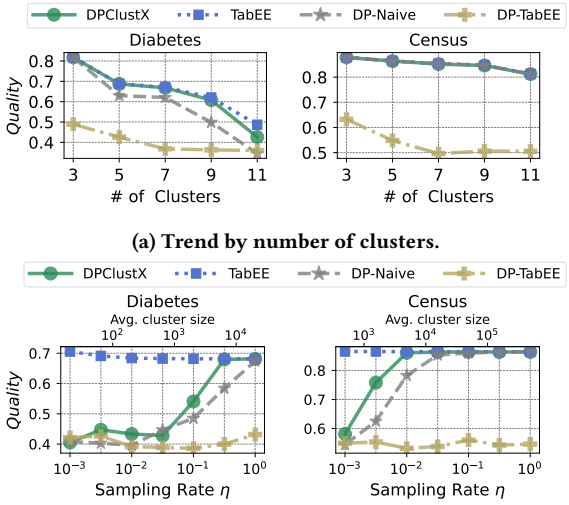


Figure 7: Explanation *Quality* values for DPClustX with a varying number of candidate attributes per cluster selected at Stage-1 (Section 5.1).

Impact of attribute correlations on quality. We assess the stability of DPClustX in the presence of attribute correlations by adopting the experimental setting of [8]. We cluster the datasets after adding attributes that are highly correlated with the original ones. Specifically, for each original attribute, we generate a corresponding correlated attribute by randomly perturbing a small portion of the records, while maintaining a Cramer's V value of 0.85, a



(a) Trend by number of clusters.

(b) Trend by sample size. The bottom x-axis indicates the sampling rate, and the top x-axis shows the corresponding average cluster size.

Figure 8: *Quality* values for the selected attribute combination as the number of clusters varies (top) and as the sample size per cluster varies (bottom). Note the differing *Quality* axis ranges.

standard association measure [9] between attributes. Then, we run DPClustX twice, once with the extended set of attributes and once with the original set, and compare the *Quality* scores of the selected attributes in both scenarios. We find that, for all datasets, the difference is below 2% on average, indicating a minor change. This difference is mostly due to the diversity measure, as an attribute and its noisy counterpart are considered different, hence selecting both maximizes diversity. In contrast, while running with the original attribute set, DPClustX may select the same attribute twice, potentially contributing less to diversity. Considering only

interestingness and sufficiency, the difference is below 0.1% in all cases.

6.3 Performance Analysis

We measure the execution time of our algorithm with various parameter settings for both datasets. We focus on k-means and GMMs clustering methods, as other methods were unable to scale to a large number of clusters or returned mostly empty clusters when the number of clusters exceeded 9. The default settings for these experiments are 9 clusters, 3 candidate attributes per cluster, and the entire dataset with all attributes used. Each experiment varies one of these parameters and is averaged over 10 runs.

Number of clusters. Figure 9a illustrates the relationship between the execution time (in seconds, log scale) and the number of clusters for which an explanation is generated. We observe that the running time increases exponentially with the number of clusters, yet remains reasonably low across both methods when the number of clusters is at most 11. For the larger Census dataset, the runtimes for up to 9 clusters across both methods are less than 6.6 seconds. For the Diabetes and Stack Overflow datasets, all runtimes are below 3 seconds up to 9 clusters, exhibiting a similar trend. For 15 clusters, the runtimes are up to 1275 seconds for Stack Overflow and Diabetes, and 1314 seconds for Census.

Number of candidate attributes per cluster. Figure 9b illustrates the relationship between the execution time (in seconds, log scale) and the size of the candidate set constructed by Algorithm 1 for each cluster. The results indicate a significant execution time increase with larger sizes, highlighting the benefit of selecting a smaller size. The execution times remain below 3 seconds with at most 3 candidates for the Diabetes and Stack Overflow datasets, and under 7 seconds for the Census dataset. However, with 5 candidates, execution times rise up to 80 seconds for Census, 75 seconds for Diabetes, and 69 seconds for Stack Overflow.

Number of attributes. In this experiment (Figure 9c), we randomly sample a subset of the attributes for each dataset, and generate an explanation using only the sampled attributes. Figure 9c depicts the relationship between running time (in seconds) and the percentage of attributes selected from the dataset. The results indicate a linear increase in running time, implying that while the number of attributes has an effect, it is relatively small. For instance, with a 50% sample, the execution times are at most 4.6 seconds for the Census dataset and 2 seconds for Diabetes and Stack Overflow, while with a 100% sample (full dataset), the execution times increase to a maximum of 6.3 seconds for Census and at most 2.9 seconds for Diabetes and Stack Overflow.

Dataset size. In this experiment (Figure 9d), we randomly sample a portion of the tuples for each dataset, and generate an explanation using the sampled data. Figure 9d shows the relationship between execution time (in seconds) and the percentage of tuples sampled from each dataset. The results indicate a linear increase in running time, suggesting that while an increase in the number of tuples leads to longer execution times, the effect remains relatively small. With a 50% sample, the execution times are at most 4.8 seconds for the Census dataset and below 2.6 seconds for Diabetes and Stack Overflow. With a 100% sample (full dataset), the execution times are at most 6.3 seconds for Census and below 2.9 seconds for Diabetes and Stack Overflow.

6.4 Case Study

We now present a case study over the Census dataset with default parameters. The dataset is clustered into 3 clusters using k-means.

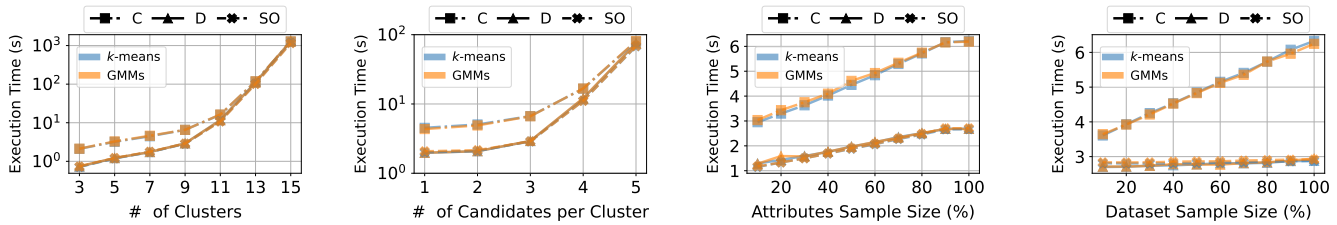
DPClustX selects the attributes `iRlabor`, which represents employment status, `iWork89`, which indicates whether the individual worked in 1989, and `dHours`, which denotes the number of working hours in the previous week. The final output is shown in Figure 10a. Figure 10b shows the non-private explanation generated by TabEE. In this case, the selected attributes are `iRlabor`, `iYearwrk`, which specifies the last year the individual worked, and `iMeans`, which describes the means of transportation to work. Since the two explanations agree on 1 out of 3 attributes, DPClustX achieves MAE = 2/3. However, the *Quality* of the non-private baseline is only 0.04% higher, and the explanation generated by DPClustX conveys similar insights.

Both explanations indicate that Cluster 1 predominantly consists of adults who are currently not working, and Cluster 2 of individuals under age 16, for whom data is unavailable. In this group, the attributes `iWork89` and `iYearwrk` are correlated. Both DPClustX and TabEE reveal that Cluster 3 primarily consists of working individuals, as they worked more than 0 hours last week and have means of transportation to work. Thus, a similar conclusion is reached, though different attributes are used, which are correlated among non-working individuals. Note that DPClustX is not guaranteed to select attributes most correlated with those chosen by TabEE in general.

7 Related Work

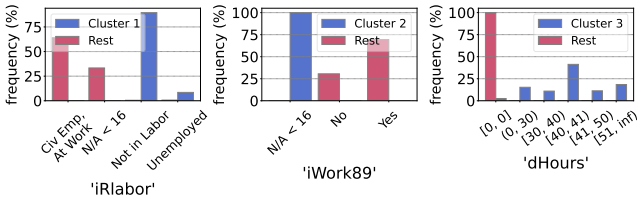
We next survey related work in relevant fields.

Differentially private explanations. Several works have explored integrating DP with ML models and query result explanation methods to address the dual challenge of transparency and privacy. The work of [58] introduced differentially private mechanisms for model explanations, providing interpretable insights into model behavior while satisfying DP. Their approach adapts traditional explanation methods such as feature importance scores and local interpretable model-agnostic explanations (LIME) to operate under DP constraints, thereby ensuring that the explanations themselves do not leak sensitive information. Additionally, significant contributions have been made in creating a framework for generating differentially private Shapley values, enabling the interpretation of model predictions with strong privacy guarantees [35]. Further approaches include a DP SVM mechanism for robust counterfactual explanations [50] and DP Locally Linear Maps, which provide both local and global model explanations while enabling a favorable privacy-accuracy trade-off by efficiently managing the number of parameters [28]. See [55] for a survey on privacy-preserving model explanations. In the context of query result explanation, DPXPlain [68] is a framework that generates differentially private explanations for aggregate queries based on the notion of intervention. These efforts represent crucial steps towards creating trustworthy ML and data analysis systems that offer both transparency and

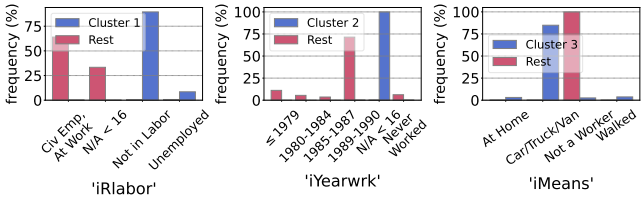


(a) Trend by number of clusters (log-scale). **(b) Trend by size of a single cluster candidate set (log-scale).** **(c) Trend by attributes sample size.** **(d) Trend by dataset sample size.**

Figure 9: DPClustX**'s execution time trends for the Census (C), Diabetes (D), and Stack Overflow (SO) datasets by different parameters.**



(a) DPClustX **explanation.**



(b) Non-private TabEE [8] explanation.

Figure 10: Explanations for the US Census dataset case study.

robust privacy guarantees, thus addressing the critical need for ethical AI deployment in sensitive applications.

Clustering explanations. In the non-private setting, clustering explanations are well-studied. For instance, [21, 23, 44, 45, 51] study tree-based interpretable clustering algorithms specifically for k-means or k-medians. Tutay and Somech [69] provides black-box clustering explanations using filter predicates. Copul et al. [8] use histogram-based explanations for clusters in tabular datasets, induced by clusters formed in the tabular-embedding space. However, *differentially-private* explanation algorithms for clustering have not been explored beyond our work.

Private data summarization and exploration tools. In the realm of databases, privacy-preserving techniques have been a focal point of research, particularly with the advent of differential privacy (DP). In recent years, DP has been increasingly adopted to practical systems for interactive data analysis, such as PINQ [48], PrivateSQL [36], FLEX [34], and Chorus [33]. Another significant contribution is the DPCube framework by Xiao et al., which enables efficient and private OLAP data cube construction, by differentially private histogram release through multidimensional partitioning, thereby enhancing the usability of summarized data under privacy constraints [73]. The RONA algorithm by McKenna et al. improves the accuracy of DP synthetic data generation through an iterative refinement process, supporting private data release [46]. Additionally, Zhang et al. introduced PrivBayes, a method for generating differentially private synthetic data using Bayesian networks to capture correlations in the data while preserving privacy [76]. In the context of data exploration, APEx [24] is a system that allows data analysts to pose adaptively chosen queries along with required accuracy bounds, identifying algorithms with the least privacy loss to answer these queries accurately under DP.

These advancements collectively highlight the progress and ongoing challenges in developing privacy-preserving database systems that balance data utility with stringent privacy requirements.

8 Conclusion and Future Work

We proposed DPClustX, a framework for generating global, histogram based explanations for black-box clustering results while preserving differential privacy. These explanations consist of histograms on carefully selected attributes for each cluster and the remaining dataset, highlighting significant distributional shifts alongside additional high-quality characteristics. We demonstrated through extensive experiments that the explanations generated by DPClustX are comparable to the non-privately generated explanations even under tight privacy budgets.

There are several interesting future directions. First, the current framework outputs one explanation per cluster, aligning with the non-private [8]. It can be generalized to output multiple explanations per cluster for a more comprehensive understanding, but complexity may increase (see Appendix B). Extending the framework to efficiently output multiple explanations per cluster is important future work. Second, DPClustX uses one-dimensional histograms due to their simplicity and interpretability, building upon existing work. One possible way to extend DPClustX to higher-dimensional histograms is by considering the Cartesian product of the domains. However, such an extension is not straightforward, as it comes at the cost of increased complexity, and may result in histograms where all counts are small, making it challenging to accurately compute them under DP. Examining the quality and interpretability of such explanations is an interesting direction. Third, it would be intriguing to examine the impact of different discretization and binning approaches on the performance of our system. Fourth, the extension of DPClustX to different score functions that emphasize different facets of explainability would be an interesting direction. These directions will enhance the usability and interpretability of clustering methods under DP.

Acknowledgments

This research was supported by the Israel Science Foundation (ISF) under grant 2707/22 of the Breakthrough Research Grant (BRG) Program. The work of Amir Gilad was funded by the Israel Science Foundation (ISF) under grant 1702/24, the Scharf-Ullman Endowment, and the Alon Scholarship.

References

[1] 2024. DPClustX Git Repository. https://github.com/ronzadi/DPClustX

[2] Karim Abouelmehdi, Abderrahim Beni Hssane, and Hayat Khaloufi. 2018. Big healthcare data: preserving security and privacy. *J. Big Data* 5 (2018), 1. https://doi.org/10.1186/S40537-017-0110-7

[3] John M. Abowd. 2018. The U.S. Census Bureau Adopts Differential Privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, 2867. https://doi.org/10.1145/3219819.3226070

[4] Gergely Acs, Claude Castelluccia, and Rui Chen. 2012. Differentially private histogram publishing through lossy compression. In *2012 IEEE 12th International Conference on Data Mining*. IEEE, 1–10.

[5] Sihem Amer-Yahia, Tova Milo, and Brit Youngmann. 2021. Exploring ratings in subjective databases. In *Proceedings of the 2021 International Conference on Management of Data*. 62–75.

[6] Allan Borodin, Aadhar Jain, Hyun Chul Lee, and Yuli Ye. 2017. Max-sum diversification, monotone submodular functions, and dynamic updates. *ACM Transactions on Algorithms (TALG)* 13, 3 (2017), 1–25.

[7] John Clore, Krzysztof Cios, Jon DeShazo, and Beata Strack. 2014. Diabetes 130-US Hospitals for Years 1999–2008. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5230J.

[8] Roni Copul, Nave Frost, Tova Milo, and Kathy Razmadze. 2024. TabEE: Tabular Embeddings Explanations. *Proceedings of the ACM on Management of Data* 2, 1 (2024), 1–26.

[9] Harald Cramér. 1999. *Mathematical methods of statistics*. Vol. 43. Princeton university press.

[10] Sanjoy Dasgupta, Nave Frost, and Michal Moshkovitz. 2022. Framework for evaluating faithfulness of local explanations. In *International Conference on Machine Learning*. PMLR, 4794–4815.

[11] Daniel Deutch, Amir Gilad, Tova Milo, Amit Mualem, and Amit Somech. 2022. FEDEX: An Explainability Framework for Data Exploration Steps. *Proceedings of the VLDB Endowment* 15, 13 (2022), 3854–3868.

[12] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. 2017. Collecting Telemetry Data Privately. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 3574–3583.

[13] Jinshuo Dong, David Durfee, and Ryan Rogers. 2020. Optimal differential privacy composition for exponential mechanisms. In *International Conference on Machine Learning*. PMLR, 2597–2606.

[14] D. Durfee and R. Rogers. 2021. One-shot DP top-k mechanisms. DifferentialPrivacy.org. https://differentialprivacy.org/one-shot-top-k/.

[15] David Durfee and Ryan M Rogers. 2019. Practical differentially private top-k selection with pay-what-you-get composition. *Advances in Neural Information Processing Systems* 32 (2019).

[16] Cynthia Dwork. 2006. Differential privacy. In *International colloquium on automata, languages, and programming*. Springer, 1–12.

[17] Cynthia Dwork. 2019. Differential Privacy and the US Census. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems* (Amsterdam, Netherlands) *(PODS '19)*. Association for Computing Machinery, New York, NY, USA, 1. https://doi.org/10.1145/3294052.3322188

[18] 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.

[19] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

[20] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (Scottsdale, Arizona, USA) *(CCS '14)*. Association for Computing Machinery, New York, NY, USA, 1054–1067. https://doi.org/10.1145/2660267.2660348

[21] Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan. 2022. Almost tight approximation algorithms for explainable clustering. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2641–2663.

[22] Sandra Gabriele and Sonia Chiasson. 2020. Understanding Fitness Tracker Users' Security and Privacy Knowledge, Attitudes and Behaviours. In *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*, Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjøn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik (Eds.). ACM, 1–12. https://doi.org/10.1145/3313831.3376651

[23] Buddhima Gamlath, Xinrui Jia, Adam Polak, and Ola Svensson. 2021. Nearly-tight and oblivious algorithms for explainable clustering. *Advances in Neural Information Processing Systems* 34 (2021), 28929–28939.

[24] Chang Ge, Xi He, Ihab F Ilyas, and Ashwin Machanavajjhala. 2019. Apex: Accuracy-aware differentially private data exploration. In *Proceedings of the 2019 International Conference on Management of Data*. 177–194.

[25] Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. 2020. Differentially private clustering: Tight approximation ratios. *Advances in Neural Information Processing Systems* 33 (2020), 4040–4054.

[26] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. 2009. Universally utility-maximizing privacy mechanisms. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 351–360.

[27] Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. 2010. Differentially private combinatorial optimization. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 1106–1125.

[28] Frederik Harder, Matthias Bauer, and Mijung Park. 2020. Interpretable and differentially private predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 4083–4090.

[29] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. 2010. Boosting the Accuracy of Differentially Private Histograms Through Consistency. *Proceedings of the VLDB Endowment* 3, 1 (2010).

[30] Robert J Hilderman and Howard J Hamilton. 2013. *Knowledge Discovery and Measures of Interest*. Vol. 638. Springer Science & Business Media.

[31] Naoise Holohan, Stefano Braghin, Pól Mac Aonghusa, and Killian Levacher. 2019. Diffprivlib: the IBM differential privacy library. *arXiv preprint arXiv:1907.02444* (2019).

[32] Lianyu Hu, Mudi Jiang, Junjie Dong, Xinying Liu, and Zengyou He. 2024. Interpretable Clustering: A Survey. *arXiv preprint arXiv:2409.00743* (2024).

[33] Noah Johnson, Joseph P Near, Joseph M Hellerstein, and Dawn Song. 2020. Chorus: a programming framework for building scalable differential privacy mechanisms. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 535–551.

[34] Noah Johnson, Joseph P Near, and Dawn Song. 2018. Towards practical differential privacy for SQL queries. *Proceedings of the VLDB Endowment* 11, 5 (2018), 526–539.

[35] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. 2019. Differentially private model personalization. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR.

[36] Ios Kotsogiannis, Yuchao Tao, Xi He, Maryam Fanaeepour, Ashwin Machanavajjhala, Michael Hay, and Gerome Miklau. 2019. Privatesql: a differentially private sql query engine. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1371–1384.

[37] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. 2016. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1675–1684.

[38] Doris Jung-Lin Lee, Dixin Tang, Kunal Agarwal, Thyne Boonmark, Caitlyn Chen, Jake Kang, Ujjaini Mukhopadhyay, Jerry Song, Micah Yong, Marti A Hearst, et al. 2021. Lux: always-on visualization recommendations for exploratory dataframe workflows. *PVLDB* 15, 3 (2021), 727–738.

[39] David A Levin and Yuval Peres. 2017. *Markov chains and mixing times*. Vol. 107. American Mathematical Soc.

[40] Bing-Rong Lin and Daniel Kifer. 2013. Information preservation in statistical privacy and bayesian estimation of unattributed histograms. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. 677–688.

[41] Jianhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory* 37, 1 (1991), 145–151.

[42] Yuyu Luo, Xuedi Qin, Nan Tang, and Guoliang Li. 2018. DeepEye: Towards Automatic Data Visualization. ICDE.

[43] Ge Lv and Lei Chen. 2023. On data-aware global explainability of graph neural networks. *Proceedings of the VLDB Endowment* 16, 11 (2023), 3447–3460.

[44] Konstantin Makarychev and Liren Shan. 2021. Near-optimal algorithms for explainable k-medians and k-means. In *International Conference on Machine Learning*. PMLR, 7358–7367.

[45] Konstantin Makarychev and Liren Shan. 2022. Explainable k-means: don't be greedy, plant bigger trees!. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. 1629–1642.

[46] Ryan McKenna, Gerome Miklau, and Daniel Sheldon. 2021. Winning the NIST Contest: A scalable and general approach to differentially private synthetic data. *arXiv preprint arXiv:2108.04978* (2021).

[47] Frank McSherry and Kunal Talwar. 2007. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. IEEE, 94–103.

[48] Frank D McSherry. 2009. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD*

International Conference on Management of data. 19–30.

[49] Chris Meek, Bo Thiesson, and David Heckerman. 2001. US Census Data (1990). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5VP42.

[50] Rami Mochaourab, Sugandh Sinha, Stanley Greenstein, and Panagiotis Papapetrou. 2021. Robust counterfactual explanations for privacy-preserving SVM. In *International Conference on Machine Learning (ICML 2021), Workshop on Socially Responsible Machine Learning*.

[51] Michal Moshkovitz, Sanjoy Dasgupta, Cyrus Rashtchian, and Nave Frost. 2020. Explainable k-means and k-medians clustering. In *International conference on machine learning*. PMLR, 7055–7065.

[52] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 607–617.

[53] Huu Hiep Nguyen. 2018. Privacy-preserving mechanisms for k-modes clustering. *Computers & Security* 78 (2018), 60–75.

[54] Huy L Nguyen, Anamay Chaturvedi, and Eric Z Xu. 2021. Differentially private k-means via exponential mechanism and max cover. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 9101–9108.

[55] Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Thanh Toan Nguyen, Phi Le Nguyen, Hongzhi Yin, and Quoc Viet Hung Nguyen. 2024. A survey of privacy-preserving model explanations: Privacy risks, attacks, and countermeasures. *arXiv preprint arXiv:2404.00673* (2024).

[56] Stack Overflow. 2018. Stack Overflow Annual Developer Survey. https://survey.stackoverflow.co.

[57] Adedoyin Tolulope Oyewole, Bisola Beatrice Oguejiofor, Nkechi Emmanuella Eneh, Chidiogo Uzoamaka Akpuokwe, and Seun Solomon Bakare. 2024. Data privacy laws and their impact on financial technology companies: a review. *Computer Science & IT Research Journal* 5, 3 (2024), 628–650.

[58] Neel Patel, Reza Shokri, and Yair Zick. 2022. Model explanations with differential privacy. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. 1895–1904.

[59] Wahbeh Qardaji, Weining Yang, and Ninghui Li. 2013. Understanding hierarchical methods for differentially private histograms. *Proceedings of the VLDB Endowment* 6, 14 (2013), 1954–1965.

[60] Kaivalya Rawal and Himabindu Lakkaraju. 2020. Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. *Advances in Neural Information Processing Systems* 33 (2020), 12187–12198.

[61] Sunita Sarawagi, Rakesh Agrawal, and Nimrod Megiddo. 1998. Discovery-driven exploration of OLAP data cubes. In *EDBT*.

[62] Uri Stemmer and Haim Kaplan. 2018. Differentially private k-means with constant multiplicative error. *Advances in Neural Information Processing Systems* 31 (2018).

[63] Beata Strack, Jonathan P DeShazo, Chris Gennings, Juan L Olmo, Sebastian Ventura, Krzysztof J Cios, and John N Clore. 2014. Impact of HbA1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. *BioMed research international* 2014, 1 (2014), 781670.

[64] Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, and Hongxia Jin. 2016. Differentially private k-means clustering. In *Proceedings of the sixth ACM conference on data and application security and privacy*. 26–37.

[65] Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, Min Lyu, and Hongxia Jin. 2017. Differentially private k-means clustering and a hybrid approach to private optimization. *ACM Transactions on Privacy and Security (TOPS)* 20, 4 (2017), 1–33.

[66] Bo Tang, Shi Han, Man Lung Yiu, Rui Ding, and Dongmei Zhang. 2017. Extracting top-k insights from multi-dimensional data. In *Proceedings of the 2017 ACM International Conference on Management of Data*. 1509–1524.

[67] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and XiaoFeng Wang. 2017. Privacy Loss in Apple's Implementation of Differential Privacy on MacOS 10.12. *CoRR* abs/1709.02753 (2017). arXiv:1709.02753 http://arxiv.org/abs/1709.02753

[68] Yuchao Tao, Amir Gilad, Ashwin Machanavajjhala, and Sudeepa Roy. 2022. DPXPlain: privately explaining aggregate query answers. *Proceedings of the VLDB Endowment* 16, 1 (2022), 113–126.

[69] Sariel Tutay and Amit Somech. 2023. Cluster-Explorer: An interactive Framework for Explaining Black-Box Clustering Results. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (Birmingham, United Kingdom) *(CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 5106–5110. https://doi.org/10.1145/3583780.3614734

[70] Marcos R Vieira, Humberto L Razente, Maria CN Barioni, Marios Hadjieleftheriou, Divesh Srivastava, Caetano Traina, and Vassilis J Tsotras. 2011. On query result diversification. In *2011 IEEE 27th International Conference on Data Engineering*. IEEE, 1163–1174.

[71] Cort J Willmott and Kenji Matsuura. 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate research* 30, 1 (2005), 79–82.

[72] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *TVCG* (2016).

[73] Yonghui Xiao, Li Xiong, Liyue Fan, and Slawomir Goryczka. 2012. DPCube: Differentially private histogram release through multidimensional partitioning. *arXiv preprint arXiv:1202.5358* (2012).

[74] Jia Xu, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, Ge Yu, and Marianne Winslett. 2013. Differentially private histogram publication. *The VLDB journal* 22 (2013), 797–822.

[75] Brit Youngmann, Sihem Amer-Yahia, and Aurelien Personnaz. 2022. Guided exploration of data summaries. *Proceedings of the VLDB Endowment (PVLDB)* 15, 9 (2022), 1798–1807.

[76] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2017. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)* 42, 4 (2017), 1–41.

A Theorems and Proofs

Quality functions for histogram-based explanations used in prior work on are closely related to the L_1 distance between histograms, viewed as vectors. This connection will also be used in our analysis. Hence, we now formally define these concepts. A histogram $h_A(D)$ can be viewed as a $|\text{dom}(A)|$-dimensional vector with value $cnt_{A=a}(D)$ in its a'th entry for every $a \in \text{dom}(A)$. The L_1-norm of a vector $v \in \mathbb{R}^d$, denoted $\|v\|_1$, is defined as $\sum_{i=1}^{d} |v_i|$. For histogram vectors, we always have $\|h_A(D)\|_1 = |D|$, as the L_1-norm is simply the sum of counts of all domain elements. The following is implied immediately from the definition:

Corollary A.1. *For a dataset D and a clustering function $f : \text{dom}(R) \to C$, a cluster label c, and an attribute A, the interestingness score (Definition 4.3) is*

$$\text{Int}_p(D, f, c, A) = \frac{1}{2} \left\| h_A(D_c) - \frac{|D_c|}{|D|} h_A(D) \right\|_1$$

Corollary A.2. *Let D be a dataset, f a clustering function, A an attribute, and $D_c, D_{c'}$ non-empty clusters such that $|D_c| \leq |D_{c'}|$. The pairwise-diversity score (Definition 4.8) is*

$$d(D, f, c, c', A, A) = \frac{1}{2} \left\| h_A(D_c) - \frac{|D_c|}{|D_{c'}|} h_A(D_{c'}) \right\|_1$$

Our global quality measures are defined as the average of low-sensitivity functions. Consequently, our analysis frequently uses the following Lemma

Lemma A.3. *For $i = 1, \ldots, m$ let $\alpha_i \in \mathbb{R}$ be a scalar and $f_i : \mathcal{D} \to \mathbb{R}$ be a function with sensitivity Δ_i. Then, the function $f = \sum_{i=1}^{m} \alpha_i \cdot f_i$ has sensitivity bounded by $\sum_{i=1}^{m} |\alpha_i| \Delta_i$.*

Proof. Let $D \sim D'$ be two neighboring datasets. By the triangle inequality,

$$\begin{aligned} \left| f(D) - f(D') \right| &= \left| \sum_{i=1}^{m} \alpha_i \cdot f_i(D) - \sum_{i=1}^{m} \alpha_i \cdot f_i(D') \right| \\ &\leq \sum_{i=1}^{m} |\alpha_i| \cdot \left| f_i(D) - f_i(D') \right| \\ &\leq \sum_{i=1}^{m} |\alpha_i| \Delta_i. \end{aligned}$$

□

A.1 Interestingness

We prove next the high sensitivity of the total variation distance interestingness measure (1), restated below for convenience.

$$\text{TVD}(\pi_A(D), \pi_A(D_c)) = \frac{1}{2} \sum_{a \in \text{dom}(A)} \left| \frac{cnt_{A=a}(D)}{|D|} - \frac{cnt_{A=a}(D_c)}{|D_c|} \right|$$

Proposition 4.1. *The sensitivity of* TVD *is at least $\frac{1}{2}$ and its range is $[0, 1]$.*

Proof. Since the range bound is standard (see, e.g., [39]), we focus on the sensitivity lower bound. Let D be a dataset of size $n \geq 1$, and A an attribute. Suppose that for all tuples $t \in D$ it holds that $t[A] = a$ for some $a \in \text{dom}(A)$. That is, $cnt_{A=a}(D) = n$. Let $D_c \subseteq D$ be a cluster of size 1. In this case $\text{TVD}(\pi_A(D), \pi_A(D_c)) = 0$, as $\pi_A(D)$ and $\pi_A(D_c)$ define the same distribution (the value a has probability 1).

Now, let $D' = D \cup \{t'\}$ and $D'_c = D_c \cup \{t'\}$ for a tuple t' with $t'[A] = a'$ for $a' \neq a$. We now have

$$\begin{aligned} 2 \cdot \text{TVD}(\pi_A(D'), \pi_A(D'_c)) &= \left| \frac{cnt_{A=a}(D')}{|D'|} - \frac{cnt_{A=a}(D'_c)}{|D'_c|} \right| \\ &\quad + \left| \frac{cnt_{A=a'}(D')}{|D'|} - \frac{cnt_{A=a'}(D'_c)}{|D'_c|} \right| \\ &= \left| \frac{n}{n+1} - \frac{1}{2} \right| + \left| \frac{1}{n+1} - \frac{1}{2} \right| \\ &= \frac{n}{n+1} - \frac{1}{2} + \frac{1}{2} - \frac{1}{n+1} \\ &= 1 - \frac{2}{n+1} \end{aligned}$$

Therefore,

$$\text{TVD}(\pi_A(D'), \pi_A(D'_c)) = \frac{1}{2} - \frac{1}{n+1}$$

Since sensitivity is defined as the supremum over all datasets and dataset sizes, the lemma follows. □

Since previous work has also considered the Jensen-Shannon distance [41] as an interestingness metric, we show that it is highly sensitive as well, making it unsuitable for the DP setting.

Definition A.4 (Jensen-Shannon Divergence [41]). For two distributions p and q over the same domain, their Jensen–Shannon *divergence* is defined as

$$JSD(p, q) = H\left(\frac{p+q}{2}\right) - \frac{1}{2}H(q) - \frac{1}{2}H(q)$$

where $H(\cdot)$ is the Shannon entropy and $\frac{p+q}{2}$ is the mixture distribution of p and q. The Jensen–Shannon *distance*, denoted d_{JS}, is the square root of the Jensen–Shannon divergence.

Let $P_A(D)$ and $P_A(D_c)$ be the probability distributions of the values in the columns of $\pi_A(D)$ and $\pi_A(D_c)$, respectively, determined by the relative frequencies of each value. Define $d_{JS}(P_A(D), P_A(D_c))$ as the Jensen-Shannon distance between these distributions. We prove the following:

Proposition A.5. *The sensitivity of d_{JS} is at least $\frac{1}{2}$ and its range is $[0, 1]$.*

Proof. The proof for the range bound can be found in [41]. We proceed with the sensitivity analysis. Let D be a dataset of size n, and A an attribute. Suppose that for all tuples $t \in D$ it holds that $t[A] = a$ for some $a \in \text{dom}(A)$. Let $D_c = \{t\}$ be a cluster of size 1. In this case, $d_{JS}(P_A(D), P_A(D_c)) = 0$, as $P_A(D)$ and $P_A(D_c)$ are the same distribution (the value a has probability 1). Now, let $D' = D \cup \{t'\}$ and $D'_c = D_c \cup \{t'\}$ for a tuple t' with $t'[A] = a'$ where $a' \neq a$. We now have

$$P_A(D') = \begin{cases} a & \text{w.p } \frac{n}{n+1} \\ a' & \text{w.p } \frac{1}{n+1} \end{cases}$$

and

$$P_A(D'_c) = \begin{cases} a & \text{w.p } \frac{1}{2} \\ a' & \text{w.p } \frac{1}{2} \end{cases}$$

therefore, the mixture distribution satisfies

$$\frac{P_A(D') + P_A(D'_c)}{2} = \begin{cases} a & \text{w.p } \frac{n}{2(n+1)} + \frac{1}{4} \\ a' & \text{w.p } \frac{1}{2(n+1)} + \frac{1}{4} \end{cases}$$

Since all three distributions are supported on $\{a, a'\}$, we have

$$JSD(P_A(D'), P_A(D'_c)) = H_b\left(\frac{1}{2(n+1)} + \frac{1}{4}\right) - \frac{H_b\left(\frac{1}{n+1}\right) + H_b\left(\frac{1}{2}\right)}{2}$$

where H_b is the binary entropy function. Note that by continuity of H_b we have

$$\lim_{n\to\infty} JSD(P_A(D'), P_A(D'_c)) = H_b\left(\frac{1}{4}\right) - \frac{1}{2} \approx 0.311$$

In particular, for sufficiently large n, $JSD(P_A(D'), P_A(D'_c)) > 0.3$, and hence $d_{JS}(P_A(D'), P_A(D'_c)) > \frac{1}{2}$. Therefore, we find that d_{JS} has sensitivity greater than $\frac{1}{2}$. □

Intuitively, since the range of d_{JS} is $[0, 1]$ [41], its sensitivity is relatively high. We proceed with the analysis of our low sensitivity interestingness function.

Proposition 4.4. *The sensitivity of* $\mathrm{Int}_p(D, f, c, A)$ *is* 1 *and its range is* $[0, |D_c|]$.

Proof. Let $f : \mathrm{dom}(R) \to C$ be a clustering function, and $c \in C$ be a cluster label. Let $A \in \mathcal{A}$ and $D \sim D'$ be neighboring datasets such that $D' = D \cup \{t\}$. Denote $a = t[A]$, and let $h_a = h_A(t)$ be the $|\mathrm{dom}(A)|$-dimensional histogram with 1 in its a'th entry and 0 elsewhere. Let D_c (respectively, D'_c) denote the set of tuples in D (respectively, D') that are mapped to c by the function f, and note that D'_c is either D_c or $D_c \cup \{t\}$.

We first consider the case that $D_c = D'_c$. By Corollary A.1 and the triangle inequality, we have

$$\begin{aligned} &\left|\mathrm{Int}_p(D', f, c, A) - \mathrm{Int}_p(D, f, c, A)\right| \\ &= \frac{1}{2}\left|\left\|h_A(D_c) - \frac{|D_c|}{|D'|}h_A(D')\right\|_1 - \left\|h_A(D_c) - \frac{|D_c|}{|D|}h_A(D)\right\|_1\right| \\ &\le \frac{1}{2}\left\|\frac{|D_c|}{|D|}h_A(D) - \frac{|D_c|}{|D'|}h_A(D')\right\|_1 \end{aligned}$$

Substituting $h_A(D') = h_A(D) + h_a$ into the inequality above and applying the triangle inequality again, we obtain

$$\begin{aligned} &= \frac{1}{2}\left\|\frac{|D_c|}{|D|+1}h_a - \frac{|D_c|}{|D|\,(|D|+1)}h_A(D)\right\|_1 \\ &\le \frac{1}{2}\left(\|h_a\|_1 + \frac{1}{|D|}\,\|h_A(D)\|_1\right) \end{aligned}$$

where for the equality, we used the fact that $|D'| = |D| + 1$, which implies $\frac{|D_c|}{|D|} - \frac{|D_c|}{|D'|} = \frac{|D_c|}{|D|(|D|+1)}$. For the inequality, we also bound $|D_c| \le |D| + 1$. Next, we substitute $\|h_a\|_1 = 1$ and $\|h_A(D)\|_1 = |D|$ to obtain

$$= \frac{1}{2}\left(1 + \frac{|D|}{|D|}\right) = 1$$

Now, consider the case that $D'_c = D_c \cup \{t\}$. Applying the triangle inequality, we have

$$\begin{aligned} &\left|\mathrm{Int}_p(D', f, c, A) - \mathrm{Int}_p(D, f, c, A)\right| \\ &\le \frac{1}{2}\left\|h_A(D'_c) - \frac{|D'_c|}{|D'|}h_A(D') - h_A(D_c) + \frac{|D_c|}{|D|}h_A(D)\right\|. \end{aligned}$$

Substituting $h_A(D') = h_A(D) + h_a$ and $h_A(D'_c) = h_A(D_c) + h_a$, and rearranging, we obtain

$$\begin{aligned} &= \frac{1}{2}\left\|h_a - \frac{|D'_c|}{|D'|}h_A(D) - \frac{|D'_c|}{|D'|}h_a + \frac{|D_c|}{|D|}h_A(D)\right\| \\ &= \frac{1}{2}\left\|\left(1 - \frac{|D'_c|}{|D'|}\right)h_a - \left(\frac{|D'_c|}{|D'|} - \frac{|D_c|}{|D|}\right)h_A(D)\right\| \end{aligned}$$

Recall that $|D'_c| = |D_c| + 1$ and $|D'| = |D| + 1$. Therefore, $|D'_c|/|D'| - |D_c|/|D| = (|D| - |D_c|)/(|D|\,(|D|+1))$, which is non-negative, as $|D_c| \le |D|$. Since $|D'_c| \le |D'|$ holds as well, we have

$$\begin{aligned} &\le \frac{1}{2}\left(1 - \frac{|D'_c|}{|D'|}\right)\|h_a\|_1 + \frac{1}{2}\left(\frac{|D| - |D_c|}{|D|\,(|D|+1)}\right)\|h_A(D)\| \\ &\le \frac{1}{2}\left(1 - \frac{|D'_c|}{|D'|}\right)\|h_a\|_1 + \frac{1}{2}\left(\frac{1}{|D|+1}\right)\|h_A(D)\|. \end{aligned}$$

Since $\|h_a\|_1 = 1$ and $\|h_A(D)\|_1 = |D|$, we conclude

$$\begin{aligned} &= \frac{1}{2}\left(1 - \frac{|D'_c|}{|D'|}\right) + \frac{1}{2}\left(\frac{1}{|D|+1}\right)|D| \\ &\le \frac{1}{2} + \frac{1}{2} = 1. \end{aligned}$$

For the range upper bound, we use Corollary A.1 to obtain

$$\begin{aligned} \mathrm{Int}_p(D, f, c, A) &= \frac{1}{2}\left\|h_A(D_c) - \frac{|D_c|}{|D|}h_A(D)\right\|_1 \\ &\le \frac{1}{2}\left(\|h_A(D_c)\|_1 + \frac{|D_c|}{|D|}\,\|h_A(D)\|_1\right) \\ &= |D_c| \end{aligned}$$

□

A.2 Sufficiency

We revisit the notion of sufficiency from prior work and provide its sensitivity analysis. For a tuple t, an attribute A, and a cluster label c, Let $r(t, A) = cnt_{A=t[A]}(D_{f(t)})/cnt_{A=t[A]}(D)$. Observe that $r(t, A)$ equals the probability that a uniformly random tuple sampled from D, conditioned on having the same value in attribute A as t, belongs the same cluster as t. The quantity $r(t, A)$ is used in [8] to quantify the extent to which an HBE that employs attribute A to explain a cluster labeled $f(t)$ applies to the tuple t. The local sufficiency of an attribute combination $\mathcal{AC}$ at a tuple t is thus defined as

$$m^s_{\mathcal{AC}}(t) = \frac{\sum_{t' \in D} \mathbf{1}_{\{f(t')=f(t)\}} \cdot r(t', \mathcal{AC}(f(t))),}{\sum_{t' \in D} r(t', \mathcal{AC}(f(t)))}. \quad (2)$$

Let $c = f(t)$ be the cluster of a tuple t, and $A_c = \mathcal{AC}(c)$ the attribute used to explain this cluster by attribute combination $\mathcal{AC}$. Observe that $m^s_{\mathcal{AC}}(t)$ attains its maximum value of 1 when the value $t[A_c]$ appears only in the cluster D_c, capturing the notion of maximal sufficiency.

To measure the *global* sufficiency of $\mathcal{AC}$, [8] averages the local sufficiency across all tuples:

$$\mathrm{Suf}(D, f, \mathcal{AC}) = \frac{1}{|D|} \sum_{t \in D} m^s_{\mathcal{AC}}(t). \tag{3}$$

In the unbounded-DP variant, where the dataset size is not fixed and which we adopt in this work, this function exhibits high sensitivity relative to its range. We remark that this issue does not arise in the bounded-DP variant, as can be seen in the proof of Proposition 4.7. However, there is still motivation to modify this function due to the other reasons outlined in Section 4.2.

Proposition 4.5. *The sensitivity of* $\mathrm{Suf}(D, f, \mathcal{AC})$ *is at least* $\frac{1}{2}$ *and its range is* $[0, 1]$.

Proof. To see that the range is $[0, 1]$, note that by definition in (2), $m^s_{\mathcal{AC}}(t) \in [0, 1]$ for every tuple t, and (3) averages these values across all tuples. We proceed with the sensitivity analysis. Consider a dataset $D = \{t_1\}$ and two clusters $D_1 = \{t_1\}$ and an empty cluster $D_2 = \emptyset$. Let $\mathcal{AC}$ be an attribute combination, and suppose denote $\mathcal{AC}(i) = A$ for $i = 1, 2$. Let us denote $a = t_1[A]$. In this case $\mathrm{Suf}_p(D, f, 1, A) = 1$, as the value a appears only inside the cluster D_1. Hence, using the the equality from item *(1)* of Proposition 4.7,

$$\mathrm{Suf}(D, \mathcal{AC}, f) = \frac{1}{|D|} \sum_{c \in C} \mathrm{Suf}_p(D, f, c, \mathcal{AC}(c)) = 1.$$

Now, suppose a tuple t_2 is added to the cluster D_2. That is, $D' = \{t_1, t_2\}$, $D'_1 = \{t_1\}$ and $D'_2 = \{t_2\}$. Suppose further that $t_2[A] = a$. In this case, for $i \in \{1, 2\}$

$$\mathrm{Suf}_p(D', f, i, A) = \sum_{b \in \mathrm{dom}_{D'_i}(A)} \frac{(cnt_{A=b}(D'_i))^2}{cnt_{A=b}(D')} = \frac{1}{2}.$$

Hence, we find that

$$\begin{aligned}\mathrm{Suf}(D', \mathcal{AC}, f) &= \frac{1}{|D'|} \sum_{c \in \{1,2\}} \mathrm{Suf}_p(D', f, c, \mathcal{AC}(c)) = \frac{1}{2} \cdot (\frac{1}{2} + \frac{1}{2}) \\ &= \frac{1}{2}.\end{aligned}$$

□

Proposition 4.7. *For an attribute combination* $\mathcal{AC}$*,*

(1) The following equality holds:

$$|D| \cdot \mathrm{Suf}(D, f, \mathcal{AC}) = \sum_{c \in C} \mathrm{Suf}_p(D, f, c, \mathcal{AC}(c))$$

(2) The sensitivity of $\mathrm{Suf}_p(D, f, c, A)$ *is* 1 *and its range is* $[0, |D_c|]$.

Proof of *(1)*. Let $\mathcal{AC}$ be an attribute combination. Denote $A_c = \mathcal{AC}(c)$. Observe that the right-hand-side of Equation (2) depends only on the cluster assignment $f(t)$, therefore it is equal across all tuples t of the same cluster. Thus, plugging in the definition of $r(t', A)$,

$$\sum_{t \in D} m^s_E(t) = \sum_{c \in C} |D_c| \cdot \frac{\sum_{t' \in D_c} \frac{cnt_{A_c=t'[A_c]}(D_c)}{cnt_{A_c=t[A_c]}(D)}}{\sum_{t' \in D} \frac{cnt_{A_c=t'[A_c]}(D_c)}{cnt_{A_c=t[A_c]}(D)}}. \tag{4}$$

Let us consider each cluster separately, and observe that each value $cnt_{A_c=a}(D_c)/cnt_{A_c=a}(D)$ appears in the numerator exactly $cnt_{A_c=a}(D_c)$ times, and $cnt_{A_c=a}(D)$ times in the denominator. Hence, by changing the order of summation

$$\begin{aligned}&\frac{\sum_{t \in D_c} \frac{cnt_{A_c=t'[A_c]}(D_c)}{cnt_{A_c=t[A_c]}(D)}}{\sum_{t' \in D} \frac{cnt_{A_c=t'[A_c]}(D_c)}{cnt_{A_c=t[A_c]}(D)}} \\ &= \frac{\sum_{a \in \mathrm{dom}_{D_c}(A)} cnt_{A_c=a}(D_c) \cdot \frac{cnt_{A_c=a}(D_c)}{cnt_{A_c=a}(D)}}{\sum_{a \in \mathrm{dom}_D(A_c)} cnt_{A_c=a}(D) \cdot \frac{cnt_{A_c=a}(D_c)}{cnt_{A_c=a}(D)}} \\ &= \frac{\sum_{a \in \mathrm{dom}_{D_c}(A_c)} \frac{(cnt_{A_c=a}(D_c))^2}{cnt_{A_c=a}(D)}}{\sum_{a \in \mathrm{dom}_D(A_c)} cnt_{A_c=a}(D_c)}\end{aligned}$$

Since the denominator equals $|D_c|$, the following equality holds:

$$\begin{aligned}&= \frac{1}{|D_c|} \sum_{a \in \mathrm{dom}_{D_c}(A_c)} \frac{(cnt_{A_c=a}(D_c))^2}{cnt_{A_c=a}(D)} \\ &= \frac{1}{|D_c|} \mathrm{Suf}_p(D, f, c, A).\end{aligned}$$

The proposition follows by substituting the equality into Equation (4). □

To prove item *(2)* of Proposition 4.7, we introduce the following to lemmas.

Lemma A.6. *The range of* $\mathrm{Suf}_p(D, f, c, A)$ *is* $[0, |D_c|]$.

Proof. We have

$$\begin{aligned}\mathrm{Suf}_p(D, f, c, A) &= \sum_{a \in \mathrm{dom}_{D_c}(A)} \frac{cnt_{A=a}(D_c)^2}{cnt_{A=a}(D)} \\ &\leq \sum_{a \in \mathrm{dom}_{D_c}(A)} cnt_{A=a}(D_c) \\ &= |D_c|\end{aligned}$$

where the inequality holds since $cnt_{A=a}(D_c) \leq cnt_{A=a}(D)$. □

Lemma A.7. *Let* $a, b \in \mathbb{R}$ *such that* $b > 0$ *and* $0 \leq a \leq b$*. Then*

(i) $\left|\frac{a^2}{b} - \frac{(a+1)^2}{b+1}\right| \leq 1$,

(ii) $\left|\frac{a^2}{b} - \frac{a^2}{b+1}\right| \leq 1$

Proof. For item *(i)*, observe that our assumption $a \le b$ implies that $\frac{(a+1)^2}{b+1} \ge \frac{a^2}{b}$. Indeed,

$$\frac{a^2}{b} = \frac{a^2(b+1)}{b(b+1)} = \frac{a^2b + a^2}{b(b+1)} \le \frac{a^2b + 2ab + b}{b(b+1)} = \frac{b(a^2+2a+1)}{b(b+1)} = \frac{(a+1)^2}{b+1}.$$

where the in the inequality we used $a^2 \le ab \le 2ab + b$. Now, we have

$$\frac{(a+1)^2}{b+1} - \frac{a^2}{b} = \frac{b(a+1)^2 - a^2(b+1)}{b(b+1)} = \frac{2ab + b - a^2}{b(b+1)} \le \frac{b^2+b}{b(b+1)} = 1$$

where the inequality follows from $2ab - a^2 \le b^2$ which holds for any $a, b \in \mathbb{R}$.

For *(ii)*, note that

$$\frac{a^2}{b} - \frac{a^2}{b+1} = \frac{a^2}{b(b+1)} < 1$$

as $a \le b$. □

Proof of Proposition 4.7 item *(2)*. Let $f : \mathrm{dom}(R) \to C$ be a clustering function, and $c \in C$ be a cluster label. Fix an attribute A. Let $D \sim D'$ be two neighboring datasets such that $D' = D \cup \{t\}$. Let D_c (respectively, D'_c) denote the set of tuples in D (respectively, D') that are mapped to c by the function f, and note that D'_c is either D_c or $D_c \cup \{t\}$. First, note that if D_c is empty,

$$\left|\mathrm{Suf}_p(D, f, c, A) - \mathrm{Suf}_p(D', f, c, A)\right| \le 1.$$

Indeed, in this case $|D'_c| \le 1$, and by Lemma A.6 we have $\mathrm{Suf}_p(D, f, c, A) = 0$ and $\mathrm{Suf}_p(D', D'_c, A) \le 1$.

Second, if $cnt_{A=t[A]}(D) = 0$, then clearly $cnt_{A=t[A]}(D_c) = 0$ and $cnt_{A=t[A]}(D'_c) \le 1$. In this case

$$\begin{aligned}&\left|\mathrm{Suf}_p(D, f, c, A) - \mathrm{Suf}_p(D', f, c, A)\right| \\ &= \left| \sum_{a \in \mathrm{dom}_{D_c}(A)} \frac{cnt_{A=a}(D_c)^2}{cnt_{A=a}(D)} - \sum_{a \in \mathrm{dom}_{D'_c}(A)} \frac{cnt_{A=a}(D'_c)^2}{cnt_{A=a}(D')} \right| \\ &= \frac{cnt_{A=t[A]}(D'_c)^2}{cnt_{A=t[A]}(D')} \\ &\le 1.\end{aligned}$$

where the second equality holds because all summands are identical in both sums, except for the term corresponding to $a = t[A]$, which appears only in the second sum. The inequality holds since the numerator is at most 1.

Hence, we assume that D_c is not empty and that $cnt_{A=t[A]}(D) > 0$. We consider two cases according to whether or not t is added to D_c.

First assume that $D'_c = D_c$. We have

$$\left|\mathrm{Suf}_p(D, f, c, A) - \mathrm{Suf}_p(D', f, c, A)\right| = \left| \frac{cnt_{A=t[A]}(D_c)^2}{cnt_{A=t[A]}(D)} - \frac{cnt_{A=t[A]}(D_c)^2}{cnt_{A=t[A]}(D)+1} \right| \le 1$$

where in the equality we have used that $cnt_{A=a}(D_c) = cnt_{A=a}(D'_c)$ for all a since $D'_c = D_c$. We also used that the only summand in the definition of Suf_p that changes is the one corresponding to $t[A]$. The inequality uses item *(i)* of Lemma A.7.

If $D'_c = D_c \cup \{t\}$, we use item *(ii)* of Lemma A.7 to obtain

$$\left|\mathrm{Suf}_p(D, f, c, A) - \mathrm{Suf}_p(D', f, c, A)\right| = \left| \frac{cnt_{A=t[A]}(D_c)^2}{cnt_{A=t[A]}(D)} - \frac{(cnt_{A=t[A]}(D_c)+1)^2}{cnt_{A=t[A]}(D)+1} \right| \le 1.$$

□

A.3 Diversity

We begin with a sensitivity analysis of the diversity function from prior work, and proceed with the analysis of our low sensitivity diversity measure (Definition 4.9).

The diversity from [8] is defined as follows. Let $\mathcal{AC} : C \to \mathcal{A}$ be an attribute combination. For an attribute A, let Let $ExpBy(\mathcal{AC}, A) = \mathcal{AC}^{-1}(\{A\})$. That is, the set of cluster labels assigned to A by $\mathcal{AC}$ (which can also by empty).

For a finite set S, let $Perm(S)$ denote its set of permutations, containing all bijections $p : \{1, \ldots, |S|\} \to S$. For a permutation $p \in Perm(ExpBy(\mathcal{AC}, A))$, its diversity is defined as [8]:

$$PermDiv_A(p) = \sum_{i=1}^{|ExpBy(\mathcal{AC},A)|} \min_{j<i} \mathrm{TVD}(\pi_A(D_{p(i)}, \pi_A(D_{p(j)}))$$

and if $|ExpBy(\mathcal{AC}, A)| = 1$, $PermDiv_A(p)$ is set as 1.

Now, the diversity measure is defined as[6]

$$\mathrm{Div}(D, f, \mathcal{AC}) = \sum_{A \in \mathcal{A}} \sum_{p \in Perm(ExpBy(\mathcal{AC},A))} \frac{PermDiv_A(p)}{|ExpBy(\mathcal{AC}, A)|!}$$

Proposition A.8. *The sensitivity of* $\mathrm{Div}(D, f, \mathcal{AC})$ *is at least* $\frac{1}{2}$ *and its range is* $[0, |C|]$.

Note that since the number of clusters $|C|$ is typically a small constant, the sensitivity is relatively high compared to the range.

Proof. Let D be a dataset of size n and $f : \mathrm{dom}(R) \to C$ be a clustering function such that $D_1 = \{t_1\}$ is a cluster of size 1. Let A be an attribute and suppose that there exists $a \in \mathrm{dom}(A)$ such that $t[A] = a$ for all tuples $t \in D$. Let $\mathcal{AC}$ be an attribute combination mapping all cluster labels to A.

In this case, we have $ExpBy(\mathcal{AC}, A) = C$. Moreover, for any $c, c' \in C$ we have that $\mathrm{TVD}(\pi_A(D_c), \pi_A(D_{c'})) = 0$, as the distributions of values of the column A is identical among all clusters. Therefore, for any permutation $p \in Perm(C)$, we find that $PermDiv_A(p) = 0$. Hence, $\mathrm{Div}(D, f, \mathcal{AC}) = 0$.

Now, suppose a new tuple is added to cluster 1. Denote $D' = D \cup \{t_2\}$ and $D'_1 = \{t_1, t_2\}$. Suppose further that $t_2[A] = a'$ for $a' \ne a$. Now, for every $c \ne 1$ we have

$$\mathrm{TVD}(\pi_A(D_1), \pi_A(D_c)) = \frac{1}{2} \tag{5}$$

and for every $c, c' \in C \setminus \{1\}$ we have

$$\mathrm{TVD}(\pi_A(D_c), \pi_A(D_{c'})) = 0 \tag{6}$$

[6] to obtain a value between 0 and 1, this function can be normalized by the number of clusters $|C|$.

as for other clusters the distribution of values is unchanged. Hence, for any permutation $p \in Perm(C)$, we find that

$$PermDiv_A(p) = \sum_{i=1}^{|C|} \min_{j<i} \text{TVD}(\pi_A(D_{p(i)}, \pi_A(D_{p(j)})) = \frac{1}{2},$$

where we used that exactly one of the summands has the form (5) and the rest have the form (6). Therefore, we conclude that $\text{Div}(D, f, \mathcal{AC}) = 1/2$.

For the range bound, observe that $\text{Div}(D, f, \mathcal{AC})$ is maximized when all pairwise distances equal 1. This can occur, for instance, when $\mathcal{AC}$ maps each cluster to a distinct attribute. Alternatively, when any two clusters assigned with the same attribute A satisfy $\text{TVD}(\pi_A(D_1), \pi_A(D_c)) = 1$. Assuming this is the case, we have

$$PermDiv_A(p) = \sum_{i=1}^{|ExpBy(\mathcal{AC},A)|} 1 = |ExpBy(\mathcal{AC}, A)|\,.$$

Thus,

$$\begin{aligned}
\text{Div}(D, f, \mathcal{AC}) &= \sum_{A\in\mathcal{A}} \sum_{p\in Perm(ExpBy(\mathcal{AC},A))} \frac{PermDiv_A(p)}{|ExpBy(\mathcal{AC}, A)|!} \\
&= \sum_{A\in\mathcal{A}} \sum_{p\in Perm(ExpBy(\mathcal{AC},A))} \frac{|ExpBy(\mathcal{AC}, A)|}{|ExpBy(\mathcal{AC}, A)|!} \\
&= \sum_{A\in\mathcal{A}} |ExpBy(\mathcal{AC}, A)| \\
&= |C|\,.
\end{aligned}$$

□

We now proceed with the sensitivity analysis of our diversity measure (Definition 4.9).

Lemma A.9. *For two clusters $D_c, D_{c'} \subseteq D$ and attributes $A_c, A_{c'}$, it holds that $d(D, f, c, c', A_c, A_{c'}) \in [0, \min\{|D_c|, |D_{c'}|\}]$*

Proof. If one of the two clusters is empty, or if $A_c \neq A_{c'}$, then the claim trivially holds. Hence, we assume that both are not empty and that $A_c = A_{c'} = A$. In this case

$$\begin{aligned}
&\frac{1}{2} \sum_{a\in\text{dom}(A)} \left| \frac{cnt_{A=a}(D_c)}{\max\{|D_c|, 1\}} - \frac{cnt_{A=a}(D_{c'})}{\max\{|D_{c'}|, 1\}} \right| \\
&\leq \frac{1}{2} \sum_{a\in\text{dom}(A)} \left| \frac{cnt_{A=a}(D_c)}{|D_c|} \right| + \frac{1}{2} \sum_{a\in\text{dom}(A)} \left| \frac{cnt_{A=a}(D_{c'})}{|D_{c'}|} \right| \\
&= \frac{1}{2} + \frac{1}{2} = 1
\end{aligned}$$

and therefore $d(D, f, c, c', A_c, A_{c'}) \leq \min\{|D_c|, |D_{c'}|\}$. □

Proposition A.10. *d has sensitivity* 1.

The proof is similar to that of Proposition 4.4, with the main difference being that we now consider the distance between histograms of disjoint subsets of the data, instead of one subset and the entire dataset.

Proof. Let $f : \text{dom}(R) \to C$ be a clustering function, and $c, c' \in C$ be two cluster labels. Fix attributes $A_c, A_{c'}$, and let $D \sim D'$ be neighboring datasets such that $D' = D \cup \{t\}$.

Let D_c and $D_{c'}$ (respectively, D'_c and $D'_{c'}$) denote the sets of tuples in D (respectively, D') that are mapped to c and c' by the function f.

First, observe that when $A_c \neq A_{c'}$, we have $d(D, f, c, c', A_c, A_{c'}) = \min\{|D_c|, |D_{c'}|\}$, which changes by at most 1 when we add or remove one tuple from D. Hence, we proceed with the assumption that $A_c = A_{c'} = A$. To simplify notation, we let $d(D, c, c', A) = d(D, f, c, c', A, A)$. If t has not been added to either cluster, i.e, $D_c = D'_c$ and $D_{c'} = D'_{c'}$, then clearly

$$\left| d(D, c, c', A) - d(D', c, c', A) \right| = 0.$$

Otherwise, consider the case that t has been added to one of the clusters. Without loss of generality, suppose that $D'_{c'} = D_{c'} \cup \{t\}$ and $D'_c = D_c$. If either of D_c or $D_{c'}$ is empty, then by Lemma A.9, we have $d(D_c, D_{c'}, A) = 0$, and $d(D_c, D'_{c'}, A) \leq 1$, and so

$$\left| d(D, c, c', A) - d(D', c, c', A) \right| \leq 1.$$

Hence, we proceed with the assumption that both are not empty. Denote $a = t[A]$, and let $h_a = h_A(t)$ be the $|\text{dom}(A)|$-dimensional histogram with 1 in its a'th entry and 0 elsewhere. First, consider the case that $|D_c| \leq |D_{c'}|$. Recalling that $|D'_{c'}| = |D_{c'}| + 1$, we have

$$\begin{aligned}
&\left| d(D, c, c', A) - d(D', c, c', A) \right| \\
&= \frac{1}{2} \left| \left\| h_A(D_c) - \frac{|D_c|}{|D_{c'}|} h_A(D_{c'}) \right\|_1 - \left\| h_A(D_c) - \frac{|D_c|}{|D_{c'}| + 1} h_A(D'_{c'}) \right\|_1 \right|
\end{aligned}$$

By the triangle inequality,

$$\leq \frac{1}{2} \left\| h_A(D_c) - \frac{|D_c|}{|D_{c'}|} h_A(D_{c'}) - h_A(D_c) + \frac{|D_c|}{|D_{c'}| + 1} h_A(D'_{c'}) \right\|$$

Substituting $h_A(D'_{c'}) = h_A(D_{c'}) + h_a$ into the inequality above, we obtain

$$= \frac{1}{2} \left\| \frac{|D_c|}{|D_{c'}| + 1} h_a - \left(\frac{|D_c|}{|D_{c'}|} - \frac{|D_c|}{|D_{c'}| + 1} \right) h_A(D_{c'}) \right\|_1$$

Applying the triangle inequality and noting that $\frac{|D_c|}{|D_{c'}|} - \frac{|D_c|}{|D_{c'}|+1} = \frac{|D_c|}{|D_{c'}|(|D_{c'}|+1)}$, we have

$$\leq \frac{1}{2} \left(\|h_a\|_1 + \frac{|D_c|}{|D_{c'}|(|D_{c'}| + 1)} \|h(D_{c'})\|_1 \right)$$

substituting $\|h_a\|_1 = 1$ and $\|h(D_{c'})\|_1 = |D_{c'}|$,

$$= \frac{1}{2} \left(1 + \frac{|D_c| \cdot |D_{c'}|}{|D_{c'}|(|D_{c'}| + 1)} \right) \leq 1$$

where we have also used our assumption that $|D_c| \leq |D_{c'}|$.

Now, consider the case that $|D_c| > |D_{c'}|$, and therefore also $|D_c| \geq |D'_{c'}| = |D_{c'}| + 1$. We have

$$\begin{aligned}
&\left| d(D, c, c', A) - d(D', c, c', A) \right| \\
&= \frac{1}{2} \left| \left\| \frac{|D_{c'}|}{|D_c|} h_A(D_c) - h_A(D_{c'}) \right\|_1 - \left\| \frac{|D_{c'}| + 1}{|D_c|} h_A(D_c) - h_A(D'_{c'}) \right\|_1 \right|
\end{aligned}$$

By the triangle inequality,

$$\leq \frac{1}{2} \left\| \frac{|D_{c'}|}{|D_c|} h_A(D_c) - h_A(D_{c'}) - \frac{|D_{c'}| + 1}{|D_c|} h_A(D_c) + h_A(D'_{c'}) \right\|_1$$

Substituting $h_A(D'_{c'}) = h_A(D_{c'}) + h_a$ into the inequality above and rearranging, we obtain

$$\leq \frac{1}{2}\left\| \frac{-1}{|D_c|} h_A(D_c) + h_a \right\|_1$$

Next, we once again use $\|h_a\|_1 = 1$ and $\|h(D_c)\|_1 = |D_c|$

$$\leq \frac{1}{2}\left(\frac{1}{|D_c|}\|h_A(D_c)\|_1 + \|h_a\|_1 \right)$$
$$= 1$$

□

Proposition 4.10. *The sensitivity of* Div_p *is bounded by* 1. *Moreover, its range is* $[0, R_{\mathrm{Div}}]$ *where* $R_{\mathrm{Div}} = \frac{1}{\binom{|C|}{2}} \sum_{i=1}^{|C|} (|C| - i)\left|D_{c_i}\right|$ *is a weighted average of the cluster sizes, and* $|D_{c_i}| \leq |D_{c_{i+1}}|$.

Proof. The sensitivity bound follows from Proposition A.10 using Lemma A.3, as Div_p is a convex combination of sensitivity-1 functions. For the range bound, note that $d(D, f, c, c', A_c, A_{c'})$ attains its upper-bound (Lemma A.9), in particular, when $A_c \neq A_{c'}$. Hence, Let $\mathcal{AC}$ be an attribute combination assigning a different attribute for each cluster. It follows that Div_p is maximized, attaining value

$$\frac{1}{\binom{|C|}{2}} \sum_{\{c,c'\} \subseteq C} \min\{|D_c|, |D_{c'}|\}. \tag{7}$$

Now, let $|D_{c_1}| \leq \cdots \leq |D_{c_{|C|}}|$ be an ordering of the clusters by increasing size. For each $i = 1, \ldots, |C|$, the value $|D_{c_i}|$ appears as a summand in (7) $|C| - i$ times, each for a coupling of c_i with c_j for $j < i$. By changing the order of summation, we obtain that (7) equals

$$\frac{1}{\binom{|C|}{2}} \sum_{i=1}^{|C|} (|C| - i)\left|D_{c_i}\right|$$

□

A.4 Combining All Quality Functions

Proposition 4.12. $\mathrm{Score}_\gamma(D, f, c, A)$ *has sensitivity bounded by* 1 *and its range is* $[0, |D_c|]$.

Proof. The proposition follows directly from Lemma A.3 and the sensitivity bounds of the sufficiency and interestingness functions (Proposition 4.4, Proposition 4.7), as Score_γ is a convex combination of the two. □

Proposition 4.14. $\mathrm{GlScore}_\lambda$ *has sensitivity bounded by* 1. *Moreover, its range is* $[0, R_{\mathrm{GlScore}_\lambda}]$ *where*

$$R_{\mathrm{GlScore}_\lambda} = (\lambda_{\mathrm{Int}} + \lambda_{\mathrm{Suf}}) \cdot \frac{1}{|C|} \sum_{c \in C} |D_c| + \lambda_{\mathrm{Div}} \cdot R_{\mathrm{Div}}$$

is a weighted average of the cluster sizes, and R_{Div} *is defined as in Proposition 4.10.*

Proof. The proposition follows directly from Lemma A.3 and the sensitivity bounds of the sufficiency, interestingness and diversity functions (Proposition 4.4, Proposition 4.7, Proposition 4.10), as $\mathrm{GlScore}_\lambda$ is a convex combination of the three. For the range bounds, note that $\frac{1}{|C|}\sum_{c \in C} \mathrm{Suf}_p(D, f, c, A)$ and $\frac{1}{|C|}\sum_{c \in C} \mathrm{Suf}_p(D, f, c, A)$ both have range $[0, \frac{1}{|C|}\sum_{c \in C} |D_c|]$. The range upper bound for Div_p, R_{Div_p}, is given by *Proposition* 4.10. □

A.5 Proof of Proposition 5.1

Proposition 5.1. *Given a clustering function* f*, a set of attributes* $\mathcal{A}$*, a privacy parameter* $\varepsilon_{CandSet}$*, non-negative hyperparameters* $\gamma_{\mathrm{Int}}, \gamma_{\mathrm{Suf}}$ *that sum to* 1*, and a size parameter* k*, the following holds:*

(1) Algorithm 1 satisfies $\varepsilon_{CandSet}$*-DP.*

(2) For $c \in C$*, denote by* $\mathrm{OPT}_c^{(\ell)}$ *the* ℓ*-th highest (true) score, and by* $A_c^{(\ell)}$ *the* ℓ*-th explanation attribute selected by Algorithm 1 to* S_c*. For all* c *and* $\ell = 1, 2, \ldots, k$*, we have*

$$\mathbf{Pr}\left[\mathrm{Score}(c, A_c^{(\ell)}) \leq \mathrm{OPT}_c^{(\ell)} - \frac{2|C| \cdot k}{\varepsilon_{CandSet}}(\ln|\mathcal{A}| + t)\right] \leq e^{-t}.$$

where we denote $\mathrm{Score}(c, A_c^{(\ell)}) = \mathrm{Score}(D, f, c, A_c^{(\ell)})$

Proof. **(1) Privacy.** For each cluster, the distribution of the selected top-k is equivalent to iteratively applying k exponential mechanisms [15], where each satisfies $\varepsilon = \varepsilon_{CandSet}/(|C| \cdot k)$. Overall, we have $|C|$ applications of Top-k, hence the output distribution is equivalent to $|C| \cdot k$ exponential mechanisms. by sequential composition (Proposition 2.7), Algorithm 1 satisfies overall $\varepsilon_{CandSet}$-DP.

(2) Utility Bound. Let $c \in C$, and $\mathcal{A}_\ell$ be the set of remaining attributes after the $(\ell - 1)$-th selection for this cluster. The distribution over the selected sequence $A_c^{(1)}, \ldots, A_c^{(k)}$ is equal to applying k exponential mechanisms [15] where $A_c^{(\ell)}$ is selected from $\mathcal{A}_\ell$. Note that we always have $\max_{A \in \mathcal{A}_\ell} \mathrm{Score}(c, A) \geq \mathrm{OPT}_c^{(\ell)}$. Hence, for $\varepsilon = \varepsilon_{CandSet}/(|C| \cdot k)$, we apply the utility theorem of exponential mechanism (Theorem 3.11 in [19]) to obtain

$$\mathbf{Pr}\left[\mathrm{Score}(c, A_c^{(\ell)}) \leq \mathrm{OPT}_c^{(\ell)} - \frac{2}{\varepsilon}(\ln|\mathcal{A}| + t)\right]$$
$$\leq \mathbf{Pr}\left[\mathrm{Score}(c, A_c^{(\ell)}) \leq \max_{A \in \mathcal{A}_\ell} \mathrm{Score}(c, A) - \frac{2}{\varepsilon}(\ln|\mathcal{A}_\ell| + t)\right] \leq e^{-t}$$

where we have used that $\Delta_{\mathrm{Score}} \leq 1$. □

B Generating Multiple Explanations per Cluster

In this section we show how DPClustX can be extended to output multiple histograms per cluster in a global explanation. To this end, we extend the definition of an *attribute combination* to a mapping $\mathcal{AC} : C \to \{S \subseteq \mathcal{A} \mid |S| = \ell\}$, assigning to each cluster label a set of ℓ attributes. Our goal is thus to find a high-quality attribute combination such that the histograms of the corresponding attributes form a high-quality HBE, where the global explanation contains ℓ histograms per cluster. This requires extending the global score function Definition 4.13 to attribute combinations with larger outputs.

We measure the overall diversity as follows. Let

$$Cand(\mathcal{AC}) = \{(c, A) \mid c \in C, A \in \mathcal{AC}(c)\}$$

and define

$$\mathrm{Div}_\ell(D, f, \mathcal{AC}) = \frac{1}{\binom{|Cand(\mathcal{AC})|}{2}} \sum_{(c,A),(c',A')} d(D, f, c, c', A_c, A_{c'})$$

where the sum is over all different pairs $\{(c, A), (c', A')\} \subseteq Cand(\mathcal{AC})$. Note that the definition coincides with Definition 4.9 when $\ell = 1$.

The sufficiency and interestingness in the global score function Definition 4.13 were averages across all candidates. With the new definition, they remain averages, but are now taken over a larger set of candidates. Overall, we have:

$$\text{GlScore}_\lambda(D, f, \mathcal{AC}) = \lambda_{\text{Int}} \cdot \text{Int}_\ell(D, f, \mathcal{AC}) + \lambda_{\text{Suf}} \cdot \text{Suf}_\ell(D, f, \mathcal{AC}) + \lambda_{\text{Div}} \cdot \text{Div}_\ell(D, f, \mathcal{AC})$$

where we extend

$$\text{Int}_\ell(D, f, \mathcal{AC}) = \frac{1}{|Cand(\mathcal{AC})|} \sum_{(c,A) \in Cand(\mathcal{AC})} \text{Int}_p(D, f, c, A))$$

and

$$\text{Suf}_\ell(D, f, \mathcal{AC}) = \frac{1}{|Cand(\mathcal{AC})|} \sum_{(c,A) \in Cand(\mathcal{AC})} \text{Suf}_p(D, f, c, A).$$

Note that the definition coincides with Definition 4.13 when $\ell = 1$. The sensitivity of this function is bounded by 1, as it remains a convex combination of sensitivity-1 functions, with an analysis analogous to that of Proposition 4.14.

Stage-1 of DPClustX is unchanged, as its purpose is to narrow the search space to high-quality candidates for each cluster, which then form the pool for a global explanation. For Stage-2, we may use the exponential mechanism to select a high-scoring attribute combination $\mathcal{AC} : C \to \{S \subseteq \mathcal{A} \mid |S| = \ell\}$ with respect to the extended low sensitivity global score. Finally, As in the case of $\ell = 1$, noisy histograms are generated only for the $|C| \times \ell$ selected attributes.

However, the drawback of this approach is that the set of all possible attribute combinations, after the filtering performed in Stage-1, now has a size of $\binom{k}{\ell}^{|C|}$. This size can be large, and computing the global score for all attribute combinations may require significant computational time. While the utility guarantee of Algorithm 1 remains the same, the candidate set provided to the exponential mechanism in Algorithm 2 now has a size of $\binom{k}{\ell}^{|C|}$ instead of $k^{|C|}$, resulting in a larger additive error term of the EM utility bound. Indeed, letting OPT be the highest score of an attribute combination $\mathcal{AC} : C \to \{S \subseteq \mathcal{A} \mid |S| = \ell\}$ and $\mathcal{AC}$ be the one selected by the EM in Algorithm 2, Theorem 3.11 in [19] implies that

$$\mathbf{Pr}\left[\text{GlScore}_\lambda(\mathcal{AC}) \leq \text{OPT} - \frac{2\,|C|}{\varepsilon}\left(\ln\binom{k}{\ell} + t\right)\right] \leq e^{-t}$$

C Supplementary Experiments

Pre-processing of the Diabetes dataset. To ensure the interpretability of the generated histograms with a bounded number of bins, we apply several preprocessing steps to the Diabetes dataset [7]. We remove the unique identifiers ‘patient_nbr‘ and ‘encounter_id‘. Numerical attributes including ‘num_lab_procedures‘ and ‘num_medications‘ are binned. The attribute ‘medical_specialty‘ is mapped to broader categories such as “General Practice" and “Surgery", following the categorization in [63] that introduced the dataset. Additionally, each ICD code in the attributes ‘diag_1‘, ‘diag_2‘, and ‘diag_3‘ is replaced with its corresponding diagnostic category (e.g., values in the range 390–459 are mapped to "Circulatory") according to the mapping defined in [63]. The preprocessing code is publicly available in [1].

Pre-processing of the Stack Overflow dataset. We chose the 2018 Survey due to its larger sample size compared to more recent years. Both numerical and categorical attributes were considered, while attributes containing textual values or multiple-choice combinations were excluded. One potential approach was to expand the multiple-choice answers into binary attributes. However, this led to a significant increase in dimensionality, causing the DP-k-means algorithm to fail in clustering the data within a reasonable privacy budget, and other methods to fail due to scalability limitations. Additionally, we discard attributes with more than 60% of missing values. The numerical attribute ‘ConvertedSalary‘ is binned. The preprocessing code is available in [1].

C.1 Selected Attributes Quality score and Error

We present additional results for the Diabetes dataset with 3 and 7 clusters. Figure 11 shows the trend of *Quality* values of the selected attribute combination as the total privacy budget ε varies. Figure 12 shows the trend of MAE values of the selected attribute combination as the total privacy budget ε varies.

C.2 Quality for different choices of weights

We present the missing empirical results on the Diabetes and the Census datasets in Table 1.

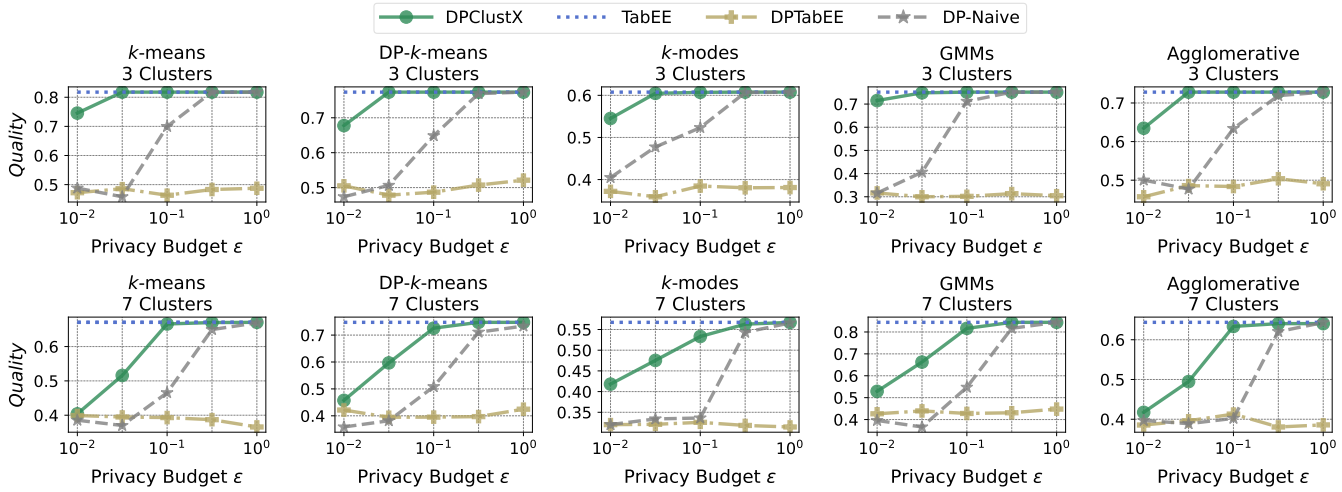


Figure 11: *Quality* **values of the selected attribute combination for the Diabetes dataset, as the total privacy budget** ε **varies.**

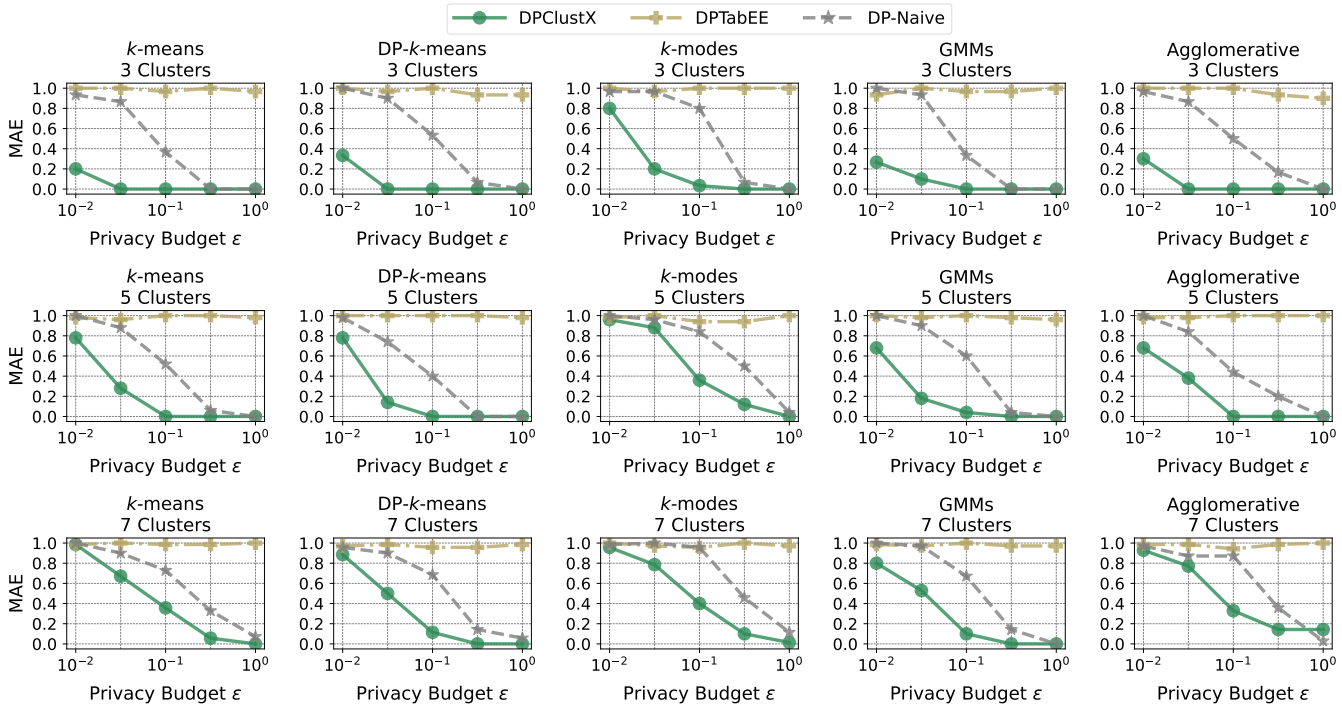


Figure 12: MAE **values of the selected attribute combination for the Diabetes dataset, as the total privacy budget** ε **varies.**

# Clusters	Clustering	Explainer	Equal	$\lambda_{\text{Int}} = 0$	$\lambda_{\text{Suf}} = 0$	$\lambda_{\text{Div}} = 0$
3	k-means	DPClustX	0.8176	0.9208	0.7882	0.7439
		TabEE	0.8176	0.9208	0.7882	0.7439
	k-modes	DPClustX	0.6076	0.7493	0.6618	0.4114
		TabEE	0.6076	0.7493	0.6621	0.4114
	Agglomerative	DPClustX	0.7281	0.8291	0.7362	0.6547
		TabEE	0.7281	0.8291	0.7362	0.6547
	DP-k-means	DPClustX	0.7735	0.8701	0.7494	0.7010
		TabEE	0.7735	0.8701	0.7662	0.7010
	GMMs	DPClustX	0.7515	0.8709	0.7563	0.6294
		TabEE	0.7515	0.8709	0.7563	0.6294
5	k-means	DPClustX	0.6874	0.7498	0.7805	0.5319
		TabEE	0.6874	0.7498	0.7805	0.5319
	k-modes	DPClustX	0.5592	0.6615	0.6821	0.3500
		TabEE	0.5639	0.6641	0.6828	0.3503
	Agglomerative	DPClustX	0.7255	0.7804	0.7899	0.6061
		TabEE	0.7255	0.7836	0.7899	0.6061
	DP-k-means	DPClustX	0.7735	0.8263	0.8310	0.6633
		TabEE	0.7735	0.8263	0.8310	0.6633
	GMMs	DPClustX	0.8164	0.8708	0.8518	0.7267
		TabEE	0.8164	0.8708	0.8518	0.7267
7	k-means	DPClustX	0.6664	0.7040	0.7893	0.5304
		TabEE	0.6706	0.7063	0.7893	0.5307
	k-modes	DPClustX	0.5613	0.6380	0.6994	0.3527
		TabEE	0.5673	0.6461	0.7049	0.3533
	Agglomerative	DPClustX	0.6396	0.6957	0.7481	0.5068
		TabEE	0.6438	0.6926	0.7630	0.5068
	DP-k-means	DPClustX	0.7442	0.7696	0.8340	0.6429
		TabEE	0.7474	0.7798	0.8340	0.6429
	GMMs	DPClustX	0.8440	0.8694	0.8967	0.7660
		TabEE	0.8440	0.8694	0.8967	0.7660

(a) Diabetes dataset.

# Clusters	Clustering	Explainer	Equal	$\lambda_{\text{Int}} = 0$	$\lambda_{\text{Suf}} = 0$	$\lambda_{\text{Div}} = 0$
3	k-means	DPClustX	0.8785	0.9888	0.8289	0.8187
		TabEE	0.8785	0.9888	0.8289	0.8187
	k-modes	DPClustX	0.8749	0.9859	0.8265	0.8131
		TabEE	0.8749	0.9859	0.8265	0.8131
	DP-k-means	DPClustX	0.8889	1.0000	0.8333	0.8333
		TabEE	0.8889	1.0000	0.8333	0.8333
	GMMs	DPClustX	0.5438	0.7122	0.6066	0.3157
		TabEE	0.5438	0.7122	0.6066	0.3157
5	k-means	DPClustX	0.8637	0.9195	0.8768	0.7987
		TabEE	0.8643	0.9197	0.8768	0.7987
	k-modes	DPClustX	0.8247	0.8981	0.8390	0.7449
		TabEE	0.8248	0.8981	0.8390	0.7449
	DP-k-means	DPClustX	0.8827	0.9552	0.8541	0.8451
		TabEE	0.8828	0.9658	0.8619	0.8451
	GMMs	DPClustX	0.4822	0.6296	0.6009	0.2258
		TabEE	0.4820	0.6296	0.6009	0.2258
7	k-means	DPClustX	0.8521	0.8825	0.8995	0.7922
		TabEE	0.8538	0.8865	0.9016	0.7922
	k-modes	DPClustX	0.7798	0.8506	0.8293	0.6719
		TabEE	0.7806	0.8513	0.8293	0.6726
	DP-k-means	DPClustX	0.8981	0.9376	0.9053	0.8515
		TabEE	0.8985	0.9376	0.9064	0.8515
	GMMs	DPClustX	0.4944	0.6174	0.6271	0.2430
		TabEE	0.4944	0.6174	0.6271	0.2430

(b) Census dataset.

Table 1: *Quality* values for with different choices of parameters λ_{Int}, λ_{Suf}, and λ_{Div}.