TOWARDS DIFFERENTIALLY PRIVATE QUERY RELEASE FOR HIERARCHICAL DATA

Terrance Liu
Carnegie Mellon University
Pittsburgh, PA 15213, USA
terrancl@cs.cmu.edu

Zhiwei Steven Wu
Carnegie Mellon University
Pittsburgh, PA 15213, USA
zstevenwu@cmu.edu

ABSTRACT

While differentially private query release has been well-studied, research in this area is commonly restricted to data that do not exhibit hierarchical structure. However, in many real-world scenarios, individual data points can be grouped together (e.g., people within households, taxi trips per driver, etc.), begging the question—what statistical properties (or queries) are important when considering data of this form? In addition, although synthetic data generation approaches for private query release have grown increasingly popular, it is unclear how one can generate synthetic data at both the *group* and *individual*-level while capturing such statistical properties. In light of these challenges, we formalize the problem of *hierarchical query release* and provide a set of statistical queries that capture relationships between attributes at both the *group* and *individual*-level. Furthermore, we propose and implement a novel synthetic data generation algorithm, H-GEM, which outputs hierarchical data subject to differential privacy to answer such statistical queries. Finally, using the American Community Survey, we evaluate H-GEM, establishing a benchmark for future work to measure against.

1 INTRODUCTION

Differential privacy (Dwork et al., 2006) provides rigorous guarantees for privacy protection that centers around limiting the influence of any individual data point when utilizing sensitive information. As a result, organizations have increasingly adopted differential privacy to release information that is beneficial to the public while protecting the privacy of individuals. The 2020 U.S. Decennial Census, for example, serves as one of the most prominent deployments of differential privacy in recent years (Abowd, 2018).

In this work, we study *differentially private query release*, where the goal is to release a set of summary statistics while preserving privacy guarantees. Query release is one of most fundamental problems in differential privacy and remains a key objective for many organizations, including the U.S. Census Bureau. While various types of methods have been proposed to tackle this problem, one approach that has gained traction in recent years is to generate synthetic data that preserves statistical properties (query answers) of the private dataset (Hardt et al., 2012; Gaboardi et al., 2014; McKenna et al., 2019; Vietri et al., 2020; Liu et al., 2021a; Aydore et al., 2021; Liu et al., 2021b). In fact, after announcing plans to incorporate differential privacy into the American Community Survey (ACS) release (Jarmin, 2019), the U.S. Census Bureau also declared, albeit informally, that it intended to replace the American Community Survey with fully synthetic data in the future (Rodríguez, 2021).

Studies using the ACS (as well as other hierarchical datasets) often study interrelationships between individuals across groups, such as trends observed between domestic partners in a household. However, past works on private query release—including those that have used the ACS itself as a testbed for their proposed algorithms (Liu et al., 2021a;b)—have ignored hierarchical data settings. In addition, social scientists have criticized the use of synthetic data generation approaches in general (even without differential privacy), arguing that they can only capture statistical relationships at the individual-level and are therefore unsuitable for ACS microdata (IPUMS USA). Attempting to preserve differential privacy guarantees only further compounds this problem.

Consequently, the objective of this paper is to initiate the study of differentially private query release for hierarchical data, and in pursuit of this goal, we make the following contributions: (1) We formulate the problem of *hierarchical query release* with two levels in the data hierarchy (*group/individual*). (2) We present a general set of queries that capture relationships between variables at different levels. In particular, we formulate queries in such a way that MWEM (Hardt et al., 2012), a synthetic data generation algorithm designed for query release in the *non-hierachical* setting, can be extended. (3) Finally, we introduce our algorithm, H-GEM, which adapts a neural network architecture to outperform MWEM while scaling to significantly larger data domains.

2 PRELIMINARIES

We consider the problem of answering a collection of queries Q about some private dataset D. To formalize this problem, we first let $\mathcal{X} = \{0,1\}^d$ denote some data domain of size d. As a result, any dataset can be represented as some histogram $D \in \mathbb{N}^d$. In the context of synthetic data generation for private query release, our goal then is to output some synthetic dataset $\hat{D}$ such that the error over all queries ($\max_{q \in Q} |q(\hat{D}) - q(D)|$) is small.

We consider synthetic data generation algorithms that satisfy differential privacy (Dwork et al., 2006), meaning that they employ randomized mechanisms $\mathcal{M} : \mathcal{X}^* \to \mathbb{R}$ that are privacy-preserving when accessing the private dataset.

Definition 1 (Differential privacy (Dwork et al., 2006)). *A randomized mechanism $\mathcal{M} : \mathcal{X}^n \to \mathbb{R}$ is (ϵ, δ)-differentially privacy, if for all neighboring datasets D, D' (i.e., differing on a single person), and all measurable subsets $S \subseteq \mathbb{R}$ we have:*

$$P\left(\mathcal{M}(D) \in S\right) \leq e^{\varepsilon} P\left(\mathcal{M}(D') \in S\right) + \delta$$

Next, we introduce our definition of *statistical linear query* as the following:

Definition 2 (statistical linear query). *Given a dataset D and predicate function $\phi : \mathcal{X} \to \mathbb{R}$, a statistical linear query is a defined as*

$$q_{\phi}^{linear}(D) = \sum_{x \in D} \phi(x)$$

For example, a statistical query counting the number of males in some dataset can be represented as a statistical linear query that uses the indicator function $\phi(x) = \mathbb{1}\{x \text{ is male}\}$. Similar to previous work, we will normalize query answers with respect to some statistic, such as the total number of rows in the dataset (e.g. *% males = # males / # people*). We formally define such queries as:

Definition 3 (normalized query). *Given a dataset D and predicate functions $\phi_1, \phi_2 : \mathcal{X} \to \mathbb{R}$, a normalized query is a defined as*

$$q_{\phi_1,\phi_2}^{norm}(D) = \frac{q_{\phi_1}^{linear}(D)}{q_{\phi_2}^{linear}(D)}$$

where for all $x \in \mathcal{X}$, $\phi_1(x) \leq \phi_2(x)$.

Normalized queries can define conditional statistics, such as—*What fraction of males in my dataset are white?* (i.e., $\phi_1(x) = \mathbb{1}\{x \text{ is male and white}\}$ and $\phi_2(x) = \mathbb{1}\{x \text{ is male}\}$). Similarly, statistical queries that output counts as fraction of rows in the dataset (rather than the total) can be reduced to the simple case in which $\phi_1(x) \in \{0,1\}$ and $\phi_2(x) = 1$ for all $x \in \mathcal{X}$ (i.e., $q_{\phi_2}^{linear}(D) = |D|$ counts the total number of rows in D).

Finally, following previous work on private query release, we will focus our attention to tabular data with columns that are either categorical or discretized, using k-way marginals for our predicate ϕ.

Definition 4 (k-way marginal). *Given a subset $S \subseteq [d]$ of k attributes and a target value $y \in \prod_{i \in S} \mathcal{X}_i$ for each feature in S, a k-way marginal query is given by:*

$$\phi_{S,y}(x) = \prod_{i \in S} \mathbb{1}\{x_i = y_i\}$$

where $x_i \in \mathcal{X}_i$ means the i-th attribute of record $x \in \mathcal{X}$.

In other words, k-way marginals define some logical conjunction over k attributes in $\mathcal{X}$. We define each subset of attributes S as a *marginal*, where each marginal is comprised of $\prod_{i=1}^{k} |\mathcal{X}_i|$ queries.

Table 1: Given some predicate function ϕ, we can distinguish between different hierarchical queries by reducing the query type to (1) whether counts are made at the *group*- or *individual*-level and (2) the domain of the predicate function ϕ (i.e. *group*- or *individual*-attributes). We then describe the corresponding condition for each combination of (1) and (2) and give an example using 1-way marginals for ϕ.

	DOMAIN(ϕ)	Conditions	Example
group	$\mathcal{G}$	satisfies predicate ϕ	What proportion of *households* reside in **rural** areas?
	$\mathcal{I}$	contains individual row that satisfies predicate ϕ	What proportion of *households* contain **at least one male** individual?
individual	$\mathcal{I}$	satisfies a predicate ϕ	What proportion of *individuals* are **male**?
	$\mathcal{G}$	belongs to a group that that satisfies predicate ϕ	What proportion of *individuals* live in **rural** households?

3 HIERARCHICAL COUNTING QUERIES

We consider data with some hierarchical structure, in which the data can first be partitioned into different groups that can then further be divided into individual rows. For example, one can form a hierarchical dataset from census data by grouping individuals into their respective households. We assume that each group contains at most M rows and has k_G features $\mathcal{G} = (\mathcal{G}_1 \times \ldots \times \mathcal{G}_{k_G})$. Similarly, we assume that each individual row belongs to some data domain of k_I features $\mathcal{I} = (\mathcal{I}_1 \times \ldots \times \mathcal{I}_{k_I})$. Together, we then have a hierarchical data universe $\mathcal{X} = \mathcal{G} \times (\mathcal{I} \times \perp)^M$, where $\perp$ represents an empty set of features in $\mathcal{I}$ (i.e., a nonexistent individual). Letting the *group-level* domain size $d_G = \prod_{i=1}^{k_G} |\mathcal{G}_i|$ and *individual-level* domain size $d_I = \prod_{i=1}^{k_I} |\mathcal{I}_i|$, the overall domain size can be written as $d = d_G \left(\sum_{i=1}^{M} (d_I)^i \right)$. Finally, we let N_G be the number of groups in D and N_I be the number of individual rows.

Having grouped the attributes of our hierarchical domain into *group* and *individual*-level attributes $\mathcal{G}$ and $\mathcal{I}$, we now introduce two classes of queries—(1) counting queries at the *group*-level $\mathcal{Q}_G$ (i.e. proportion of households) and (2) counting queries at the *individual*-level $\mathcal{Q}_I$ (e.g. proportion of individuals). In other words, all queries in this work follow the form—What proportion of [groups/individuals] satisfy condition C?—where C is some boolean condition that depends on (1) some predicate function ϕ (and its corresponding domain) and (2) the query class (*group/individual*) they belong to. Consequently, for any dataset $D \in \mathcal{X}$ with N_I individual rows comprising N_G groups, the ℓ_1-sensitivities of queries in $\mathcal{Q}_G$ and $\mathcal{Q}_I$ are $\frac{1}{N_G}$ and $\frac{1}{N_I}$ respectively.

We summarize in Table 1 the different query conditions for both *group* and *individual*-level queries that we consider. In particular, we use the 1-way marginal as our predicate function to define the conditions found in Table 1. Given some set S_G of *group*-level attributes and set S_I of *individual*-level attributes where $k = |S_G| + |S_I|$, we construct a set of *group* and *individual*-level queries by combining conditions C_i for each attribute $\mathcal{X}_i \in S_G \cup S_I$ (i.e., each query condition is the conjunction of k conditions found in Table 1). For example, given the *group*-level attribute URBAN/RURAL and *individual*-level attributes SEX and RACE, we have queries of the following form:

1. ($\mathcal{Q}_G$) What proportion of *households* are located in an **urban** area and contain *at least one individual* who is **female** and **white**?
2. ($\mathcal{Q}_I$) What proportion of *individuals* are **female**, **white**, and reside in a household located in an **urban** area?

4 MODELING

We first describe how to represent queries in $\mathcal{Q}_G$ and $\mathcal{Q}_I$ w.r.t. a histogram representation $\mathcal{D}$ of $\mathcal{X}$. Using Definition 3 to represent queries in Q, we are then able to extend MWEM to hierarchical data in $\mathcal{X}$. To overcome the computational bottlenecks of MWEM, we then introduce our main method, H-GEM, which models the hierarchical structure in $\mathcal{X}$ as a mixture of various product distributions.

4.1 EXPLICIT HISTOGRAM REPRESENTATIONS

First, we consider algorithms that optimize over the distributional family $\mathcal{D}$, where $\mathcal{D} = \left\{\mathbf{x} \mid \mathbf{x} \in [0,1]^d, \|\mathbf{x}\|_1 = 1\right\}$ is the set of all normalized histograms over $\mathcal{X}$. Specifically, we demonstrate how to evaluate any hierarchical query on some histogram D, which in turn allows us to directly optimize our objective using the *multiplicative weights* update rule in MWEM. Given some dataset $D \in \mathcal{D}$, we can write both *group* and *individual*-level queries according to Definition 3 where $q_\phi^{linear}(D) = \langle \vec{q}_\phi, D \rangle$ and $\vec{q}_\phi = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_d)]$.

Group-level counting queries ($\mathcal{Q}_G$). Given that each histogram D is a joint distribution over all possible group types, we have that $\phi_1 : \mathcal{X} \to \{0, 1\}$. Moreover, since we are normalizing over the total number of groups, we simply have that $\phi_2(x) = 1 \implies q_{\phi_2}^{linear}(D) = 1$ for all $D \in \mathcal{D}$. Therefore, *group*-level counting queries $q(D) = \langle \vec{q}_{\phi_1}, D \rangle$ are linear in $\mathcal{D}$.

Individual-level counting queries ($\mathcal{Q}_I$). Each group can contribute up to M individuals when counting the total individuals satisfying some predicate function. Therefore, we must instead write such queries as normalized queries q_{ϕ_1,ϕ_2}^{norm} where $\phi_1, \phi_2 : \mathcal{X} \to \mathbb{N}_{\leq M}$ and $\phi_2(\mathbf{x})$ evaluates to the number of individual rows in the group (i.e., attribute $\mathcal{X}^{(c)}$). Then in this case, $q(D) = \langle \vec{q}_{\phi_1}, D \rangle / \langle \vec{q}_{\phi_2}, D \rangle$

4.2 PRODUCT DISTRIBUTIONS VIA GENERATIVE NEURAL NETWORKS

Next, we introduce our algorithm, Hierarchical GEM (or H-GEM), which models the joint distribution (over $\mathcal{X}$) of group types using a collection of product distribution mixtures parametrized by neural networks. We derive our algorithm from GEM (Liu et al., 2021b), which uses a single neural network to parametrizes distributions for non-hierarchical data. We describe our changes below:

Modeling. Given some noise $\mathbf{z} \sim \mathcal{N}(0, I)$, we use a multi-headed neural network F in H-GEM and let $F_c(\mathbf{z})$, $F_G(\mathbf{z})$, and $F_I(\mathbf{z})$ denote the output of each head. In our model, $F_c(\mathbf{z})$ defines a probability distribution over the possible number individual rows in a particular group. Meanwhile, $F_G(\mathbf{z})$ and $F_I(\mathbf{z})$ model the distributions of groups and of individual rows within in each group respectively. For our experiments, we use an MLP (with residual connections) for F, where F_G is the concatenation of M separate product distributions that we designate as $F_{I,1} \ldots F_{I,M}$. Thus, combining all three heads, we have that $F(\mathbf{z}_i)$ is some $(M + k_G + Mk_I)$-dimensional vector.

Sampling. To sample rows from $F(\mathbf{z}_i)$, we carry out a 2-stage sampling procedure. First, we sample from $F_G(\mathbf{z}_i)$ a set of attributes in $\mathcal{G}$ to define the group-level features and the number of individuals $m \sim F_c(\mathbf{z}_i)$ in each group. Subsequently, we sample m sets of individual-level attributes $\mathcal{I}$ according to the distribution $F_I(\mathbf{z}_i)$. Note that we fix which *individual*-level distribution $F_{I,j}$ corresponds to each group size. For notational purposes, we designate that the set of distributions $\{(F_G)_j \mid j = 1 \ldots m\}$ correspond to individual rows in a group of size m (i.e., the first row is sampled from $F_{I,1}(\mathbf{z}_i)$, the second from $F_{I,2}(\mathbf{z}_i)$, etc.).

Loss function. Following the Adaptive Measurements framework (Liu et al., 2021b), at each round t, H-GEM is given a set of selected queries $\tilde{Q}_{1:t} = \{\tilde{q}_1, \ldots, \tilde{q}_t\}$ and their noisy measurements $\tilde{M}_{1:t} = \{\tilde{m}_1, \ldots, \tilde{m}_t\}$. As in GEM, we have some batch size B such that $\mathbf{z} \sim \mathcal{N}(0, I_B)$. Then for each query q, there exists a corresponding function $f_q : \mathbb{R}^d \to \mathbb{R}$ such that the answer given by our neural network F for any query q is $f_q\left(F\left(z_j\right)\right)$. Using ℓ_2-norm, we have the loss function:

$$\mathcal{L}^{\text{H-GEM}}\left(\tilde{Q}_{1:t}, \tilde{M}_{1:t}\right) = \sum_{i=1}^{t} \left\|\tilde{m}_i - f_{\tilde{q}_i}\left(F\left(\mathbf{z}\right)\right)\right\|_2 \tag{1}$$

where

$$f_q\left(F\left(\mathbf{z}\right)\right) = \frac{1}{B}\sum_{k=1}^{B}\left(\prod_{i\in S_G}\left(F_G(\mathbf{z}_k)\right)_i\right)\left(\sum_{m=1}^{M}\left(F_c(\mathbf{z}_k)\right)_m\left(1-\prod_{j}^{m}\left(1-\prod_{i\in S_I}\left(F_{I,j}(\mathbf{z}_k)\right)_i\right)\right)\right)$$

for *group*-level hierarchical queries and

$$f_q\left(F\left(\mathbf{z}\right)\right) = \frac{f_{q,1}\left(F\left(\mathbf{z}\right)\right)}{f_{q,2}\left(F\left(\mathbf{z}\right)\right)}$$

$$f_{q,1}\left(F\left(\mathbf{z}\right)\right) = \sum_{k=1}^{B}\left(\prod_{i\in S_G}\left(F_G(\mathbf{z}_k)\right)_i\right)\left(\sum_{m=1}^{M}\left(F_c(\mathbf{z}_k)\right)_m\left(\sum_{j=1}^{m}\left(\prod_{i\in S_I}\left(F_{I,j}(\mathbf{z}_k)\right)_i\right)\right)\right)$$

$$f_{q,2}\left(F\left(\mathbf{z}\right)\right) = \sum_{k=1}^{B}\sum_{m=1}^{M} m\left(F_c(\mathbf{z}_k)\right)_m$$

for *individual*-level hierarchical queries.

5 Experiments

We evaluate our methods on the American Community Survey (ACS) (Ruggles et al., 2021), selecting data from 2019 for the state of New York. In addition to our main evaluation dataset (ACS NY-19), we take a low-dimensional extract (ACS-small NY-19) of the data so that we can evaluate the performance of MWEM[1] across privacy budgets $\varepsilon \in \{0.125, 0.25, 0.5, 1.0\}$. We provide more details of how we construct our datasets and queries in Appendix A.1. We show in Figure 1 that both MWEM and H-GEM are able to produce synthetic data to answer hierarchical queries, with H-GEM outperforming MWEM and scaling to higher-dimensional data domains.[2]

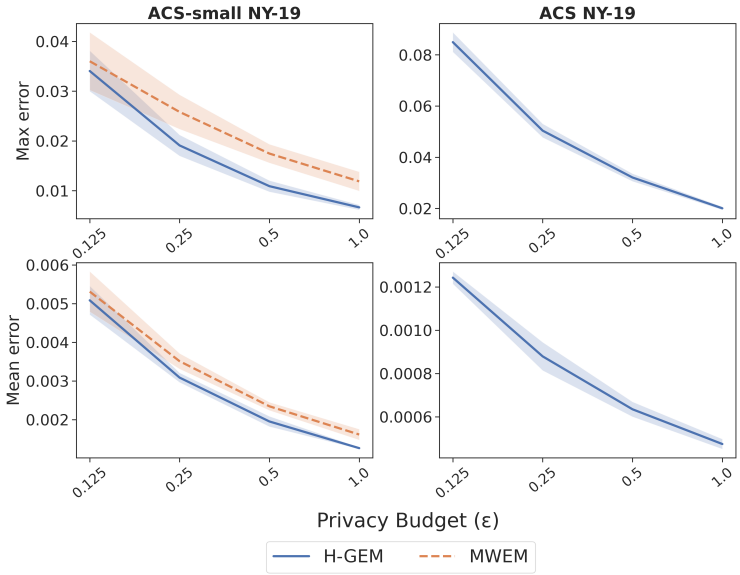

Figure 1: Max and mean errors for *group* and *individual*-level hierarchical queries evaluated on ACS/ACS-small NY-19 where $\varepsilon \in \{0.125, 0.25, 0.5, 1\}$ and $\delta = \frac{1}{N_G^2}$. The *x-axis* uses a logarithmic scale. Results are averaged over 5 runs, and error bars represent one standard error.

[1]To empirical performance, we incorporate modifications detailed in Liu et al. (2021a)—namely (1) using the *Gaussian mechanism* with *zCDP* composition (Bun & Steinke, 2016) and (2) recycling past measurements.

[2]It is computationally infeasible to run MWEM on ACS NY-19 since it requires maintaining a distribution over a domain of size $d \approx 7.3 \times 10^{71}$

REFERENCES

John M. Abowd. The U.S. census bureau adopts differential privacy. In *ACM International Conference on Knowledge Discovery & Data Mining*, pp. 2867, 2018.

Sergul Aydore, William Brown, Michael Kearns, Krishnaram Kenthapadi, Luca Melis, Aaron Roth, and Ankit Siva. Differentially private query release through adaptive projection. *arXiv preprint arXiv:2103.06641*, 2021.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Proceedings of the 14th Conference on Theory of Cryptography*, TCC '16-B, pp. 635–658, Berlin, Heidelberg, 2016. Springer.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pp. 265–284, Berlin, Heidelberg, 2006. Springer.

Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Zhiwei Steven Wu. Dual query: Practical private query release for high dimensional data. In *International Conference on Machine Learning*, pp. 1170–1178. PMLR, 2014.

Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems*, pp. 2339–2347, 2012.

IPUMS USA. Changes to census bureau data products, 2022. URL https://www.ipums.org/changes-to-census-bureau-data-products.

Ron Jarmin. Census bureau continues to boost data safeguards. *U.S. Census: Random Samplings Blog*, 2019. URL https://www.census.gov/newsroom/blogs/random-samplings/2019/07/boost-safeguards.html.

Terrance Liu, Giuseppe Vietri, Thomas Steinke, Jonathan Ullman, and Zhiwei Steven Wu. Leveraging public data for practical private query release. *arXiv preprint arXiv:2102.08598*, 2021a.

Terrance Liu, Giuseppe Vietri, and Zhiwei Steven Wu. Iterative methods for private synthetic data: Unifying framework and new methods. *arXiv preprint arXiv:2106.07153*, 2021b.

Ryan McKenna, Daniel Sheldon, and Gerome Miklau. Graphical-model based estimation and inference for differential privacy. In *International Conference on Machine Learning*, pp. 4435–4444. PMLR, 2019.

Rolando A. Rodríguez. Disclosure avoidance and the american community survey. 2021 ACS Data Users Conference, 2021. URL https://acsdatacommunity.prb.org/m/2021-acs-conference-files/147/download.

Steven Ruggles, Sarah Flood, Sophia Foster, Ronald Goeken, Jose Pacas, Megan Schouweiler, and Matthew Sobek. Ipums usa: Version 11.0 [dataset]. minneapolis, mn: Ipums, 2021. URL https://doi.org/10.18128/D010.V11.0.

Giuseppe Vietri, Grace Tian, Mark Bun, Thomas Steinke, and Steven Wu. New oracle-efficient algorithms for private synthetic data release. In *International Conference on Machine Learning*, pp. 9765–9774. PMLR, 2020.

A Appendix

A.1 Experimental Setup

A.1.1 Data

We select households of size $M = 5$ or smaller for ACS and $M = 3$ or smaller for ACS-SMALL. In addition, we binarize attributes in $\mathcal{I}$ for ACS-SMALL by combining categories in order to reduce the domain size further. We detail in Table 2 the attributes we select for each dataset.

ACS-SMALL has total domain size of 262080. We note that if we ignore hierarchical structure of this data domain (as in typical settings in the literature) and consider the datasets as only collections of individuals, the domain size would be significantly smaller at 960. On the other hand, using the same histogram representation for ACS gives us domain size of $d \approx 7.3 \times 10^{71}$, necessitating the more compact representation found in H-GEM.

Table 2: Data Attributes

Dataset	Domain	Attributes
ACS	$\mathcal{G}$	COUNTYFIP, METRO, FARM, OWNERSHP FARMPROD, ACREHOUS, ROOMS BUILTYR2, FOODSTMP, MULTGEN
	$\mathcal{I}$	SFRELATE, SEX, MARST, RACE HISPAN, CITIZEN, EDUC, SCHOOL EMPSTAT, LOOKING, AGE
ACS-Small	$\mathcal{G}$	METRO, OWNERSHP, FARM, FOODSTMP
	$\mathcal{I}$	SEX, AGE, EMPST, MARST

A.1.2 Queries

As shown in Section 3, given some set of *group* and *individual*-level attributes $S = S_G \cup S_I$, we construct a set *group* and *individual*-level queries $\mathcal{Q}_G$ and $\mathcal{Q}_I$. We restrict the number of attributes for each query to $|S| = 3$. Because of the size of ACS-SMALL, we simply choose all possible queries in this settings ($|Q| = 1640$ queries). For ACS, we randomly sample k sets of attributes for each possible number of *group*-level attributes (i.e., $|S_G| = 0 \ldots 3$). This in total gives us $4 \times k$ sets of attributes S to construct $\mathcal{Q}$. In our experiments, we select $k = 128$, which amounts to 434774 queries.

A.2 Additional Algorithm Details

We provide the exact details of our variant of MWEM and H-GEM in the following sections. Both algorithms fall under Adaptive Measurements where we assume all queries $q \in \mathcal{Q}$ have ℓ_1-sensitivity $\Delta_{\mathcal{Q}} = \max\left(\frac{1}{N_G}, \frac{1}{N_I}\right) = \frac{1}{N_G}$. Note that the ℓ_p-sensitivity of function (i.e., query) captures the effect of changing an individual in the dataset and is used for deriving the noise required to be added for preserving differential privacy.

Definition 5 (ℓ_p-sensitivity)**.** *The ℓ_p-sensitivity of a function $f : \mathcal{X}^* \to \mathbb{R}^k$ is*

$$\Delta f = \max_{\textit{neighboring } D, D'} \|f(D) - f(D')\|_p$$

Consequently, both algorithms satisfy ρ-zCDP and (ε, δ)-DP for $\varepsilon \leq \rho + 2\sqrt{\rho \log(1/\delta)}$ (Liu et al., 2021b, Theorem 1).

A.2.1 MWEM

We restate MWEM in Algorithm 1, with a slight change to the *multiplicative weights* update rule—in our case, we rescale $q_t(x)$ by a factor of $\frac{1}{M}$ (Algorithm 2) when q_t is an *individual*-level query so that $|q_t| \leq 1$. In addition, we add empirical improvements described in Liu et al. (2021a), which are presented in Algorithm 2.

Algorithm 1: MWEM

Input: Private hierarchical dataset $D \in \mathcal{X}$, query class $\mathcal{Q} = \mathcal{Q}_G \cup \mathcal{Q}_I$
Parameters: Privacy parameter $\rho > 0$, number of iterations T, max per-round iterations T_{max}
Let N_G be the number of groups in D
Let M be the maximum possible number of individual rows belonging to a group in $\mathcal{X}$
Let $\varepsilon_0 = \sqrt{\frac{2\rho}{T\left(\alpha^2 + (1-\alpha)^2\right)}}$ for $\alpha = \frac{1}{2}$
Initialize A_0 be a uniform distribution over $\mathcal{X}$
for $t = 1$ **to** T **do**
Sample: Select query $\tilde{q}_t \in \mathcal{Q}$ using the *exponential mechanism* with parameter ε_0 and score function

$$P[\tilde{q}_t = q] \propto \exp\left\{\frac{\alpha\varepsilon_0 N_G}{2}|q(D) - q(A_{i-1})\right\}$$

Measure: Take (via the *Gaussian mechanism*) measurement

$$m_t = \tilde{q}_t(D) + \mathcal{N}\left(0, \left(\frac{1}{N_G(1-\alpha)\varepsilon_0}\right)^2\right)$$

Update: A_t = MWEM-Update $(A_{t-1}, \tilde{Q}_t, \tilde{M}_t, T_{\text{max}})$ where $\tilde{Q}_t = \langle \tilde{q}_1, \ldots, \tilde{q}_t \rangle$ and $\tilde{M}_t = \langle \tilde{m}_1, \ldots, \tilde{m}_t \rangle$
end for
Output: $A = \text{avg}_{t \in [T]} A_{t-1}$

Algorithm 2: MWEM-Update

Input: Normalized histogram A, queries $Q = \langle q_1, \ldots, q_t \rangle$, noisy measurements $M = \langle m_1, \ldots, m_t \rangle$, max iterations T_{max}
Let $a_{\text{max}} = \max_{1 \leq t \leq |Q|} |m_t - q_t(A)|$ be the max error across queries in Q
Let S_{top} be the collection of indices t for the top T_{max} queries with highest error $|m_t - q_t(A)|$
Let $S_{\text{threshold}} = \left\{t \mid t \in S_{\text{top}}, |m_t - q_t(A)| \geq \frac{a_{\text{max}}}{2}\right\}$ be the collection of indices $t \in S_{\text{threshold}}$ such that the error for q_t is greater than $\frac{a_{\text{max}}}{2}$
for $t \in$ Randomize$(S_{\text{threshold}})$ **do**
Let A be a distribution s.t.

$$A(x) \propto A(x) \exp\left\{\hat{q}_t(x)\left(\frac{m_t - q_t(A)}{2}\right)\right\}$$

where

$$\hat{q}(x) = \begin{cases} q(x) & q \in \mathcal{Q}_G \\ \frac{1}{M} q(x) & q \in \mathcal{Q}_I \end{cases}$$

end for
Output: A

A.2.2 H-GEM

In Section 4.2, we proposes a new neural architecture structure for modeling hierarchical data, while also providing details on how the output probabilities can be used to either answer hierarchical queries directly or sample synthetic data. We note that our overall method, H-GEM shares the same

training procedure as GEM under the Adaptive Measurements framework. However for the sake of completeness, we include this procedure in Algorithms 3 and 4 for readers to refer to. Finally, we note that in our experiments, F is a multi-headed MLP with two hidden layers (size 256 and 512).

Algorithm 3: H-GEM

Input: Private hierarchical dataset $D \in \mathcal{X}$, query class $\mathcal{Q} = \mathcal{Q}_G \cup \mathcal{Q}_I$
Parameters: Privacy parameter $\rho > 0$, number of iterations T, privacy weighting parameter α, batch size B, max per-round iterations T_{max}
Let N_G be the number of groups in D
Let $\varepsilon_0 = \sqrt{\frac{2\rho}{T\left(\alpha^2 + (1-\alpha)^2\right)}}$
Initialize generator network F_0
for $t = 1$ **to** T **do**
Sample: Sample $\mathbf{z} = \langle z_1 \dots z_B \rangle \sim \mathcal{N}(0, I_B)$
Select query $\tilde{q}_t \in \mathcal{Q}$ using the *exponential mechanism* with parameter ε_0 and score function

$$P[\tilde{q}_t = q] \propto \exp\left\{\frac{\alpha\varepsilon_0 N_G}{2}|q(D) - q(A_{i-1})\right\}$$

Measure: Take (via the *Gaussian mechanism*) measurement

$$m_t = \tilde{q}_t(D) + \mathcal{N}\left(0, \left(\frac{1}{N_G(1-\alpha)\varepsilon_0}\right)^2\right)$$

Update: F_t = H-GEM-Update $(F_{t-1}, \tilde{Q}_t, \tilde{M}_t, T_{\text{max}}, \gamma)$ where $\tilde{Q}_t = \langle \tilde{q}_1, \dots, \tilde{q}_t \rangle$, $\tilde{M}_t = \langle \tilde{m}_1, \dots, \tilde{m}_t \rangle$, and $\gamma =$ EMA $\left(|\tilde{Q}_t - \tilde{M}_t|\right)$
end for
Let $\theta_{out} =$ EMA $\left(\{\theta_j\}_{j=\frac{T}{2}}^{T}\right)$ where θ_j parameterizes F_j
Let F_{out} be the generator parameterized by θ_{out}
Output: $F_{out}(\mathbf{z})$

Algorithm 4: H-GEM-Update

Input: Neural network F, queries $Q = \langle q_1, \dots, q_t \rangle$, noisy measurements $M = \langle m_1, \dots, m_t \rangle$, max iterations T_{max}, stopping threshold γ
Sample $\mathbf{z} = \langle z_1 \dots z_B \rangle \sim \mathcal{N}(0, I_B)$
Let $\mathbf{a} = M - f_Q(F(\mathbf{z}))$ be errors over queries in Q (where $f_Q(\cdot) = \langle f_{q_1}(\cdot), \dots, f_{q_t}(\cdot) \rangle$)
for $t = 1$ **to** T **do**
Let $S_{\text{threshold}} = \{i \mid |a_i| \geq \gamma\}$
Let $\hat{Q} = \{q_i \mid i \in S_{\text{threshold}}\}$ and $\hat{M} = \{m_i \mid i \in S_{\text{threshold}}\}$
Update F via stochastic gradient decent according to Equation 1: $\mathcal{L}^{\text{H-GEM}}\left(\hat{Q}, \hat{M}\right)$
Resample $\mathbf{z} = \langle z_1 \dots z_B \rangle \sim \mathcal{N}(0, I_B)$ and update $\mathbf{a} = M - f_Q(F(\mathbf{z}))$
if $\max_i |a_i| < \gamma$ **then**
break;
end if
end for
Output: F