Answering Private Linear Queries Adaptively using the Common Mechanism

Yingtai Xiao[1], Guanhong Wang[2], Danfeng Zhang[1], Daniel Kifer[1]
[1]Penn State, [2]University of Maryland, College Park
yxx5224@psu.edu,guanhong@umd.edu,dbz5017@psu.edu,duk17@psu.edu

ABSTRACT

When analyzing confidential data through a privacy filter, a data scientist often needs to decide which queries will best support their intended analysis. For example, an analyst may wish to study noisy two-way marginals in a dataset produced by a mechanism $\mathcal{M}_1$. But, if the data are relatively sparse, the analyst may choose to examine noisy one-way marginals, produced by a mechanism $\mathcal{M}_2$, instead. Since the choice of whether to use $\mathcal{M}_1$ or $\mathcal{M}_2$ is data-dependent, a typical differentially private workflow is to first split the privacy loss budget ρ into two parts: ρ_1 and ρ_2, then use the first part ρ_1 to determine which mechanism to use, and the remainder ρ_2 to obtain noisy answers from the chosen mechanism. In a sense, the first step seems wasteful because it takes away part of the privacy loss budget that could have been used to make the query answers more accurate.

In this paper, we consider the question of whether the choice between $\mathcal{M}_1$ and $\mathcal{M}_2$ can be performed without wasting any privacy loss budget. For linear queries, we propose a method for decomposing $\mathcal{M}_1$ and $\mathcal{M}_2$ into three parts: (1) a mechanism $\mathcal{M}_*$ that captures their shared information, (2) a mechanism $\mathcal{M}'_1$ that captures information that is specific to $\mathcal{M}_1$, (3) a mechanism $\mathcal{M}'_2$ that captures information that is specific to $\mathcal{M}_2$. Running $\mathcal{M}_*$ and $\mathcal{M}'_1$ together is completely equivalent to running $\mathcal{M}_1$ (both in terms of query answer accuracy and total privacy cost ρ). Similarly, running $\mathcal{M}_*$ and $\mathcal{M}'_2$ together is completely equivalent to running $\mathcal{M}_2$.

Since $\mathcal{M}_*$ will be used no matter what, the analyst can use its output to decide whether to subsequently run $\mathcal{M}'_1$ (thus recreating the analysis supported by $\mathcal{M}_1$) or $\mathcal{M}'_2$ (recreating the analysis supported by $\mathcal{M}_2$), without wasting privacy loss budget.

PVLDB Reference Format:
Yingtai Xiao[1], Guanhong Wang[2], Danfeng Zhang[1], Daniel Kifer[1]. Answering Private Linear Queries Adaptively using the Common Mechanism. PVLDB, (): , .
doi:

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/cmla-psu/commonmech.

1 INTRODUCTION

Consider an analyst who is working with confidential demographics data through a differential privacy filter – the analyst poses queries and receives noisy answers. The analyst has a privacy loss budget ρ and wishes to use it to study age and race distributions in a region. The analyst wishes to get a noisy age by race marginal (115 age values and 63 race categories used by the census, for a total of 63×115 cells). But, for small populations, this marginal would be sparse and the noise would swamp the data. In that case, the analyst could prefer two one-way marginals: one marginal on race and a separate marginal on age. The analyst does not know in advance whether the two-way race by age marginal (Option 1) is better for this region or if two one-way marginals (Option 2) are better.

In a typical workflow, the analyst would split the privacy loss budget ρ into two pieces ρ_1 and ρ_2 (with $\rho_1+\rho_2 = \rho$). The first piece would be used to somehow determine which of the two options would provide a good signal to (privacy) noise ratio. For example, the analyst could ask for a noisy population total to make the decision, or could use the exponential mechanism [54], which is a common technique for selecting among several options. The remaining privacy loss budget ρ_2 would be used to provide a noisy answer to the chosen option (either a noisy age by race marginal, or the two one-way marginals).

Now, this procedure comes with some regret because, if the analyst had known in advance which option to pick, then the entire privacy loss budget ρ (instead of only ρ_2) could have been used to provide a noisy marginal, providing more accuracy. Thus the analyst may feel that ρ_1, the portion of the privacy loss budget used to select between the two options, was wasted or lost.

In this paper, we consider the question of how the analyst can choose between Options 1 and 2 so that no privacy loss budget is lost, and the entire ρ is spent on the chosen analysis. Suppose $\mathcal{M}_1$ is the mechanism used to provide noisy answers in Option 1 with privacy budget ρ and $\mathcal{M}_2$ is the mechanism used to provide noisy answers in Option 2 with privacy budget ρ. We show how to split $\mathcal{M}_1$ into two mechanisms $\mathcal{M}_*$ and $\mathcal{M}'_1$, so that running $\mathcal{M}_1$ with privacy budget ρ is completely equivalent to running $\mathcal{M}_*$ and $\mathcal{M}'_1$ together. Similarly, we split $\mathcal{M}_2$ into $\mathcal{M}_*$ and $\mathcal{M}'_2$. This "common" mechanism $\mathcal{M}_*$ represents information that is common to both $\mathcal{M}_1$ and $\mathcal{M}_2$. That is, this is a piece of information that would be provided to the analyst by either mechanism. Meanwhile, the "residual" $\mathcal{M}'_1$ encapsulates information that is specific to $\mathcal{M}_1$ (one can think of $\mathcal{M}'_1$ as the result of removing the information in $\mathcal{M}_*$ from $\mathcal{M}_1$). Similarly, $\mathcal{M}'_2$ is the information that is specific to $\mathcal{M}_2$. The analyst's workflow becomes the following.

1. Given a dataset $\mathcal{D}$, first run $\mathcal{M}_*(\mathcal{D})$ to get an output ω_*.

2a. Based on ω_*, the analyst can choose to run the residual mechanism $\mathcal{M}'_1(\mathcal{D})$ to get an output ω'_1,

2b. Or, based on ω_*, the analyst can instead run the residual mechanism $\mathcal{M}'_2(\mathcal{D})$ to get an output ω'_2.

Proceedings of the VLDB Endowment, Vol. , No. ISSN 2150-8097.
doi:

If the analyst runs $\mathcal{M}_*$ followed by $\mathcal{M}'_1$, the total privacy budget spent is ρ and the resulting outputs ω_* and ω'_1 provide the same information as if $\mathcal{M}_1$ had been run with privacy budget ρ. Similarly, if the analyst runs $\mathcal{M}_*$ followed by $\mathcal{M}'_2$, then the total privacy budget spent again is ρ and the resulting outputs ω_* and ω'_2 provide the same information as running $\mathcal{M}_2$ with privacy budget ρ. Thus, the analyst adaptively chooses which mechanism to run without any wasted privacy budget.

Our technique works for any mechanism that answers linear queries with Gaussian noise. It is compatible with Renyi differential privacy [56], zCDP [8], Gaussian Differential Privacy [20], (ϵ, δ)-differential privacy [21], and personal differential privacy [24].

This kind of scenario, where an analyst needs to choose between pre-specified analyses depending on which provides an appropriate signal to (privacy) noise ratio, is expected to become more common. For example, the 2020 Census Detailed Demographic and Housing Characteristics data products are going to include sex-by-age tabulations where the binning of age in a region is data-dependent [29]. For small regions, only the population totals will be published. For more populous regions, the age will be binned into 4, 9, or 23 buckets, depending on how populous the region is. To choose which bucketization to use, noise will be first added to the population in a region [29]. This noisy count will be checked against manually-specified thresholds to determine the buckets to use. We empirically show that our proposed approach is better at selecting the correct analysis, reduces the need for manual tuning, and uses all of the privacy loss budget on the sex-by-age histograms instead of taking away some of it for the purposes of selecting the histogram to use.

Another application for our framework is privacy budget savings in interactive query systems (e.g., Apex [27, 50]). Suppose an analyst has already run mechanisms $\mathcal{M}_a, \mathcal{M}_b, \mathcal{M}_c$ to obtain noisy answers to linear queries, and then wishes to run $\mathcal{M}_d$. In our framework one can view the set $\{\mathcal{M}_a, \mathcal{M}_b, \mathcal{M}_c\}$ as the combined linear query mechanism $\mathcal{M}_1$ and one can view $\mathcal{M}_d$ as $\mathcal{M}_2$. Then the common mechanism $\mathcal{M}_*$ is the part of $\mathcal{M}_2$ that is already answerable from $\mathcal{M}_1$ (i.e., from the results of $\mathcal{M}_a, \mathcal{M}_b, \mathcal{M}_c$). The residual $\mathcal{M}'_2$ is the extra information needed to recover the answer to $\mathcal{M}_2$. Running $\mathcal{M}'_2$ would therefore save the analyst some privacy budget as it would avoid re-asking for information the analyst already has.

The contributions of this paper are the following:

- A framework for adaptively choosing between two linear mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$ without additional expenditure of privacy loss budget.
- We formalize the definition of the common mechanism $\mathcal{M}_*$ of $\mathcal{M}_1$ and $\mathcal{M}_2$ as an optimization problem, and also formalize the associated residual mechanisms $\mathcal{M}'_1, \mathcal{M}'_2$. This framework can be extended to choosing among multiple mechanisms (e.g., $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \dots$), but there are analytical solutions when choosing among two mechanisms.
- Algorithms for computing common and residual mechanisms.
- We give suggestions on how to use the output of the common mechanism to decide between $\mathcal{M}_1$ and $\mathcal{M}_2$, thus providing another tool for the construction of differentially private systems.
- We demonstrate the efficacy of this approach using real datasets and apply it to a real-world application involving census data.

We present notation and background in Section 2. We formally define the problem statement in Section 3. Related work is discussed in Section 4. Algorithms for the common and residual mechanisms are in Section 5. Suggestions on how to decide between mechanisms based on the output of the common mechanism are in Section 6. Experiments are in Section 7 and conclusions are in Section 8. All proofs can be found in the full version of this paper [70].

2 NOTATION AND BACKGROUND

In this section, we explain our notation (summarized in Table 1) and provide background information on differential privacy and the type of mechanisms we consider.

We denote vectors as bold lower-case letters (e.g., $\mathbf{x}$), matrices as bold upper-case (e.g., $\mathbf{B}$), scalars as non-bold lower-case (e.g., σ). If $\mathbf{A}$ and $\mathbf{B}$ are positive semidefinite matrices, we say $\mathbf{B} \preceq \mathbf{A}$ if $\mathbf{A} - \mathbf{B}$ is positive semidefinite ($\succeq$ is defined analogously). The relation $\preceq$ defines a partial ordering on semidefinite matrices called the *Loewner* order [33].

A **dataset** $\mathcal{D}$ is a table of records. Following earlier work on differentially private linear queries [40, 69, 72, 73], we assume the record attributes are categorical (or have been discretized). As in prior work, we represent the dataset $\mathcal{D}$ as a vector $\mathbf{x}$ of counts and we refer to it as the **data vector**. That is, letting $\{t_0, t_1, \dots t_{d-1}\}$ be the set of possible record values, $\mathbf{x}[i]$ is the number of times record t_i appears in $\mathcal{D}$. For example, if each record consists of two attributes, *adult* (yes/no) and *Hispanic* (yes/no), then there are 4 possible types of records, which are t_0 = "not adult, not Hispanic", t_1 ="adult, not Hispanic", t_2 ="not adult, Hispanic", t_3 ="adult, Hispanic". In our representation, $\mathbf{x}[3]$ is the number of Hispanic adults in the dataset $\mathcal{D}$. We say that two dataset vectors $\mathbf{x}$ and $\mathbf{x}'$ are **neighboring** (denoted as $\mathbf{x} \sim \mathbf{x}'$) if $\mathbf{x}$ can be obtained from $\mathbf{x}'$ by adding or subtracting 1 from some component of $\mathbf{x}'$ (this means $||\mathbf{x} - \mathbf{x}'||_1 = 1$) – this is the same as adding or removing 1 person from the underlying dataset.

A single **linear query** $\mathbf{q}$ is a vector, whose answer is $\mathbf{q} \cdot \mathbf{x}$. A **query set** is a set of m linear queries represented by an $m \times d$ matrix $\mathbf{B}$, where each row corresponds to a single linear query. We let k denote the rank of $\mathbf{B}$. The answers to the queries are obtained by matrix multiplication: $\mathbf{Bx}$. Continuing our running example of a two-attribute dataset, if $\mathbf{B} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$, then this is a set of two queries. The first query is the number of times records t_1 or t_3 appear in the dataset (i.e., the number of adults) and the second query is the total number of people. A **mechanism** $\mathcal{M}$ is an algorithm whose input is the confidential data (either $\mathcal{D}$ or $\mathbf{x}$) and whose output ω is considered safe to release (because it protects privacy). When we have two mechanisms $\mathcal{M}_a$ and $\mathcal{M}_b$, we let $(\mathcal{M}_a, \mathcal{M}_b)$ denote the mechanism that runs both of them on the data and releases their results. In other words, its output is $(\mathcal{M}_a(\mathcal{D}), \mathcal{M}_b(\mathcal{D}))$.

2.1 Differential Privacy

Differential Privacy [8, 20–22, 56] is a family of privacy definitions that place restrictions on how a mechanism $\mathcal{M}$ can work. It has become a de facto standard for protecting confidentiality when creating publicly available data products, with an ever-increasing list of real-world deployments, including the U.S. Census Bureau [1, 10, 49], Uber [35, 36], Apple [64], Facebook [55, 57], Microsoft

[17], and Google [26, 28, 66]. Differential privacy provides a rigorous plausible deniability guarantee – it limits the ability of an attacker to determine whether a target person's record was in the dataset or not. The most common variation of differential privacy is:

Definition 2.1 (Approximate Differential Privacy [21]). *Given privacy parameters $\epsilon > 0$ and $\delta \in (0,1)$, a randomized algorithm $\mathcal{M}$ satisfies (ϵ,δ)-differential privacy if for all pairs of neighboring dataset vectors $\mathbf{x}$ and $\mathbf{x}'$ and all sets S, the following equations hold:*

$$P(\mathcal{M}(\mathbf{x}) \in S) \leq e^{\epsilon} P(\mathcal{M}(\mathbf{x}') \in S) + \delta$$

A mechanism M typically satisfies approximate differential privacy for infinitely many (ϵ,δ) pairs, which defines a curve in space. There is also another popular variant known as *zero-concentrated differential privacy* (zCDP):

Definition 2.2 (zCDP [8]). *Given a privacy parameters $\rho > 0$, a randomized algorithm $\mathcal{M}$ satisfies ρ-zCDP if for all pairs of neighboring dataset vectors $\mathbf{x}$ and $\mathbf{x}'$ and all numbers $\alpha > 1$,*

$$\int_{\omega} P(\mathcal{M}(\mathbf{x}) = \omega) \frac{P(\mathcal{M}(\mathbf{x}) = \omega)^{\alpha-1}}{P(\mathcal{M}(\mathbf{x}') = \omega)^{\alpha-1}}\, d\omega \leq e^{(\alpha-1)\alpha\rho},$$

where $P(\mathcal{M}(\mathbf{x}) = \omega)$ is interpreted as a probability density function in the continuous case.

The parameters (ϵ,δ) are known as the *privacy loss budget* of approximate differential privacy and the privacy parameter ρ is known as the privacy loss budget of zCDP. Note that the privacy parameter of zCDP is difficult to interpret (see [37] for an extended discussion) but easier to compute than approximate differential privacy. Thus, one typically determines the zCDP privacy loss parameter ρ of a mechanism $\mathcal{M}$ and then converts it to ϵ and δ for interpretability [8, 12].

Each version of differential privacy also has a "personalized" version, in which each possible record type t_i is assigned a privacy loss budget. The privacy parameter for record type t_i can be obtained from Definition 2.1 or 2.2 by considering only neighbors $\mathbf{x}$ and $\mathbf{x}'$ that differ in their i^{th} coordinate [24].

2.2 The Linear Gaussian Mechanism

The linear Gaussian mechanism adds Gaussian noise to the output of linear queries and is compatible with many versions of differential privacy. It is defined as follows.

Definition 2.3 (Linear Gaussian Mechanism). *Given a query matrix $\mathbf{B}$ and nonsingular covariance matrix Σ (not necessarily diagonal), the linear Gaussian mechanism $\mathcal{M}$ is defined as $\mathcal{M}(\mathbf{x}) = \mathbf{B}\mathbf{x} + N(\mathbf{0},\Sigma)$. The quantity $\mathbf{B}^T\Sigma^{-1}\mathbf{B}$ is called the* ***privacy cost matrix*** *of $\mathcal{M}$.*

The importance of the privacy cost matrix is that the privacy parameters of the Gaussian mechanism for (ϵ,δ)-differential privacy and ρ-zCDP (both the basic and personalized versions) are all functions of this privacy cost matrix, as the following result shows:[1]

Lemma 2.4 ([69]). *Let $\mathcal{M}$ be the linear Gaussian mechanism (Definition 2.3) defined as $\mathcal{M}(\mathbf{x}) = \mathbf{B}\mathbf{x} + N(\mathbf{0},\Sigma)$ with privacy cost matrix $\mathbf{C} = \mathbf{B}^T\Sigma^{-1}\mathbf{B}$. Let c_i be the i^{th} diagonal entry of $\mathbf{C}$ and let $c_{max} = \max_i c_i$ be the largest diagonal entry of $\mathbf{C}$. Let Φ be the CDF of the standard normal distribution. Then:*

- *$\mathcal{M}$ satisfies (ϵ,δ)-differential privacy for*
$$\delta = \Phi\left(\frac{\sqrt{c_{max}}}{2} - \frac{\epsilon}{\sqrt{c_{max}}}\right) - e^{\epsilon}\Phi\left(-\frac{\sqrt{c_{max}}}{2} - \frac{\epsilon}{\sqrt{c_{max}}}\right)$$
and δ is an increasing function of c_{max}. In particular, this means the entire (ϵ,δ) curve of $\mathcal{M}$ is determined by c_{max}.
- *The personalized approximate differential privacy parameters (ϵ_i,δ_i) for record type t_i are obtained from the formula:*
$$\delta_i = \Phi\left(\frac{\sqrt{c_i}}{2} - \frac{\epsilon_i}{\sqrt{c_i}}\right) - e^{\epsilon_i}\Phi\left(-\frac{\sqrt{c_i}}{2} - \frac{\epsilon_i}{\sqrt{c_i}}\right)$$
- *$\mathcal{M}$ satisfies ρ-zCDP for $\rho = c_{max}/2$.*
- *The personal zCDP privacy parameter for record t_i is $c_i/2$.*

[1] This is also true of Renyi [56] and Gaussian [20] differential privacy.

Table 1: Table of Notation

$\mathcal{D}$:	Dataset
$\mathbf{x}$:	Data vector representation of $\mathcal{D}$
d:	Number of possible records
$\mathcal{M}$:	Mechanism.
ω:	Output of a mechanism.
$(\mathcal{M}_a, \mathcal{M}_b)$	Combined mechanism that runs $\mathcal{M}_a$ and $\mathcal{M}_b$
$\mathbf{B}$:	Query matrix.
m:	Number of rows of $\mathbf{B}$ ($\mathbf{B}$ has size $m \times d$).
k:	Rank of $\mathbf{B}$.
Σ:	Covariance matrix.
$\mathbf{B}^T\Sigma^{-1}\mathbf{B}$:	Privacy cost matrix of mechanism $\mathcal{M}(\mathbf{x}) = \mathbf{B}\mathbf{x} + N(\mathbf{0},\Sigma)$.
$\preceq, \succeq$	Loewner order ($\mathbf{A_2} \preceq \mathbf{A_2}$ iff $\mathbf{A_1} - \mathbf{A_2}$ is positive semidefinite)

3 PROBLEM DEFINITION AND SOLUTION OVERVIEW

The motivation for our problem is the following. An analyst is interested in obtaining noisy linear query answers either from mechanism $\mathcal{M}_1$ or $\mathcal{M}_2$, defined as follows:

$$\mathcal{M}_1(\mathbf{x}) = \mathbf{B}_1\mathbf{x} + N(0,\Sigma_1)$$
$$\mathcal{M}_2(\mathbf{x}) = \mathbf{B}_2\mathbf{x} + N(0,\Sigma_2).$$

where $\mathcal{M}_1$ and $\mathcal{M}_2$ both satisfy zCDP with the same privacy parameter ρ.[2] For example, $\mathcal{M}_1$ could compute all the one-way marginals of a dataset, and $\mathcal{M}_2$ could compute all the two-way marginals. However, the choice of which mechanism to use depends on the properties of the data that the analyst does not know.

For instance, if the dataset is "large enough" then the noisy two-way marginals (i.e., $\mathcal{M}_2$) will be very accurate with low relative error. Otherwise, the analyst would prefer to use the noisy one-way marginals produced by $\mathcal{M}_1$. The problem is that the analyst does not know whether the dataset is "large enough", or even how to precisely define what "large enough" means (i.e., exactly how many records are needed for the dataset to be considered large enough).

Thus, the analyst needs extra information about the data in order to make a choice between $\mathcal{M}_1$ and $\mathcal{M}_2$. One option is to take some privacy budget ρ_0 away from $\mathcal{M}_1$ and $\mathcal{M}_2$. This ρ_0 would be assigned to some other mechanism $\mathcal{M}^{\circ}$ that queries that data.

[2] For concreteness, we focus on zCDP, but, as we show later, this approach works for any post-processing invariant privacy definition.

Based on the answers to $\mathcal{M}^\circ$, the analyst would modify either $\mathcal{M}_1$ or $\mathcal{M}_2$ to use the remaining privacy budget $\rho - \rho_0$ (by rescaling the Gaussian covariance matrices Σ_1 and Σ_2) and then run it on the data. This option produces noisier answers than the analyst wants, because only $\rho - \rho_0$ instead of ρ is allocated towards the noisy answers. In this sense, this ρ_0 can feel like a wasted expenditure of privacy loss budget since it takes away from the accuracy of the query answers that the analyst desires.

In this paper, we consider a second option – whether the analyst can make an informed choice without wasting any privacy loss budget. The main idea is to rewrite mechanism $\mathcal{M}_1$ as a sequence of two mechanisms $\mathcal{M}_*$ and $\mathcal{M}'_1$ such that running $\mathcal{M}_1$ is exactly equivalent, both in terms of information content and privacy cost, to running $\mathcal{M}_*$ and $\mathcal{M}'_1$ together. We similarly decompose $\mathcal{M}_2$ into $\mathcal{M}_*$ and $\mathcal{M}'_2$. This means that if the analyst wants to run $\mathcal{M}_1$ or $\mathcal{M}_2$, then $\mathcal{M}_*$ is going to be run no matter what. Thus the analyst can first run $\mathcal{M}_*$ and then can decide whether to run $\mathcal{M}'_1$ (to finish the execution of $\mathcal{M}_1$) or to run $\mathcal{M}'_2$ (to finish the execution of $\mathcal{M}_2$). We refer to $\mathcal{M}_*$ as the mechanism that is *common* to $\mathcal{M}_1$ and $\mathcal{M}_2$ since it represents the information that they share. We refer to $\mathcal{M}'_1$ and $\mathcal{M}'_2$ as the *residual* mechanisms since they represent the information that is specific to $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively.

We next formalize this discussion. We examine what it means for two mechanisms to be equivalent in Section 3.1, we formally define the common mechanism of $\mathcal{M}_1$ and $\mathcal{M}_2$ in Section 3.2, we formally define the residual mechanisms in Section 3.3, and then we present the technical problem statement in Section 3.4. The use of more than 2 mechanisms in the framework is discussed in Section 3.5. Limitations are discussed in Section 3.6.

3.1 Exact Answerability and Equivalence of Linear Gaussian Mechanisms

Suppose there are two mechanisms $\mathcal{M}_a$ and $\mathcal{M}_b$ and a randomized postprocessing algorithm $\mathcal{A}$ such that for all data vectors $\mathbf{x}$, the output distribution of $\mathcal{M}_b(\mathbf{x})$ is the same as the output distribution of $\mathcal{A}(\mathcal{M}_a(\mathbf{x}))$. This means that we can simulate the output of $\mathcal{M}_b$ by taking the output of $\mathcal{M}_a$ and feeding it to $\mathcal{A}$.

When this is the case, we say that $\mathcal{M}_b$ is *exactly answerable* from $\mathcal{M}_a$ and it means that $\mathcal{M}_a$ produces at least as much information as $\mathcal{M}_b$. For any post-processing invariant privacy definition (such as ρ-zCDP, Gaussian differential privacy, approximate differential privacy, etc.), the privacy cost of $\mathcal{M}_a$ is also at least as large as the privacy cost of $\mathcal{M}_b$ (e.g., the ρ parameter of $\mathcal{M}_a$ under zCDP is greater than or equal to the ρ parameter of $\mathcal{M}_b$). For linear Gaussian mechanisms, exact answerability can be defined as follows.

Definition 3.1. *Let $\mathcal{M}_a(\mathbf{x}) = \mathbf{B}_a\mathbf{x} + N(0, \Sigma_a)$ and $\mathcal{M}_b(\mathbf{x}) = \mathbf{B}_b\mathbf{x} + N(0, \Sigma_b)$ be linear Gaussian mechanisms. We say that $\mathcal{M}_b$ is exactly answerable from $\mathcal{M}_a$ if there exist matrices $\mathbf{A}$ and $\mathbf{C}$ such that for every $\mathbf{x}$, $\mathcal{M}_b(\mathbf{x})$ has the same distribution as $\mathbf{A}\mathcal{M}_a(\mathbf{x}) + \mathbf{C}N(0, \mathbf{I})$, where $\mathbf{I}$ is the identity matrix. In other words, $\mathcal{M}_b$ can be obtained from $\mathcal{M}_a$ by applying a linear transformation and adding additional noise (that does **not** depend on the data).*

For linear Gaussian mechanisms, exact answerability is easy to check using the following result whose proof is straightforward.

Lemma 3.2. *Let $\mathcal{M}_a(\mathbf{x}) = \mathbf{B}_a\mathbf{x} + N(0, \Sigma_a)$ and $\mathcal{M}_b(\mathbf{x}) = \mathbf{B}_b\mathbf{x} + N(0, \Sigma_b)$ be linear Gaussian mechanisms. $\mathcal{M}_b$ is exactly answerable from $\mathcal{M}_a$ if and only if there exist matrices $\mathbf{A}$ and $\mathbf{C}$ such that:*

$$\mathbf{B}_b = \mathbf{A}\mathbf{B}_a$$
$$\Sigma_b = \mathbf{A}\Sigma_a\mathbf{A}^T + \mathbf{C}\mathbf{C}^T$$

Example 3.3. Suppose $\mathbf{x}$ is two-dimensional and consider the mechanisms:

$$\mathcal{M}_a(\mathbf{x}) = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x} + N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}\right)$$
$$\mathcal{M}_b(\mathbf{x}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x} + N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 4/3 & -2/3 \\ -2/3 & 4/3 \end{bmatrix}\right)$$

Both mechanisms satisfy ρ-zCDP with $\rho = 1/2$. For any $\mathbf{x}$, the output distribution of $\mathcal{M}_b$ is the multivariate Gaussian with mean $\mathbf{x}$ and covariance matrix $\begin{bmatrix} 4/3 & -2/3 \\ -2/3 & 4/3 \end{bmatrix}$. This is exactly the same as the distribution of $\begin{bmatrix} \frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ \frac{1}{3} & \frac{-1}{3} & \frac{2}{3} \end{bmatrix} \mathcal{M}_a(\mathbf{x})$ and so $\mathcal{M}_b$ is exactly answerable from $\mathcal{M}_a$.

Similarly, the output distribution of $\mathcal{M}_a(\mathbf{x})$ is the same as the distribution of $\begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \mathcal{M}_b(\mathbf{x}) + \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)$ and thus $\mathcal{M}_a$ is also exactly answerable from $\mathcal{M}_b$.

Remark 3.4. *It is noteworthy that even though $\mathcal{M}_a$ and $\mathcal{M}_b$ in Example 3.3 are exactly answerable from each other, $\mathcal{M}_a$ is obtained from $\mathcal{M}_b$ by linear postprocessing followed by noise addition. The reason is because the rows of the query matrix $\mathbf{B}_a = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$ in mechanism $\mathcal{M}_a$ are linearly dependent. The sole purpose of the noise is to convert the 2-dimensional Gaussian distribution obtained by linear postprocessing of $\mathcal{M}_b$ into the 3-dimensional Gaussian distribution that $\mathcal{M}_a$ uses. This noise does not add or remove privacy, since $\mathcal{M}_b$ can also be obtained from $\mathcal{M}_a$ by linear postprocessing. Another way to view this phenomenon is to note that the linear dependency in $\mathbf{B}_a$ causes inconsistency – the first component of the output of $\mathcal{M}_a$ is a noisy sum, and the sum of the 2nd and 3rd components is also a version of the noisy sum. Enforcing consistency [32, 40] would convert $\mathcal{M}_a$ into $\mathcal{M}_b$, and the noise that is removed by consistency is the same noise that is added back when reconstructing $\mathcal{M}_a$ from $\mathcal{M}_b$.*

One observation we make from Example 3.3 is that proving exact answerability can be cumbersome because one must produce the matrices $\mathbf{A}$ and $\mathbf{C}$ as in Definition 3.1. The following result allows us to check exact answerability in a more mechanical way.

Theorem 3.5. *Lets $\mathcal{M}_a(\mathbf{x}) = \mathbf{B}_a\mathbf{x} + N(\mathbf{0}, \Sigma_a)$ and $\mathcal{M}_b(\mathbf{x}) = \mathbf{B}_b\mathbf{x} + N(\mathbf{0}, \Sigma_b)$ be linear Gaussian mechanisms. $\mathcal{M}_b$ is exactly answerable from $\mathcal{M}_a$ if and only if $\mathbf{B}_b^T\Sigma_b^{-1}\mathbf{B}_b \preceq \mathbf{B}_a^T\Sigma_a^{-1}\mathbf{B}_a$ (i.e., $\mathbf{B}_a^T\Sigma_a^{-1}\mathbf{B}_a - \mathbf{B}_b^T\Sigma_b^{-1}\mathbf{B}_b$ is positive semidefinite, and hence its eigenvalues are non-negative).*

One interesting consequence of Theorem 3.5 is that exact answerability depends on the quantities $\mathbf{B}_a^T\Sigma_a^{-1}\mathbf{B}_a$ and $\mathbf{B}_b^T\Sigma_b^{-1}\mathbf{B}_b$, which are the privacy cost matrices of mechanisms $\mathcal{M}_a$ and $\mathcal{M}_b$, respectively (see Section 2.2). In particular, if $\mathbf{B}_a^T\Sigma_a^{-1}\mathbf{B}_a = \mathbf{B}_b^T\Sigma_b^{-1}\mathbf{B}_b$ then $\mathcal{M}_a$ and $\mathcal{M}_b$ are not only exactly answerable from each other, but also have the exact same privacy cost under zCDP, approximate differential privacy, Gaussian differential privacy, and their personalized versions (as well as any other postprocessing invariant privacy definitions). In this sense, they are completely identical in terms of information content and privacy. This leads to the following definition.

Definition 3.6 (Equivalence). *Two mechanisms $\mathcal{M}_a$ and $\mathcal{M}_b$ are* equivalent *if $\mathcal{M}_a$ is exactly answerable from $\mathcal{M}_b$ and vice versa. In particular, if $\mathcal{M}_a(\mathbf{x}) = \mathbf{B}_a\mathbf{x}+N(\mathbf{0},\Sigma_a)$ and $\mathcal{M}_b(\mathbf{x}) = \mathbf{B}_b\mathbf{x}+N(\mathbf{0},\Sigma_b)$ are linear Gaussian mechanisms, then $\mathcal{M}_a$ and $\mathcal{M}_b$ are* equivalent *if $\mathbf{B}_a^T\Sigma_a^{-1}\mathbf{B}_a = \mathbf{B}_b^T\Sigma_b^{-1}\mathbf{B}_b$.*

3.2 Shared Information & Common Mechanism

Intuitively, a piece of information is shared by $\mathcal{M}_1$ and $\mathcal{M}_2$ if that information can be derived from the output of $\mathcal{M}_1$ *and* it can also be derived from the output of $\mathcal{M}_2$. We formalize "information" as a mechanism $\mathcal{M}_c$ that can be answered exactly from $\mathcal{M}_1$ and from $\mathcal{M}_2$. We call it a *common* mechanism of $\mathcal{M}_1$ and $\mathcal{M}_2$.

Definition 3.7 (Common Mechanism). *A mechanism $\mathcal{M}_c$ is common to $\mathcal{M}_1$ and $\mathcal{M}_2$ if $\mathcal{M}_c$ is exactly answerable from each one of them. When they are all linear Gaussian mechanisms:*

$$\mathcal{M}_1(\mathbf{x}) = \mathbf{B}_1\mathbf{x} + N(\mathbf{0},\Sigma_1)$$
$$\mathcal{M}_2(\mathbf{x}) = \mathbf{B}_2\mathbf{x} + N(\mathbf{0},\Sigma_2)$$
$$\mathcal{M}_c(\mathbf{x}) = \mathbf{B}_c\mathbf{x} + N(\mathbf{0},\Sigma_c)$$

then by Theorem 3.5, $\mathcal{M}_c$ is common to $\mathcal{M}_1$ and $\mathcal{M}_2$ whenever:

$$\mathbf{B}_c^T\Sigma_c^{-1}\mathbf{B}_c \preceq \mathbf{B}_a^T\Sigma_a^{-1}\mathbf{B}_a \text{ and}$$
$$\mathbf{B}_c^T\Sigma_c^{-1}\mathbf{B}_c \preceq \mathbf{B}_b^T\Sigma_b^{-1}\mathbf{B}_b$$

Example 3.8. Consider the following four mechanisms, where $\mathcal{M}_1, \mathcal{M}_3, \mathcal{M}_4$ are noisy sum queries with variances 1, 2, and 1.5, respectively, while $\mathcal{M}_2$ is a combination of a noisy sum query with variance 2 and a noisy identity query with variance 2:

$$\mathcal{M}_1(\mathbf{x}) = [\,1\ 1\ 1\,]\mathbf{x} + N(0, \sigma^2 = 1)$$
$$\mathcal{M}_2(\mathbf{x}) = \begin{bmatrix} 1&1&1\\1&0&0\\0&1&0\\0&0&1 \end{bmatrix}\mathbf{x} + N\left(\begin{bmatrix}0\\0\\0\\0\end{bmatrix}, \begin{bmatrix}2&0&0&0\\0&2&0&0\\0&0&2&0\\0&0&0&2\end{bmatrix}\right)$$
$$\mathcal{M}_3(\mathbf{x}) = [\,1\ 1\ 1\,]\mathbf{x} + N(0, \sigma^2 = 2)$$
$$\mathcal{M}_4(\mathbf{x}) = [\,1\ 1\ 1\,]\mathbf{x} + N(0, \sigma^2 = 1.5)$$

The mechanism $\mathcal{M}_3$ is common to both $\mathcal{M}_1$ and $\mathcal{M}_2$ because it can be answered using only the output of either mechanism (with no additional access to the underlying data). For example, $\mathcal{M}_3$ can be answered by adding noise to the output of $\mathcal{M}_1$ as follows: $\mathcal{M}_3(\mathbf{x}) = \mathcal{M}_1(\mathbf{x}) + N(0,1)$. Also $\mathcal{M}_3$ can be answered from $\mathcal{M}_2$ by taking the noisy sum that $\mathcal{M}_2$ directly provides (mathematically, $\mathcal{M}_3(\mathbf{x}) = [\,1\ 0\ 0\ 0\,]\mathcal{M}_2(\mathbf{x})$). However, $\mathcal{M}_4$ is also common to $\mathcal{M}_1$ and $\mathcal{M}_2$ as we can see from the following equations:

$$\mathcal{M}_4(\mathbf{x}) = \mathcal{M}_1(\mathbf{x}) + N(0, 0.5)$$
$$\mathcal{M}_4(\mathbf{x}) = [0.75, 0.25, 0.25, 0.25]\mathcal{M}_2(\mathbf{x})$$

As we see from Example 3.8, both $\mathcal{M}_3$ and $\mathcal{M}_4$ are common mechanisms for $\mathcal{M}_1$ and $\mathcal{M}_2$, and so they both capture information that is shared by $\mathcal{M}_1$ and $\mathcal{M}_2$. However, $\mathcal{M}_4$ clearly captures more of this shared information than $\mathcal{M}_3$. This leads to a concept of a *maximally* common mechanism.

Definition 3.9 (Maximally common mechanism). *A mechanism $\mathcal{M}_*$ is maximally common to $\mathcal{M}_1$ and $\mathcal{M}_2$ if (1) $\mathcal{M}_*$ is common to $\mathcal{M}_1$ and $\mathcal{M}_2$, (2) if there is another mechanism $\mathcal{M}^\dagger$ that is common to $\mathcal{M}_1$ and $\mathcal{M}_2$ and if $\mathcal{M}_*$ is exactly answerable from $\mathcal{M}^\dagger$, then $\mathcal{M}_*$ and $\mathcal{M}^\dagger$ are equivalent.*

It turns out that $\mathcal{M}_4$ is a maximally common mechanism to $\mathcal{M}_1$ and $\mathcal{M}_2$. We show how to compute maximally common mechanisms in Section 5.

3.3 Decomposition into Common and Residual Mechanisms

Now that a maximal common mechanism $\mathcal{M}_*$ for $\mathcal{M}_1$ and $\mathcal{M}_2$ has been defined, we next define the residual mechanisms $\mathcal{M}_1'$ and $\mathcal{M}_2'$. Intuitively, $\mathcal{M}_1'$ (resp., $\mathcal{M}_2'$) represents the least amount of additional information that, when combined with $\mathcal{M}_*$ allows us to recreate $\mathcal{M}_1$ (resp., $\mathcal{M}_2$). Alternatively, $\mathcal{M}_1'$ (resp., $\mathcal{M}_2'$) is the result of "subtracting away" the information about $\mathcal{M}_*$ from $\mathcal{M}_1$ (resp., $\mathcal{M}_2$). Recalling that the notation $(\mathcal{M}_*, \mathcal{M}_1')$ is a mechanism that runs both $\mathcal{M}_*$ and $\mathcal{M}_1'$ on the data and releases their outputs, we can formally define residual mechanisms as follows:

Definition 3.10 (Residual Mechanism). *Given mechanisms $\mathcal{M}_1$, $\mathcal{M}_2$ and a maximally common mechanism $\mathcal{M}_*$, we say that $\mathcal{M}_1'$ and $\mathcal{M}_2'$ are* residual *mechanisms if:*

- *$(\mathcal{M}_*, \mathcal{M}_1')$ is equivalent (see Definition 3.6) to $\mathcal{M}_1$ and*
- *$(\mathcal{M}_*, \mathcal{M}_2')$ is equivalent to $\mathcal{M}_2$.*

Note that, by virtue of equivalence, $(\mathcal{M}_*, \mathcal{M}_1')$ has the same privacy cost as $\mathcal{M}_1$ under any postprocessing-invariant privacy definition (including all the ones studied in this paper), and similarly with $(\mathcal{M}_*, \mathcal{M}_2')$ and $\mathcal{M}_2$. The checking of equivalence between $(\mathcal{M}_*, \mathcal{M}_1')$ and $\mathcal{M}_1$ can be done using the following result:

Lemma 3.11. *Suppose that $\mathcal{M}_1(\mathbf{x}) = \mathbf{B}_1\mathbf{x} + N(\mathbf{0},\Sigma_1)$ and $\mathcal{M}_2(\mathbf{x}) = \mathbf{B}_2\mathbf{x} + N(\mathbf{0},\Sigma_2)$ are linear Gaussian mechanisms and that $\mathcal{M}_*(\mathbf{x}) = \mathbf{B}_*\mathbf{x}+N(\mathbf{0},\Sigma_*)$ is their maximally common mechanism. Then $\mathcal{M}_1'(\mathbf{x}) = \mathbf{B}_1'\mathbf{x} + N(\mathbf{0},\Sigma_1')$ and $\mathcal{M}_2'(\mathbf{x}) = \mathbf{B}_2'\mathbf{x} + N(\mathbf{0},\Sigma_2')$ are residual mechanisms if and only if:*

$$\mathbf{B}_*{}^T\Sigma_*{}^{-1}\mathbf{B}_* \ + \ (\mathbf{B}_1')^T(\Sigma_1')^{-1}\mathbf{B}_1' \ = \ \mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1$$
$$\mathbf{B}_*{}^T\Sigma_*{}^{-1}\mathbf{B}_* \ + \ (\mathbf{B}_2')^T(\Sigma_2')^{-1}\mathbf{B}_2' \ = \ \mathbf{B}_2^T\Sigma_2^{-1}\mathbf{B}_2$$

In which case $(\mathcal{M}_*, \mathcal{M}_1')$ is equivalent to $\mathcal{M}_1$ and $(\mathcal{M}_*, \mathcal{M}_2')$ is equivalent to $\mathcal{M}_2$.

Example 3.12. Consider a table with 2 attributes Att_1 and Att_2, each attribute taking 3 possible values a, b, c. The data vector $\mathbf{x}$ then has 9 components. A marginal on Att_1 then consists of 3 numbers: the number of people for which $Att_1 = a$, the number of people for which $Att_1 = b$, and the number of people for which $Att_1 = c$. The marginal on Att_2 is defined analogously. Consider a mechanism $\mathcal{M}_1$ that adds independent $N(0,1)$ noise to the marginal on Att_1 and a mechanism $\mathcal{M}_2$ that adds independent $N(0,1)$ noise to the marginal on Att_2. In matrix notation, they are represented as follows:

$$\mathcal{M}_1(\mathbf{x}) = \begin{bmatrix} 1&1&1&0&0&0&0&0&0 \\ 0&0&0&1&1&1&0&0&0 \\ 0&0&0&0&0&0&1&1&1 \end{bmatrix} \mathbf{x} + N\left(\begin{bmatrix} 0\\0\\0 \end{bmatrix}, \begin{bmatrix} 1&0&0\\0&1&0\\0&0&1 \end{bmatrix}\right)$$
$$\mathcal{M}_2(\mathbf{x}) = \begin{bmatrix} 1&0&0&1&0&0&1&0&0 \\ 0&1&0&0&1&0&0&1&0 \\ 0&0&1&0&0&1&0&0&1 \end{bmatrix} \mathbf{x} + N\left(\begin{bmatrix} 0\\0\\0 \end{bmatrix}, \begin{bmatrix} 1&0&0\\0&1&0\\0&0&1 \end{bmatrix}\right)$$

From the output of $\mathcal{M}_1$ we can add up the noisy counts of people having values a, b, and c for attribute Att_1 to get an estimate of the total number of people. This estimate has variance 3. We can do the same with $\mathcal{M}_2$ to get a noisy total with variance 3. Running either mechanism thus provides a noisy total with variance 3, and the noisy total mechanism is in fact their maximal common mechanism, and is represented as:

$$\mathcal{M}_*(\mathbf{x}) = \begin{bmatrix} 1&1&1&1&1&1&1&1&1 \end{bmatrix} \mathbf{x} + N(0, 3).$$

The corresponding residual mechanisms are:

$$\mathcal{M}'_1(\mathbf{x}) = \begin{bmatrix} 0&0&0&1&1&1&-1&-1&-1 \\ -1&-1&-1&1&1&1&0&0&0 \end{bmatrix} \mathbf{x} + N(\begin{bmatrix} 0\\0 \end{bmatrix}, \begin{bmatrix} 2&1\\1&2 \end{bmatrix})$$
$$\mathcal{M}'_2(\mathbf{x}) = \begin{bmatrix} 0&1&-1&0&1&-1&0&1&-1 \\ -1&1&0&-1&1&0&-1&1&0 \end{bmatrix} \mathbf{x} + N(\begin{bmatrix} 0\\0 \end{bmatrix}, \begin{bmatrix} 2&1\\1&2 \end{bmatrix})$$

The residual mechanism $\mathcal{M}'_1$ is answering two queries: (1) # of records with $Att_1 = b$ minus the # with $Att_1 = c$, and (2) # of records with $Att_1 = b$ minus the # with $Att_1 = a$. Both queries get variance 2 and are correlated with covariance 1. The residual mechanism $\mathcal{M}'_2$ works analogously for attribute Att_2.

The original mechanism $\mathcal{M}_1$ can be recovered from the outputs of $\mathcal{M}'_1$ and $\mathcal{M}_*$ as follows:

$$\mathcal{M}_1(\mathbf{x}) = \begin{bmatrix} 1/3 & -2/3 \\ 1/3 & 1/3 \\ -2/3 & 1/3 \end{bmatrix} \mathcal{M}'_1(\mathbf{x}) + \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \mathcal{M}_*(\mathbf{x})$$

and it is easy to check using Lemma 3.11 that $\mathcal{M}_1$ and $(\mathcal{M}_*, \mathcal{M}'_1)$ are indeed equivalent (providing the same information and having the same privacy cost).

We show how to compute the residual mechanisms in Section 5.

3.4 Formalizing Decision Making with the Common Mechanism

Having formally defined the (maximally) common mechanism $\mathcal{M}_*$ of linear Gaussian mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$, and having defined the residual mechanisms $\mathcal{M}'_1$ and $\mathcal{M}'_2$, the formal problem statement can be defined as the following sequence of steps:

(1) Given two linear Gaussian query mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$, compute their maximal common mechanism $\mathcal{M}_*$ (algorithms are provided in Section 5).
(2) Given $\mathcal{M}_1, \mathcal{M}_2$ and $\mathcal{M}_*$, compute the residual mechanisms $\mathcal{M}'_1$ and $\mathcal{M}'_2$ (algorithms are provided in Section 5).
(3) Run $\mathcal{M}_*$ on the data to produce an output ω_*.
(4) Based on ω_*, decide whether to run the residual mechanism $\mathcal{M}'_1$ or $\mathcal{M}'_2$. An analyst is free to choose how to make a decision based on ω^*, but for more automated approaches, we provide suggestions in Section 6.
(5) Based on the decision, either run $\mathcal{M}'_1$ on the data (and combine the result with ω_* to obtain an answer to $\mathcal{M}_1$) or run $\mathcal{M}'_2$ on the data (and combine the result with ω_* to obtain an answer to $\mathcal{M}_2$). Algorithms for recovering $\mathcal{M}_1$ from $(\mathcal{M}_*, \mathcal{M}'_1)$ and recovering $\mathcal{M}_2$ from $(\mathcal{M}_*, \mathcal{M}'_2)$ are given in Section 5.

3.5 The Case of Multiple Mechanisms

There are several ways in which multiple mechanisms can appear in this framework. For example, one may want to choose between two sets of mechanisms: either $\{\mathcal{M}_a, \mathcal{M}_b, \mathcal{M}_c\}$ or $\{\mathcal{M}_d, \mathcal{M}_e, \mathcal{M}_f\}$. One can represent $\{\mathcal{M}_a, \mathcal{M}_b, \mathcal{M}_c\}$ as a single Gaussian linear query mechanism $\mathcal{M}_1$ by vertically stack their query matrices to get $\mathbf{B}_1 = \begin{bmatrix} \mathbf{B}_a \\ \mathbf{B}_b \\ \mathbf{B}_c \end{bmatrix}$ and setting the covariance matrix to be $\Sigma_1 = \begin{bmatrix} \Sigma_a & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Sigma_b & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Sigma_c \end{bmatrix}$. $\mathcal{M}_2$ then takes a similar representation. Effectively, this becomes a choice between two mechanisms and so the closed form solutions in Section 5 can be applied.

A different situation occurs when one wants to choose to among 3 or more mechanisms: either run $\mathcal{M}_1$ or run $\mathcal{M}_2$ or run $\mathcal{M}_3$ (as with the census case study in Section 7.5). In this case, one would like to find a mechanism $\mathcal{M}_*$ that is common to all three of them and make the selection based on the output of $\mathcal{M}_*$. Our framework can handle this setting, but we do not have an analystical solution – finding $\mathcal{M}_*$ requires solving an optimization problem that we discuss in Section 5.1.

3.6 Limitations

Our algorithms have the following limitations. (1) They represent the query matrices explicitly, which limits scalability to large domains. Using kronecker product representations, as in HDMM [51] is an area of future work for improving scalability. (2) Our algorithms require the mechanisms to answer linear queries with Gaussian noise (but $\mathcal{M}_1$ and $\mathcal{M}_2$ can have different covariance matrices and different means). Extensions to other noise distributions or arbitrary mechanisms are open problems. A special case is when the analyst gets $f(\mathcal{M}_1)$ or $g(\mathcal{M}_2)$, where f and g are postprocessing functions. If f and g are both linear, then the resulting mechanisms are still linear Gaussian mechanisms and our algorithms handles them. If f and g are non-linear then computing the common mechanism is an open problem, but the analyst could always ask for the un-postprocessed answers as a simple workaround. (3) Choosing among 3 or more mechanisms requires solving a semidefinite program which is the same limitation shared by matrix mechanisms that represent query matrices explicitly (e.g., [25, 40, 69, 72–74]) and becomes problematic in high dimensions. Ideas from HDMM [51] could again help improve scalability.

4 RELATED WORK

There are roughly two types of tasks that differentially private algorithms perform: (1) query *selection*, which involves deciding *which* queries need answers, and (2) query *measurement* in which noisy answers to the queries are created.

Query selection is an important problem, with many important applications, such as synthetic data generation [4, 11, 30, 44–46, 52, 53, 75], as well as hyperparameter tuning [15, 43], feature selection [65], frequent itemset mining [6], exploring a privacy/accuracy tradeoff [42], data pre-processing [16], PAC learning [7], etc.

Query selection can be performed in a *data-independent* way, meaning that the queries are chosen in advance and no privacy loss budget is spent on the choice. In the case of linear queries, the techniques that plan the optimal set of queries in advance are generally called *matrix mechanisms* [25, 40, 51, 69, 72–74]. Matrix mechanisms *implicitly* take advantage of shared information between queries by setting up an optimization problem that finds a query strategy matrix that can answer the prespecified queries as accurately as possible under a privacy constraint. Our framework is complementary to the matrix mechanisms. For example, an analyst can use the matrix mechanism to construct $\mathcal{M}_1$. After observing the noisy answers, the analyst uses the matrix mechanism to plan $\mathcal{M}_2$ to answer a new batch of linear queries. By using our framework to decompose $\mathcal{M}_2$ into $\mathcal{M}_*$ and $\mathcal{M}'_2$, the analyst can run just $\mathcal{M}'_2$ (instead of $\mathcal{M}_2$) since she already has the information $\mathcal{M}_*$ would provide because it is also obtainable from the prior answers to $\mathcal{M}_1$. This saves privacy budget. There is also a relationship to consistency. If $\mathcal{M}_1$ and $\mathcal{M}_2$ are run independently, then the query matrix of $\mathcal{M}_*$ represents the linear queries on which $\mathcal{M}_1$ and $\mathcal{M}_2$ are inconsistent (i.e., the queries that they both could answer). In fact, running $\mathcal{M}_1$ and $\mathcal{M}_2$ independently is equivalent to running $\mathcal{M}'_1$, $\mathcal{M}'_2$, $\mathcal{M}_*$ plus a second independent run of $\mathcal{M}_*$. Enforcing consistency via postprocessing [32] is therefore equivalent to averaging the two copies of $\mathcal{M}_*$.

Query selection can also be performed in a data-dependent way by allocating some of the privacy loss budget to specially designed selection mechanisms. The sparse vector technique [23, 48], exponential mechanism [54] and various generalizations [5, 14, 18, 19, 42, 43, 58, 60, 65] are commonly used for this task, along with bespoke algorithms targeted at specific applications [2, 39, 71, 76].

Head-to-head comparisons between data-dependent and data-independent methods show that there is no clear winner [31] – in some situations, data-dependent selection provides the best choice of queries. In other situations, data-independent methods prevail. Thus it is important to keep expanding the available toolkit for query selection.

There is also a much smaller class of zero-waste differentially private selection algorithms [38, 41, 67, 68] whose purpose is to adaptively determine how much noise to add to a query, without wasting privacy loss budget. For example, suppose one is interested in the count of the number of people over 18. An analyst is prepared to spend up to ϵ_1 of her privacy loss budget (using pure ϵ-differential privacy) to get the answer. But, if the true number is large, she would prefer to use a smaller privacy loss budget value ϵ_2 (this results in a higher absolute error but is tolerable when the true answer is large because it still results in a small relative error). NoiseDown, which was introduced by Xiao et al. [67] (but had a bug in the algorithm) and later corrected by Koufogiannis et al. [38], is one technique to solve this problem. The analyst adds Laplace noise with privacy budget ϵ_2 to the true answer. Based on this answer, the analyst can either keep it or refine the noise so that the total privacy loss budget is ϵ_1 and so that the accuracy is the same as if she had used Laplace noise with budget ϵ_1 in the first place. Variations for this single-query noise refinement were also studied for randomized response [68] and Gaussian noise [41].

Our approach is also zero-waste, while being much more general. Instead of choosing between two versions of the same mechanism (the only difference being its privacy cost/amount of noise), our method allows the choice between two or more linear mechanisms that use Gaussian noise (and the mechanisms may also have different privacy costs). This adds another tool to the algorithmic toolbox for differential privacy. Although not necessarily a replacement for the exponential mechanism, we show empirically that our approach is useful in situations where it is difficult to specify the quality function that the exponential mechanism needs. One direction of future work is to combine the exponential mechanism with our technique – using the output of the common mechanism to fine-tune the construction of the exponential mechanism.

5 ALGORITHMS

We show how to compute the common mechanism in Section 5.1, residual mechanisms in Section 5.2, and recreate the original mechanism from the common and residual mechanism in Section 5.3.

5.1 Computing the Common Mechanism

The full procedure for computing the common mechanism is shown in Algorithm 1. We now explain how it is derived.

Algorithm 1: CommonMechanism($\mathcal{M}_1, \mathcal{M}_2$)

Input: Linear Gaussian Mechanisms $\mathcal{M}_1, \mathcal{M}_2$

1 $\mathbf{B}_1 \leftarrow$ Standardization($\mathcal{M}_1$) `// see Algorithm 2`

2 $\mathbf{B}_2 \leftarrow$ Standardization($\mathcal{M}_2$) `// see Algorithm 2`

3 $\mathbf{B}_* \leftarrow$ basis for rowspace($\mathbf{B}_1$) $\cap$ rowspace($\mathbf{B}_2$)

`// Calculate the covariance matrix` Σ

4 $\mathbf{A}_1 \leftarrow \mathbf{B}\mathbf{B}_1^\dagger$ `// † is the Moore-Penrose pseudoinverse`

5 $\mathbf{A}_2 \leftarrow \mathbf{B}\mathbf{B}_2^\dagger$

6 $\Sigma_* \leftarrow \frac{\mathbf{A}_1\mathbf{A}_1^T + \mathbf{A}_2\mathbf{A}_2^T}{2} + \frac{|\mathbf{A}_2\mathbf{A}_2^T - \mathbf{A}_1\mathbf{A}_1^T|}{2}$ `// where |·| replaces negative eigenvalues in a matrix with positive eigenvalues`

7 **Return** Mechanism $\mathcal{M}_*(\mathbf{x}) = \mathbf{B}_*\mathbf{x} + N(\mathbf{0}, \Sigma_*)$

Algorithm 2: Standardization($\mathcal{M}$)

Input: Linear Gaussian Mechanism $\mathcal{M}$ with query matrix $\mathbf{B}_{orig}$ and covariance matrix Σ_{orig}.

$\mathbf{X} \leftarrow \mathbf{B}_{orig}^T \Sigma_{orig}^{-1} \mathbf{B}_{orig}$ `// Privacy cost matrix`

Use eigenvalue decomposition to represent $\mathbf{X} = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \cdots + \lambda_k \mathbf{v}_k \mathbf{v}_k^T + \cdots + \lambda_d \mathbf{v}_d \mathbf{v}_d^T$, where $\lambda_1, \cdots, \lambda_k > 0$, $\lambda_{k+1} = \cdots = \lambda_d = 0$

$\mathbf{B} \leftarrow [\sqrt{\lambda_1}\mathbf{v}_1, \cdots, \sqrt{\lambda_k}\mathbf{v}_k]^T$ `// B is matrix sqrt of X`

Return $\mathbf{B}$

First, it is easier to work with mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$ when their corresponding matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ have linearly independent rows, and when the covariance matrices are the identity matrix. Thus we first perform a standardization step (Lines 1, 2) by calling Algorithm 2, which rewrites $\mathcal{M}_1$ (resp., $\mathcal{M}_2$) into an equivalent mechanism (Definition 3.6) whose query matrix has linearly independent rows and the covariance matrix is the identity.

Lemma 5.1. *Let $\mathcal{M}_{orig}$ be a linear Gaussian mechanism with query matrix $\mathbf{B}_{orig}$ and covariance matrix Σ_{orig}. Let $\mathcal{M}$ be a linear Gaussian mechanism with identity covariance and query matrix $\mathbf{B}$ obtained by running Algorithm 2 on $\mathcal{M}_{orig}$. Then $\mathcal{M}_{orig}$ and $\mathcal{M}$ are equivalent.*

5.1.1 An Optimization Problem for the Common Mechanism. Computing a maximal common mechanism $\mathcal{M}_*$ requires finding a query matrix $\mathbf{B}_*$ and covariance matrix Σ_*. The following theorem allows us to select $\mathbf{B}_*$ easily.

THEOREM 5.2. *Let $\mathcal{M}_1(\mathbf{x}) = \mathbf{B}_1\mathbf{x} + N(\mathbf{0}, \Sigma_1)$ and $\mathcal{M}_2(\mathbf{x}) = \mathbf{B}_2\mathbf{x} + N(\mathbf{0}, \Sigma_2)$ be linear Gaussian mechanisms.*

- *If $\mathcal{M}_c(\mathbf{x}) = \mathbf{B}_c\mathbf{x} + N(\mathbf{0}, \Sigma_c)$ is common to $\mathcal{M}_1$ and $\mathcal{M}_2$ then rowspace$(\mathbf{B}_c) \subseteq$ rowspace$(\mathbf{B}_1) \cap$ rowspace$(\mathbf{B}_2)$.*
- *If $\mathcal{M}_c$ is* maximally *common then rowspace$(\mathbf{B}_c) =$ rowspace$(\mathbf{B}_1) \cap$ rowspace$(\mathbf{B}_2)$*
- *The choice for basis of rowspace$(\mathbf{B}_1) \cap$ rowspace$(\mathbf{B}_2)$ does not matter. If $\mathcal{M}_c$ is maximally common and if $\mathbf{B}_* \neq \mathbf{B}_c$ is any matrix whose rows form a linearly independent basis, then there exists a common mechanism that is equivalent to $\mathcal{M}_c$ and has query matrix $\mathbf{B}_*$.*

Thus we set $\mathbf{B}_*$ to be a matrix whose rows form a basis for rowspace$(\mathbf{B}_1) \cap$rowspace$(\mathbf{B}_2)$ in Line 3 in Algorithm 1. This can be done in multiple ways, such as using the Zassenhaus algorithm [47] or eigendecompositions [34]. Then we use the following theorem to set up an optimization problem for finding a covariance matrix.

THEOREM 5.3. *Let $\mathcal{M}_1(\mathbf{x}) = \mathbf{B}_1\mathbf{x}+N(\mathbf{0}, \mathbf{I})$ and $\mathcal{M}_2(\mathbf{x}) = \mathbf{B}_2\mathbf{x}+N(\mathbf{0}, \mathbf{I})$ be linear Gaussian mechanisms that are standardized (i.e., produced by Algorithm 2). Let $\mathbf{B}_*$ be a matrix whose rows are linearly independent and spans rowspace$(\mathbf{B}_1) \cap$ rowspace$(\mathbf{B}_2)$. Then one can obtain a maximally common mechanism by using the Σ_* that optimizes the following problem (here $\dagger$ represents the Moore-Penrose Pseudoinverse operation):*

$$\Sigma_* \leftarrow \min_{\Sigma} \; trace(\Sigma) \; s.t. \; \Sigma \succeq \mathbf{B}_*\mathbf{B}_1^\dagger(\mathbf{B}_1^\dagger)^T\mathbf{B}_*{}^T \qquad (1)$$
$$\Sigma \succeq \mathbf{B}_*\mathbf{B}_2^\dagger(\mathbf{B}_2^\dagger)^T\mathbf{B}_*{}^T$$

To find the maximally common mechanism of 3 or more mechanisms, simply add an additional constraint for each mechanism in the optimization for Equation 1. However, when dealing with just two mechanisms, the optimization problem in Equation 1 has a symbolic solution that is used in Line 6 in Algorithm 1. This kind of matrix optimization was studied and solved by Stott [61–63]:

THEOREM 5.4. *[Stott [61–63]] The solution to the optimization problem in Equation 1 is $\Sigma_* = \frac{\mathbf{A}_1\mathbf{A}_1^T+\mathbf{A}_2\mathbf{A}_2^T}{2} + \frac{|\mathbf{A}_2\mathbf{A}_2^T-\mathbf{A}_1\mathbf{A}_1^T|}{2}$, where $\mathbf{A}_1 = \mathbf{B}_*\mathbf{B}_1^\dagger$, $\mathbf{A}_2 = \mathbf{B}_*\mathbf{B}_2^\dagger$, and $|\cdot|$ is the operator that replaces negative eigenvalues with positive eigenvalues (i.e., if the eigendecomposition of $\mathbf{V} = \mathbf{P}^T Diag(\lambda)\mathbf{P}$ then $|\mathbf{V}| = \mathbf{P}^T Diag(|\lambda|)\mathbf{P}$).*

5.2 Computing Residual Mechanisms

Given a mechanism $\mathcal{M}_1(\mathbf{x}) = \mathbf{B}_1\mathbf{x} + N(\mathbf{0}, \Sigma)$ and a maximally common mechanism $\mathcal{M}_*(\mathbf{x}) = \mathbf{B}_*\mathbf{x} + N(\mathbf{0}, \Sigma_*)$, computing the residual mechanism $\mathcal{M}'_1$ is greatly simplified by Lemma 3.11. One simply needs to find a $\mathbf{B}'_1$ and Σ'_1 that satisfies:

$$(\mathbf{B}'_1)^T(\Sigma'_1)^{-1}\mathbf{B}'_1 \;=\; \mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1 \;-\; \mathbf{B}_*{}^T\Sigma_*{}^{-1}\mathbf{B}_*.$$

This operation is performed by Algorithm 3.

Algorithm 3: ResidualMechanism($\mathcal{M}_i, \mathcal{M}_*$)

Input: Intended Mechanism $\mathcal{M}_i(\mathbf{x}) = \mathbf{B}_i\mathbf{x} + N(\mathbf{0}, \Sigma_i)$. Common mechanism $\mathcal{M}_*(\mathbf{x}) = \mathbf{B}_*\mathbf{x} + N(\mathbf{0}, \Sigma_*)$

$\mathbf{X} \leftarrow \mathbf{B}_i^T\Sigma_i^{-1}\mathbf{B}_i - \mathbf{B}_*{}^T\Sigma_*{}^{-1}\mathbf{B}_*$

Use eigenvalue decomposition to represent $\mathbf{X} = \lambda_1\mathbf{v}_1\mathbf{v}_1^T + \cdots + \lambda_k\mathbf{v}_k\mathbf{v}_k^T + \cdots + \lambda_d\mathbf{v}_d\mathbf{v}_d^T$, where $\lambda_1, \cdots, \lambda_k > 0, \lambda_{k+1} = \cdots = \lambda_d = 0$

$\mathbf{B}'_i \leftarrow [\sqrt{\lambda_1}\mathbf{v}_1, \cdots, \sqrt{\lambda_k}\mathbf{v}_k]^T$ `//` $\mathbf{B}'_i$ `is matrix sqrt of` $\mathbf{X}$

$\Sigma'_i \leftarrow \mathbf{I}_*$

Return Residual mechanism $\mathcal{M}'(\mathbf{x}) = \mathbf{B}'_i\mathbf{x} + N(\mathbf{0}, \Sigma'_i)$

5.3 Recreating the target mechanisms

Once one has obtained the output ω_* of the common mechanism $\mathcal{M}_*$, one would run the residual mechanism $\mathcal{M}'_i$ of the chosen mechanism $\mathcal{M}_i$ (i.e., $\mathcal{M}_1$ or $\mathcal{M}_2$) to obtain the output ω'_i.

The next step is to use ω_* and ω'_i to provide the same answer $\mathcal{M}_i$ would have produced. It is a postprocessing step and does not consume any privacy budget. It is a linear function of the vectors ω_* and ω'_i, shown in Algorithm 4 and justified by Theorem 5.5.

Algorithm 4: Recreate(ω_*, ω')

Input: ω_*: output of the common mechanism
ω'_i: output of a residual mechanism.
$\mathbf{B}_*, \Sigma_*$: query and cov. matrices of common mech.
$\mathbf{B}_i, \Sigma_i$: matrices for target mech. (e.g., $\mathcal{M}_1$ or $\mathcal{M}_2$).
$\mathbf{B}'_i.\Sigma'_i$: query and cov. matrices of residual mech.

$\Sigma_i^{1/2} \leftarrow$ symmetric matrix sqrt of Σ_i

$\mathbf{W} \leftarrow \mathbf{B}_i^T(\Sigma_i^{1/2})^{-1}$

$\mathbf{A}_* \leftarrow \Sigma_i^{1/2}\mathbf{W}^\dagger\mathbf{B}_*{}^T\Sigma_*{}^{-1}$

$\mathbf{A}'_i \leftarrow \Sigma_i^{1/2}\mathbf{W}^\dagger\mathbf{B}_i'^T(\Sigma'_i)^{-1}$

return $\mathbf{A}_*\omega_* + \mathbf{A}'_i\omega'_i$

THEOREM 5.5. *Let $\mathcal{M}_1(\mathbf{x}) = \mathbf{B}_1\mathbf{x} + N(\mathbf{0}, \Sigma_1)$ and $\mathcal{M}_2(\mathbf{x}) = \mathbf{B}_2\mathbf{x} + N(\mathbf{0}, \Sigma_2)$ be linear Gaussian mechanisms. Let $\mathcal{M}_*(\mathbf{x}) = \mathbf{B}_*\mathbf{x}+N(\mathbf{0}, \Sigma_*)$ be their maximally common mechanism and let $\mathcal{M}'_1(\mathbf{x}) = \mathbf{B}'_1\mathbf{x} + N(\mathbf{0}, \Sigma'_1)$ be the residual mechanism for $\mathcal{M}_1$. Define:*

- *$\Sigma_1^{1/2}$ to be the symmetric matrix square root of Σ_1,*
- *$\Sigma_1^{-1/2}$ to be the inverse of $\Sigma_1^{1/2}$,*
- *$\mathbf{A}_* = \Sigma_1^{1/2}(\mathbf{B}_1^T\Sigma_1^{-1/2})^\dagger\mathbf{B}_*{}^T\Sigma_*{}^{-1}$, where $\dagger$ is the Moore-Penrose pseudo-inverse,*
- *$\mathbf{A}'_1 = \Sigma_1^{1/2}(\mathbf{B}_1^T\Sigma_1^{-1/2})^\dagger\mathbf{B}_1'^T(\Sigma'_1)^{-1}$.*

Then $\mathcal{M}_1(\mathbf{x}) = \mathbf{A}_*\mathcal{M}_*(\mathbf{x}) + \mathbf{A}'_1\mathcal{M}'_1(\mathbf{x}) + N(\mathbf{0}, \Sigma_1 - \mathbf{A}_*\Sigma_*\mathbf{A}_*^T - \mathbf{A}'_1\Sigma'_1\mathbf{A}_1'^T)$ and $\mathcal{M}_1(\mathbf{x})$ is equivalent to $\mathbf{A}_*\mathcal{M}_*(\mathbf{x}) + \mathbf{A}'_1\mathcal{M}'_1(\mathbf{x})$.

6 MAKING DECISIONS BASED ON THE COMMON MECHANISM

We next consider how one could use the output ω_* of the common mechanism to decide whether to run the residual mechanism $\mathcal{M}'_1$ in order to do the analysis supported by $\mathcal{M}_1$, or whether to run $\mathcal{M}'_2$ instead. In general, this would be user/application dependent, but we list some suggestions here. We first consider nested analyses, such as 1-way vs. 2-way marginals, as this is a common special case. Then we consider the general case where $\mathcal{M}_1$ and $\mathcal{M}_2$ are arbitrary linear Gaussian mechanisms.

6.1 Nested Analyses

A query set Q_1 is nested in a query set Q_2 if the answers to Q_1 can be obtained from Q_2. An example is one-way marginals (Q_1) and two-way marginals Q_2. This situation can arise when the analyst has a primary analyses, such as studying the race distribution of a region, and the age distribution of a region (e.g., these are the one-way marginals Q_1). If possible, however, the analyst would also like to dig deeper with a secondary analysis such as studying the interactions between age and race in the population (this is the 2-way marginal Q_2). When an analyst is only allowed to obtain marginals that are noisy, an important decision must be made.

- If the analyst requests noisy one-way marginals via a mechanism $\mathcal{M}_1(\mathbf{x}) = \mathbf{B}_1\mathbf{x} + N(\mathbf{0}, \Sigma_1)$ then the analyst gets the most accurate noisy information needed for their primary analysis, but no secondary analysis can be performed.
- If the analyst requests noisy two-way marginals via a mechanism $\mathcal{M}_2(\mathbf{x}) = \mathbf{B}_2\mathbf{x} + N(\mathbf{0}, \Sigma_2)$. This would allow the secondary analysis to proceed. Since noisy two-way marginals can be used to compute noisier one-way marginals (these one-way marginals will be noisier than in the first option), then the primary analysis can also be performed if there was enough signal-to-noise ratio.

Here we define the signal-to-noise ratio (SNR) as the true (unknown) count c of a cell divided by the standard deviation σ of the privacy noise in the cell (hence SNR of a cell is $\frac{c}{\sigma}$). Since Gaussian noise is within 3σ of its mean 99.7% of the time, the relative error of the noisy cell count is within $\frac{3\sigma}{c}$ for 99.7% of the time, and hence SNR can be thought of as the reciprocal of relative error. Thus, if noisy two-way marginals can be used to compute one-way marginals with enough signal-to-noise ratio to perform the primary analysis, then $\mathcal{M}_2$ is preferable since $\mathcal{M}_2$ also has additional uses. Otherwise, $\mathcal{M}_1$ would be preferable. The goal is for the analyst to use the output of the common mechanism to estimate what the SNR of each marginal cell would be if $\mathcal{M}_1$ or $\mathcal{M}_2$ were used.

We say that the analysis supported by query matrix $\mathbf{B}_1$ (e.g., one-way marginals) is nested in the analysis supported by $\mathbf{B}_2$ (e.g., two-way marginals) if there exists a matrix $\mathbf{A}$ such that $\mathbf{B}_1 = \mathbf{A}\mathbf{B}_2$. In this case, $\mathbf{A}\mathcal{M}_2(\mathbf{x})$ (i.e, multiplying the output of $\mathcal{M}_2$ by $\mathbf{A}$) is a noisier version of $\mathcal{M}_1$. We say that $\mathbf{B}_1$ is the primary analysis and $\mathbf{B}_2$ is the secondary analysis.

Now, the common mechanism $\mathcal{M}_*$ of $\mathcal{M}_1$ and $\mathcal{M}_2$ can provide a noisy answer to any query in the intersection of the row spaces of $\mathbf{B}_1$ and $\mathbf{B}_2$ (by Theorem 5.2) and so there exists a matrix $\mathbf{A}_*$ such that $E[\mathcal{M}_1(\mathcal{D})] = E[\mathbf{A}_*\mathcal{M}_*(\mathcal{D})]$, which means that $\mathbf{A}_*\mathcal{M}_*$ is also a noisier version of $\mathcal{M}_1$, and the variance of the i^{th} query is the i^{th} diagonal element of $\mathbf{A}_*\Sigma_*\mathbf{A}_*^T$, denoted by $(\mathbf{A}_*\Sigma_*\mathbf{A}_*^T)[i,i]$. Thus the output ω_* of $\mathcal{M}_*$ can be used to estimate the signal to noise ratio of using mechanism $\mathcal{M}_2$ to do the nested primary analysis as follows (all of this is a postprocessing of ω_*):

- The quantity $L_i = (\mathbf{A}_*\omega_*)[i] - 3\sqrt{(\mathbf{A}_*\Sigma_*\mathbf{A}_*^T)[i,i]}$ is a 3 sigma lower confidence interval for the true value of the i^{th} query in $\mathbf{B}_1\mathbf{x}$. The quantity $U_i = (\mathbf{A}_*\omega_*)[i] + 3\sqrt{(\mathbf{A}_*\Sigma_*\mathbf{A}_*^T)[i,i]}$ is the corresponding upper confidence interval.
- The quantity $SNR\_Lower_i^{(1)} = \frac{L_i}{\sqrt{\Sigma_1[i,i]}}$ is a lower bound on the expected signal-to-noise ratio of using $\mathcal{M}_1$ to get a noisy answer to the i^{th} query in $\mathbf{B}_1\mathbf{x}$. Similarly, $SNR\_Upper_i^{(1)} = \frac{U_i}{\sqrt{\Sigma_1[i,i]}}$ would be an upper bound on the signal-to-noise ratio.
- The quantity $SNR\_Lower_i^{(2)} = \frac{L_i}{\sqrt{(\mathbf{A}\Sigma_2\mathbf{A}^T)[i,i]}}$ (resp., $SNR\_Upper_i^{(2)} = \frac{U_i}{\sqrt{(\mathbf{A}\Sigma_2\mathbf{A}^T)[i,i]}}$) is a lower (resp., upper) bound on the expected signal-to-noise ratio of using $\mathcal{M}_2$ to get a noisy answer to the i^{th} query in $\mathbf{B}_1\mathbf{x}$ (the nested analysis).

These quantities can be used in a variety of ways. For example, the user may want at least x% of the queries of $\mathbf{B}_1$ to have SNR above y. In this case, if x% of the $SNR\_Lower_i^{(2)}$ values are $\geq y$, then $\mathcal{M}_2$ is good enough for the primary analysis and also provides an opportunity to perform the secondary analysis. Thus, the user should decide to run the residual mechanism $\mathcal{M}_2'$ to get the answer to $\mathcal{M}_2$.

Other possibilities also exist. If too many of the $SNR\_Upper_i^{(1)}$ values are below the desired signal-to-noise ratio y, then even $\mathcal{M}_1$ is not accurate enough for the primary analysis and the analyst can stop here, without using any further privacy budget beyond what $\mathcal{M}_*$ had cost. On the other hand, if the SNR bound y is between $SNR\_Lower_i^{(2)}$ and $SNR\_Upper_i^{(2)}$ for many i, then the analyst also has the option of using more privacy budget to help make the decision between $\mathcal{M}_1$ and $\mathcal{M}_2$, or may opt for $\mathcal{M}_1$ just to be safe.

6.2 The General Case

In the general case, there may not be a nice structure (e.g., nesting) to take advantage of. In this case an analyst, who needs to choose between $\mathcal{M}_1$ and $\mathcal{M}_2$, could run the common mechanism and, based on its output, would determine how much uncertainty she has about the queries represented by the query matrix $\mathbf{B}_1$ vs. uncertainty about the queries of $\mathbf{B}_2$. If, for example, $\mathbf{B}_1$ has the most remaining uncertainty, this means that the analyst could gain the most new information by running $\mathcal{M}_1$. We briefly sketch a Bayesian and a Frequentist idea for measuring this uncertainty.

If an analyst is able to compute or approximate the posterior distribution $P(\mathbf{x} \mid \mathcal{M}_*(\mathbf{x}) = \omega_*)$, then the analyst can use it to compute the variance of each query in $\mathbf{B}_1\mathbf{x}$ and in $\mathbf{B}_2\mathbf{x}$. This is their measure of uncertainty. They can compare this to the variances that $\mathcal{M}_1$ and $\mathcal{M}_2$ could provide, namely the diagonals of Σ_1 and Σ_2. The analyst can then complete the choice of $\mathcal{M}_1$ vs. $\mathcal{M}_2$ based on whichever one represents the largest reduction in variance.

In the Frequentist view, if query $\mathbf{B}_*$ of the common mechanism is answered with Gaussian noise having covariance matrix Σ_* to get an output ω_*, then one can establish a confidence interval that should contain the true answer $\mathbf{B}_*\mathbf{x}_{true}$ to the common mechanism (where $\mathbf{x}_{true}$ is the unknown true dataset). For a given significance level α (e.g., 0.95) one can find the cutoff z such a chi-squared random variable with m degrees of freedom (m is the number of rows in $\mathbf{B}_*$) exceeds z with probability $1-\alpha$. This means that with probability α, we must have $(\mathbf{B}_*\mathbf{x}_{true} - \omega_*)^T\Sigma_*^{-1}(\mathbf{B}_*\mathbf{x}_{true} - \omega_*) \leq z$. One can then use methods such as hit-and-run sampling [3, 59] to sample datasets consistent with this information (i.e., sample datasets $\widehat{\mathbf{x}}$ for which $(\mathbf{B}_*\widehat{\mathbf{x}}-\omega_*)^T\Sigma_*{}^{-1}(\mathbf{B}_*\widehat{\mathbf{x}}-\omega_*) \leq z$); this is just a postprocessing of ω_* and so has no effect on privacy. Given m such sampled datasets $\widehat{\mathbf{x}}_1,\dots,\widehat{\mathbf{x}}_m$, one can again estimate how much variability there is for each query in $\mathbf{B}_1\widehat{\mathbf{x}}_i$ and $\mathbf{B}_2\widehat{\mathbf{x}}_i$ and, as before, compare that to the variances that $\mathcal{M}_1$ and $\mathcal{M}_2$ can provide.

7 EXPERIMENTS

We conduct experiments to examine how well the common mechanism can guide the analyst when choosing between nested analyses, as described in Section 6.1. We use three real datasets and several applications, one of which (Section 7.5) is a comparison to an algorithm that the Census Bureau will use as part of the Detailed Demographic and Housing Characteristics (DHC) data release [29].

7.1 Datasets

We use three datasets: **HispRace**, **AgeGender**, and **Brazil** [13]. The **HispRace** dataset is extracted from the 2010 Summary File 1 (SF1) tabulations P4 and P5 [9]. For each of the 6,257,947 occupied census blocks in the dataset, a record has 7 binary race and ethnicity attributes for a domain size of $2^7 = 128$ for each block. The binary attributes are *Hispanic or Latino*, *White*, *Black or African American*, *American Indian and Alaska Native*, *Asian*, *Native Hawaiian and Other Pacific Islander*, *Some Other Race*. The **AgeGender** dataset is extracted from the SF1 tabulation PCT12. For each of the 73,426 census tracts in the dataset, a record has a binary gender variable and 103 possible age values. The 2010 **Brazil** dataset is obtained from IPUMS [13] and consists of 20,635,472 census records from Brazil. We extract the following attributes: *state* (25 possible values), *occupation* (437 possible values), *age* (101 possible values), and *gender* (2 possible values).

7.2 Evaluation Measures

We consider the nested analysis setting (Section 6.1) in which the primary analysis is represented by $\mathcal{M}_1(\mathbf{x}) = \mathbf{B}_1\mathbf{x} + N(\mathbf{0},\Sigma_1)$. The secondary analysis is represented by $\mathcal{M}_2(\mathbf{x}) = \mathbf{B}_2\mathbf{x} + N(\mathbf{0},\Sigma_2)$. Thus, the primary analysis can also be done, but less accurately, with the output of $\mathcal{M}_2$. However, $\mathcal{M}_2$ supports additional analyses that $\mathcal{M}_1$ cannot.

The output ω_* of the common mechanism is used to estimate whether the output of $\mathcal{M}_2$ is accurate enough to perform the primary analysis – if at least x% of the marginal cells are believed to have an SNR at least y and then the residual mechanism $\mathcal{M}_2'$ is run to recreate the output of $\mathcal{M}_2$ without wasting privacy loss budget (otherwise the residual mechanism $\mathcal{M}_1'$ is used). An **alternate approach**, that doesn't use the common mechanism, is to reserve some privacy budget to estimate the SNR (by getting a coarse noisy answer to $\mathbf{B}_1\mathbf{x}$ and then using that noisy answer to estimate SNR as in Section 6.1) and then run either $\mathcal{M}_1$ or $\mathcal{M}_2$. The alternate approach uses the *optimal* Gaussian mechanism [69] that matches the variance of the common mechanism (for an apples-to-apples comparison) while minimizing privacy cost. We measure the following quantities:

- $\underline{\rho}$: the concentrated differential privacy parameter (zcdp) [8] that represents the total privacy loss budget.
- $\underline{\%\mathcal{M}_1, \%\mathcal{M}_2}$: based on the ground truth (computed from the true count divided by noise std), the percent of the time that $\mathcal{M}_1$ (resp., $\mathcal{M}_2$) should have been chosen. A good decision-making strategy should outperform the maximum of these two. In Section 7.5, where we must choose between 4 mechanisms, we report $\%\mathcal{M}_1$, $\%\mathcal{M}_2$, $\%\mathcal{M}_3$, $\%\mathcal{M}_4$.
- Acc: accuracy of the selection based on the common mechanism. This is the percentage of time that the correct mechanism has been chosen without wasting any privacy budget.
- %PLB Saved: this is how much privacy loss budget the optimal alternate approach needs to allocate to the estimation of SNR in order to match the estimation quality of the common mechanism. This is how much privacy loss budget is saved by using the common mechanism methodology instead of the traditional alternate approach. We express this as a percentage of the total privacy loss budget. Note that %PLB Saved depends only on the query matrices of $\mathcal{M}_1$ and $\mathcal{M}_2$, so there is just one %PLB Saved value per table.

7.3 Marginals on HispRace

For the *HispRace* dataset, we consider two settings. In the first case, $\mathcal{M}_1$ adds independent noise to 1-way marginals while $\mathcal{M}_2$ adds independent noise to 2-way marginals. For each census block, the analysts must choose between $\mathcal{M}_1$ and $\mathcal{M}_2$. When $\mathcal{M}_2$ is accurate enough to allow an analyst to derive 1-way marginals that exceed an SNR bound, then $\mathcal{M}_2$ is preferred. In the second set of experiments, $\mathcal{M}_1$ adds independent noise to 1-way marginals while $\mathcal{M}_2$ adds independent noise to the identity query (i.e., each cell of the data vector $\mathbf{x}$). Table 2 shows the results as the privacy loss is varied. The common mechanism allows the analyst to correctly choose the right analysis with high accuracy. In the case of 1-way vs. 2-way marginals, the privacy loss budget saved (compared to methods that allocate some privacy loss budget for SNR estimation) is significant (75%), while for 1-way vs. identity, the savings are more moderate (6.25%). Tables 3 and 4 show the accuracy of selection based on the common mechanism as the SNR parameters x (desired fraction of cells with high signal) and y (desired minimum signal-to-noise ratio) are varied. Overall, when one wishes to run either $\mathcal{M}_1$ or $\mathcal{M}_2$, then the common mechanism represents information that comes for free because both mechanisms provide it. This information is accurate enough for choosing between the mechanisms and does not waste privacy budget. By avoiding the traditional approach of reserving privacy loss budget for making a decision, we replace the budget allocation tuning parameter with an analyst-provided SNR utility specification.

7.4 Marginals on the Brazil Dataset

In the **Brazil** dataset, we consider the setting where, for each combination of *state* and *occupation*, the analyst needs to choose whether to run $\mathcal{M}_1$ to produce noisy 1-way marginals or $\mathcal{M}_2$ to produce noisy 2-way marginals by adding independent noise.

The accuracy of making the choice, for each state/occupation combination is shown in Table 5. Again, the common mechanism provides enough information for choosing between the two mechanisms (i.e. choosing between which residual mechanism to run).

1-way vs. 2-way Marginals			
ρ	$\%\mathcal{M}_1$	$\%\mathcal{M}_2$	Acc
2	45.54	54.46	98.64
1	54.35	45.65	97.98
1/2	65.47	34.53	98.37
1/8	84.12	15.81	98.84
1/32	94.41	5.59	99.56
%**PLB Saved**: 75%			

1-way vs. Identity			
ρ	$\%\mathcal{M}_1$	$\%\mathcal{M}_2$	Acc
2	50.20	49.80	95.00
1	61.46	38.54	95.37
1/2	71.32	28.68	95.40
1/8	88.27	11.73	97.53
1/32	95.98	4.02	99.08
%**PLB Saved**: 6.25%			

Table 2: Experiments on HispRace dataset as zCDP privacy budget ρ varies. The SNR parameters are $(x, y) = (0.5, 5)$.

1-way vs. 2-way Marginals			
y	$\%\mathcal{M}_1$	$\%\mathcal{M}_2$	Acc
2	58.08	41.92	95.56
3	70.57	29.43	97.22
4	78.75	21.25	98.20
5	84.19	15.81	98.83
6	87.82	12.18	99.22
8	92.10	7.90	99.61
%**PLB Saved**: 75%			

1-way vs. Identity			
y	$\%\mathcal{M}_1$	$\%\mathcal{M}_2$	Acc
2	64.52	35.48	88.83
3	76.58	23.42	93.56
4	83.85	16.15	96.08
5	88.27	11.73	97.54
6	91.07	8.93	98.35
8	94.27	5.73	99.17
%**PLB Saved**: 6.25%			

Table 3: Experiments on HispRace dataset for zCDP parameter $\rho = 1/8$ and SNR parameters $(0.5, y)$ as y varies.

1-way vs. 2-way Marginals			
x	$\%\mathcal{M}_1$	$\%\mathcal{M}_2$	Acc
0.2	79.61	20.39	99.07
0.3	81.25	18.75	98.97
0.4	82.96	17.04	98.95
0.5	84.19	15.81	98.83
0.6	97.95	2.05	99.85
%**PLB Saved**: 75%			

1-way vs. Identity			
x	$\%\mathcal{M}_1$	$\%\mathcal{M}_2$	Acc
0.2	84.42	15.58	97.05
0.3	85.86	14.14	97.44
0.4	87.30	12.70	97.55
0.5	88.27	11.73	97.52
0.6	98.50	1.50	99.65
%**PLB Saved**: 6.25%			

Table 4: Experiments on HispRace dataset for zCDP parameter $\rho = 1/8$ and SNR parameters $(x, 5)$ as x varies.

1-way vs. 2-way Marginals			
ρ	$\%\mathcal{M}_1$	$\%\mathcal{M}_2$	Acc
2	71.47	28.53	96.95
1	71.47	28.53	97.86
1/2	78.65	21.35	98.29
1/8	85.30	14.70	98.94
1/32	90.66	9.34	99.20
%**PLB Saved**: 50.5%			

Table 5: Experiments on Brazil dataset as zCDP privacy budget ρ varies. The SNR parameters are $(x, y) = (0.3, 3)$.

1-way vs. 2-way Marginals			
x	$\%\mathcal{M}_1$	$\%\mathcal{M}_2$	Acc
0.2	58.56	41.44	96.27
0.3	71.47	38.53	96.95
0.4	81.14	18.86	97.55
0.5	88.78	11.22	98.46
0.6	94.58	5.42	99.18
%**PLB Saved**: 50.5%			

1-way vs. 2-way Marginals			
y	$\%\mathcal{M}_1$	$\%\mathcal{M}_2$	Acc
3	71.47	28.53	96.95
4	71.47	28.53	98.41
5	75.78	24.22	99.03
6	78.65	21.35	99.00
8	80.94	19.06	99.40
%**PLB Saved**: 50.5%			

Table 6: Experiment on Brazil dataset for zCDP parameter $\rho = 2$. Left: SNR parameters $(x, 3)$ as x varies. Right: SNR parameters $(0.3, y)$ as y varies.

It also provides significant privacy budget savings (50.5%) compared to the approach that first allocates privacy loss budget to estimating SNR just as accurately, before making a choice, with the side benefit being that there is no privacy budget allocation tuning parameter necessary when using the common mechanism. The accuracy of selection based on the common mechanism, as we vary the SNR parameters, is shown in Table 6 and again shows fairly good accuracy.

7.5 Census Age/Gender Application

Our next set of experiments is a case study for a data product that will be released as part of the 2020 Decennial Census Detailed Demographics and Housing Characteristics [29].

7.5.1 Problem Description. This data product has age-by-gender histograms for different sub-populations, such as for an ethnic group in a given region. Since the sub-population might be sparse, one of 4 pre-defined age bucketization schemes will be used [29]:

- **Total**. This consists of one age bucket: $[0, 103)$. In this case, the age-by-gender histogram is simply the number of females and the number of males.
- **Age4**. This consists of the following four buckets: $[0, 18)$; $[18, 45)$; $[45, 65)$; $[65, 103]$.
- **Age9**. This consists of the following nine buckets: $[0, 5)$; $[5, 18)$; $[18, 25)$; $[25, 35)$; $[35, 45)$; $[45, 55)$; $[55, 65)$; $[65, 75)$; $[75, 103]$.
- **Age23**. This consists of the following 23 buckets: $[0, 5)$; $[510)$; $[10, 15)$; $[15, 18)$; $[18, 20)$; $[20, 21)$; $[21, 22)$; $[22, 25)$; $[25, 30)$; $[30, 35)$; $[35, 40)$; $[40, 45)$; $[45, 50)$; $[50, 55)$; $[55, 60)$; $[60, 62)$; $[62, 65)$; $[65, 67)$; $[67, 70)$; $[70, 75)$; $[75, 80)$; $[80, 85)$; $[85, 103]$.

We note that these represent nested analyses, as **Age23** is a refinement of **Age9**, which is a refinement of **Age4**, which is a refinement of **Total**. The idea is that the smaller the sub-population is, the coarser the age buckets should be in order for the noise not to overwhelm the actual counts.

7.5.2 The Census algorithm. One of the algorithms we compare against is the DHC algorithm, which is the one that will actually be used for the problem [29] by the Census Bureau. This DHC algorithm will use a fraction γ of the privacy loss budget to estimate the total size of the sub-population. There are also three threshold parameters $\theta_1 < \theta_2 < \theta_3$. If the noisy sub-population count is $< \theta_1$, the remaining privacy budget will be used to produce the gender by age histogram using the **total** bucketization. If the noisy sub-population count is in the range $[\theta_1, \theta_2)$, then the **Age4** bucketization will be used. If it is in the range of $[\theta_2, \theta_3)$, then **Age9** will be used. Otherwise, **Age23** will be used. Thus the algorithm has 4 parameters that must be carefully tuned: $\gamma, \theta_1, \theta_2, \theta_3$. The decision on parameter values has not been made public (possibly a difficult decision) but the rest of the algorithm is public.

7.5.3 The Common Mechanism Approach. In the problem setting, there are four algorithms to consider: $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4$, which produce noisy gender by age histograms with the bucketizations **total**, **Age4**, **Age9**, **Age23**, respectively, by adding independent Gaussian noise. We define 3 common mechanisms: CM_{1234} is common to $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4$ (i.e., it is the maximal mechanism exactly answerable from all of them). It is obtained by numerically solving the optimization in Equation 1 with a constraint added for each mechanism. CM_{234} is common to $\mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4$, and CM_{34} is common to $\mathcal{M}_3$ and $\mathcal{M}_4$. Now, CM_{1234} is exactly answerable from CM_{234}, which is exactly answerable from CM_{34}. Therefore we can perform the following procedure that wastes no privacy loss budget:

- Get output ω_{1234} of $\mathcal{M}_{1234}$ and decide whether the **total** bucketization should be used based on SNR. If yes, run the residual mechanism RM_1 such that (CM_{1234}, RM_1) is equivalent to $\mathcal{M}_1$.
- If no, run the residual mechanism RM_{234} such that $(\mathcal{M}_{1234}, RM_{234})$ is equivalent to CM_{234}. Use that to decide whether to use **Age4**. If yes, run the residual mechanism RM_2 such that (CM_{234}, RM_2) is equivalent to $\mathcal{M}_2$.
- If no, run the residual mechanism RM_{34} such that $(\mathcal{M}_{234}, RM_{34})$ is equivalent to CM_{34}. Use that to decide whether to use **Age9**. If yes, run the residual mechanism RM_3 such that (CM_{34}, RM_3) is equivalent to $\mathcal{M}_3$.
- If no, run the residual mechanism RM_4 such that (CM_{34}, RM_4) is equivalent to $\mathcal{M}_4$.

Thus, at the end, one gets something that is equivalent to either $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$ or $\mathcal{M}_4$ without wasting any privacy budget.

7.5.4 The Alternative Approach. We also consider a third approach that mirrors the previous experiments. Instead of making decisions based on the common mechanism, at each step, some privacy loss budget is allocated to obtain the best linear Gaussian mechanism to estimate the SNR of the histogram being considered. This is also an alternative to making decisions based on noisy population counts.

7.5.5 Results. Since the input data that the DHC algorithm will use is not public, we use the **AgeGender** dataset described in Section 7.1 to produce an Age by Gender histogram at each census tract using the three algorithms described.

Since the DHC algorithm requires additional parameters, we give it a strong non-private advantage: the $\theta_1, \theta_2, \theta_3$ are learned using a non-private logistic regression model, and the DHC algorithm is given the exact sub-population count (i.e., a noiseless threshold).

We set the desired SNR parameters to be $x = 0.5$ and $y = 20$. In a histogram cell, a ratio of 20 between a count c and the privacy noise standard deviation σ means that 95% of the time, the relative error of the noisy cell count is at most $2\sigma/c = 2/20 = 10\%$.

In Figure 1, we compare the accuracy of the DHC algorithm to the common mechanism in choosing the right age bucketization to use. Even with the advantages we gave it (e.g., tuned parameters and noise-free thresholds), it is still outperformed by the common mechanism, which avoids all those tuning parameters. This shows that the noisy information provided by the common mechanism

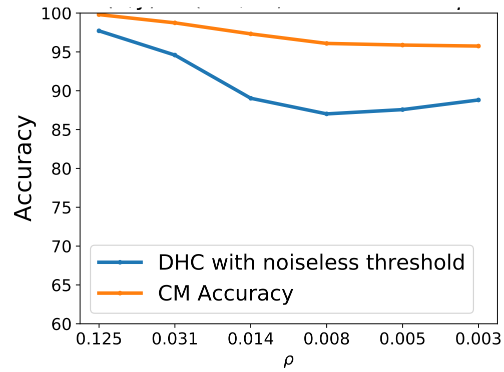

Figure 1: Common mechanism vs. tuned DHC algorithm as privacy budget ρ varies.

ρ	$\%\mathcal{M}_1$	$\%\mathcal{M}_2$	$\%\mathcal{M}_3$	$\%\mathcal{M}_4$	Acc	**% PLB saved**
1/8	0.34	0.68	1.64	97.34	99.80	49.67
1/32	0.39	1.31	12.61	85.69	98.74	49.59
1/72	0.45	3.46	31.51	64.58	97.32	48.92
1/128	0.50	8.08	49.95	41.47	96.09	47.74
1/200	0.56	14.50	61.43	23.51	95.88	46.11
1/288	0.63	22.55	64.85	11.97	95.75	46.06

Table 7: Privacy budget savings vs. the alternate approach.

for free is more informative than the population thresholds, even when the population thresholds are completely accurate.

We next compare the common mechanism to the alternate approach (Section 7.5.4) that, for an apples-to-apples comparison, allocates privacy budget to the optimal linear Gaussian mechanism that has equal variance to the common mechanism, but with smallest possible privacy cost. Table 7 shows the accuracy of choosing the correct age bucketization and the fraction of PLB saved by using the common mechanism (nearly 50%). Since the choice among the four bucketizations is done sequentially, the sooner one is chosen, the better for the alternate approach since it no longer has to allocate PLB for future choices. For this reason, the percentage of PLB saved varies with the experimental setting.

8 CONCLUSIONS

In this paper, we formalized the problem and provided algorithms for the computation of the common mechanism $\mathcal{M}_*$ for two linear Gaussian mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$. The common mechanism represents information that is provided by both mechanisms, while the residual mechanisms $\mathcal{M}_1'$ and $\mathcal{M}_2'$ reflect the remaining information in $\mathcal{M}_1$ and $\mathcal{M}_2$ after the information from $\mathcal{M}_*$ has been removed from them. We presented an application where an analyst can decide whether to get answers of $\mathcal{M}_1$ or $\mathcal{M}_2$ using the help of the common and residual mechanisms. This represents another tool that can be used for differentially private algorithm design.

Acknowledgments. This work was supported by NSF Awards CNS-1702760 and CNS-1931686.

REFERENCES

[1] John M. Abowd, Robert Ashmead, Ryan Cumings-Menon, Simson Garfinkel, Micah Heineck, Christine Heiss, Robert Johns, Daniel Kifer, Philip Leclerc, Ashwin Machanavajjhala, Brett Moran, William Sexton, Matthew Spence, and Pavel Zhuravlev. forthcoming. Preprint https://www.census.gov/library/working-papers/2022/adrm/CED-WP-2022-002.html. The 2020 Census Disclosure Avoidance System TopDown Algorithm. *Harvard Data Science Review* (forthcoming. Preprint https://www.census.gov/library/working-papers/2022/adrm/CED-WP-2022-002.html).

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

[3] Hans C. Andersen and Persi Diaconis. 2007. Hit and run as a unifying device. *Journal de la société française de statistique* 148, 4 (2007), 5–28. http://eudml.org/doc/93471

[4] Sergul Aydore, William Brown, Michael Kearns, Krishnaram Kenthapadi, Luca Melis, Aaron Roth, and Ankit A Siva. 2021. Differentially private query release through adaptive projection. In *International Conference on Machine Learning*. PMLR, 457–467.

[5] Amos Beimel, Kobbi Nissim, and Uri Stemmer. 2016. Private Learning and Sanitization: Pure vs. Approximate Differential Privacy. *Theory of Computing* 12, 1 (2016), 1–61.

[6] Raghav Bhaskar, Srivatsan Laxman, Adam Smith, and Abhradeep Thakurta. 2010. Discovering Frequent Patterns in Sensitive Data. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[7] Mark Bun, Gautam Kamath, Thomas Steinke, and Steven Z Wu. 2019. Private hypothesis selection. *Advances in Neural Information Processing Systems* 32 (2019).

[8] Mark Bun and Thomas Steinke. 2016. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. In *Proceedings, Part I, of the 14th International Conference on Theory of Cryptography - Volume 9985*.

[9] U.S. Census Bureau. [n.d.]. Decennial Census: 2010 Summary Files. https://www.census.gov/mp/www/cat/decennial_census_2010/.

[10] U. S. Census Bureau. [n.d.]. On The Map: Longitudinal Employer-Household Dynamics. https://lehd.ces.census.gov/applications/help/onthemap.html#!confidentiality_protection.

[11] Kuntai Cai, Xiaoyu Lei, Jianxin Wei, and Xiaokui Xiao. 2021. Data synthesis via differentially private markov random fields. *Proceedings of the VLDB Endowment* 14, 11 (2021), 2190–2202.

[12] Clement L Canonne, Gautam Kamath, and Thomas Steinke. 2020. The Discrete Gaussian for Differential Privacy. In *NeurIPS*.

[13] Minnesota Population Center. 2020. Integrated Public Use Microdata Series, International: Version 7.3 [dataset]. Minneapolis, MN: IPUMS, 2020. https://doi.org/10.18128/D020.V7.3

[14] Kamalika Chaudhuri, Daniel Hsu, and Shuang Song. 2014. The Large Margin Mechanism for Differentially Private Maximization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*.

[15] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research* 12, Mar (2011), 1069–1109.

[16] Yan Chen, Ashwin Machanavajjhala, Jerome P. Reiter, and Andrés F. Barrientos. 2016. Differentially Private Regression Diagnostics. In *IEEE 16th International Conference on Data Mining (ICDM)*.

[17] 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., USA, 3574–3583. http://dl.acm.org/citation.cfm?id=3294996.3295115

[18] Zeyu Ding, Yuxin Wang, Yingtai Xiao, Guanhong Wang, Danfeng Zhang, and Daniel Kifer. 2022. Free gap estimates from the exponential mechanism, sparse vector, noisy max and related algorithms. *The VLDB Journal* (2022).

[19] Zeyu Ding, Yuxin Wang, Danfeng Zhang, and Daniel Kifer. 2019. Free Gap Information from the Differentially Private Sparse Vector and Noisy Max Mechanisms. *Proc. VLDB Endow.* 13, 3 (nov 2019), 293–306. https://doi.org/10.14778/3368289.3368295

[20] Jinshuo Dong, Aaron Roth, and Weijie J. Su. 2022. Gaussian differential privacy. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 84, 1 (2022), 3–37. https://doi.org/10.1111/rssb.12454 arXiv:https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/rssb.12454

[21] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our Data, Ourselves: Privacy via Distributed Noise Generation. In *EUROCRYPT*. 486–503.

[22] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis.. In *TCC*.

[23] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3–4 (2014), 211–407. https://doi.org/10.1561/0400000042

[24] Hamid Ebadi, David Sands, and Gerardo Schneider. 2015. Differential Privacy: Now It's Getting Personal. In *POPL*.

[25] Alexander Edmonds, Aleksandar Nikolov, and Jonathan Ullman. 2020. The power of factorization mechanisms in local and central differential privacy. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 425–438.

[26] 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)*. ACM, New York, NY, USA, 1054–1067.

[27] 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* (Amsterdam, Netherlands) *(SIGMOD '19)*. Association for Computing Machinery, New York, NY, USA, 177–194. https://doi.org/10.1145/3299869.3300092

[28] Google. [n.d.]. https://github.com/tensorflow/privacy.

[29] Samuel Haney, William Sexton, Ashwin Machanavajjhala, Michael Hay, and Gerome Miklau. 2021. Differentially Private Algorithms for 2020 Census Detailed DHC Race \& Ethnicity. *CoRR* abs/2107.10659 (2021). arXiv:2107.10659 https://arxiv.org/abs/2107.10659

[30] Moritz Hardt, Katrina Ligett, and Frank McSherry. 2012. A simple and practical algorithm for differentially private data release. *Advances in neural information processing systems* 25 (2012).

[31] Michael Hay, Ashwin Machanavajjhala, Gerome Miklau, Yan Chen, and Dan Zhang. 2016. Principled evaluation of differentially private algorithms using dpbench. In *Proceedings of the 2016 International Conference on Management of Data*. 139–154.

[32] 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).

[33] Roger A Horn and Charles R Johnson. 2012. *Matrix analysis*. Cambridge university press.

[34] Batta (https://math.stackexchange.com/users/488522/batta). [n.d.]. Linear Algebra, Vector Space: how to find intersection of two subspaces? Mathematics Stack Exchange. arXiv:https://math.stackexchange.com/q/2477195 https://math.stackexchange.com/q/2477195 URL:https://math.stackexchange.com/q/2477195 (version: 2017-10-18).

[35] 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.

[36] Noah M. Johnson, Joseph P. Near, Joseph M. Hellerstein, and Dawn Song. 2018. Chorus: Differential Privacy via Query Rewriting. *CoRR* abs/1809.07750 (2018). arXiv:1809.07750 http://arxiv.org/abs/1809.07750

[37] Daniel Kifer, John M. Abowd, Robert Ashmead, Ryan Cumings-Menon, Philip Leclerc, Ashwin Machanavajjhala, William Sexton, and Pavel Zhuravlev. 2022. Bayesian and Frequentist Semantics for Common Variations of Differential Privacy: Applications to the 2020 Census. https://doi.org/10.48550/ARXIV.2209.03310

[38] Fragkiskos Koufogiannis, Shuo Han, and George J Pappas. 2015. Gradual release of sensitive data under differential privacy. *arXiv preprint arXiv:1504.00429* (2015).

[39] Chao Li, Michael Hay, Gerome Miklau, and Yue Wang. 2014. A Data-and Workload-Aware Algorithm for Range Queries Under Differential Privacy. *Proceedings of the VLDB Endowment* 7, 5 (2014).

[40] Chao Li, Gerome Miklau, Michael Hay, Andrew Mcgregor, and Vibhor Rastogi. 2015. The Matrix Mechanism: Optimizing Linear Counting Queries under Differential Privacy. *The VLDB Journal* 24, 6 (Dec. 2015), 757–781. https://doi.org/10.1007/s00778-015-0398-x

[41] Yaping Li, Minghua Chen, Qiwei Li, and Wei Zhang. 2011. Enabling multilevel trust in privacy preserving data mining. *IEEE Transactions on Knowledge and Data Engineering* 24, 9 (2011), 1598–1612.

[42] Katrina Ligett, Seth Neel, Aaron Roth, Bo Waggoner, and Steven Z. Wu. 2017. Accuracy First: Selecting a Differential Privacy Level for Accuracy Constrained ERM. In *NIPS*.

[43] Jingcheng Liu and Kunal Talwar. 2018. Private Selection from Private Candidates. *arXiv preprint arXiv:1811.07971* (2018).

[44] Terrance Liu, Giuseppe Vietri, Thomas Steinke, Jonathan Ullman, and Steven Wu. 2021. Leveraging public data for practical private query release. In *International Conference on Machine Learning*. PMLR, 6968–6977.

[45] Terrance Liu, Giuseppe Vietri, and Steven Z Wu. 2021. Iterative methods for private synthetic data: Unifying framework and new methods. *Advances in Neural Information Processing Systems* 34 (2021).

[46] Terrance Liu and Steven Wu. 2022. Towards Differentially Private Query Release for Hierarchical Data. In *ICLR 2022 Workshop on PAIR^2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*. https://openreview.net/forum?id=BOulQJ9hLlq

[47] Eugene M. Luks, Ferenc Rákóczi, and Charles R.B. Wright. 1997. Some Algorithms for Nilpotent Permutation Groups. *Journal of Symbolic Computation* 23, 4 (1997),

335–354. https://doi.org/10.1006/jsco.1996.0092
[48] Min Lyu, Dong Su, and Ninghui Li. 2017. Understanding the sparse vector technique for differential privacy. *PVLDB* 10, 6 (2017), 637–648.
[49] Ashwin Machanavajjhala, Daniel Kifer, John Abowd, Johannes Gehrke, and Lars Vilhuber. 2008. Privacy: From Theory to Practice On the Map. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*. 277–286.
[50] Miti Mazmudar, Thomas Humphries, Matthew Rafuse, and Xi He. 2020. Cache Me If You Can: Accuracy-Aware Inference Engine for Differentially Private Data Exploration. In *TPDP*.
[51] Ryan McKenna, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. 2018. Optimizing error of high-dimensional statistical queries under differential privacy. *Proceedings of the VLDB Endowment* 11, 10 (2018).
[52] 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).
[53] Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. 2022. AIM: An Adaptive and Iterative Mechanism for Differentially Private Synthetic Data. *arXiv preprint arXiv:2201.12677* (2022).
[54] Frank McSherry and Kunal Talwar. 2007. Mechanism Design via Differential Privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS '07)*. IEEE Computer Society, Washington, DC, USA, 94–103.
[55] Solomon Messing, Bogdan State, Chaya Nayak, Gary King, and Nate Persily. 2018. URLs Dataset for RFP.pdf. In *Facebook URL Shares*. Harvard Dataverse. https://doi.org/10.7910/DVN/EIAACS/PMQG9X
[56] Ilya Mironov. 2017. Rényi Differential Privacy. In *30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017*. 263–275.
[57] Opacus [n.d.]. Opacus PyTorch library. Available from opacus.ai.
[58] Sofya Raskhodnikova and Adam D. Smith. 2016. Lipschitz Extensions for Node-Private Graph Statistics and the Generalized Exponential Mechanism. In *FOCS*. IEEE Computer Society, 495–504.
[59] Robert L. Smith. 1984. Efficient Monte Carlo Procedures for Generating Points Uniformly Distributed Over Bounded Regions. *Operations Research* 32, 6 (1984), 1296–1308. http://www.jstor.org/stable/170949
[60] Thomas Steinke and Jonathan Ullman. 2017. Tight lower bounds for differentially private selection. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 552–563.
[61] Nikolas Stott. 2016. Maximal lower bounds in the L\" owner order. *arXiv preprint arXiv:1612.05664* (2016).
[62] Nikolas STOTT. 2017. *l'Université Paris-Saclay*. Ph.D. Dissertation. University of Colorado.
[63] Nikolas Stott. 2017. *Minimal upper bounds in the löwner order and application to invariant computation for switched systems*. Ph.D. Dissertation. Université Paris Saclay (COmUE).
[64] Apple Differential Privacy Team. 2017. Learning with Privacy at Scale. *Apple Machine Learning Journal* 1, 8 (2017).
[65] Abhradeep Guha Thakurta and Adam Smith. 2013. Differentially Private Feature Selection via Stability Arguments, and the Robustness of the Lasso. In *Proceedings of the 26th Annual Conference on Learning Theory*.
[66] Royce Wilson, Celia Yuxin Zhang, William Lam, Damien Desfontaines, Daniel Simmons-Marengo, and Bryant Gipson. 2020. Differentially Private SQL with Bounded User Contribution. In *Proceedings on Privacy Enhancing Technologies Symposium*.
[67] Xiaokui Xiao, Gabriel Bender, Michael Hay, and Johannes Gehrke. 2011. iReduct: Differential privacy with reduced relative errors. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. 229–240.
[68] Xiaokui Xiao, Yufei Tao, and Minghua Chen. 2009. Optimal Random Perturbation at Multiple Privacy Levels. *Proc. VLDB Endow.* 2, 1 (aug 2009), 814–825.
[69] Yingtai Xiao, Zeyu Ding, Yuxin Wang, Danfeng Zhang, and Daniel Kifer. 2021. Optimizing fitness-for-use of differentially private linear queries. In *VLDB*.
[70] Yingtai Xiao, Guanhong Wang, Danfeng Zhang, and Daniel Kifer. 2022. Answering Private Linear Queries Adaptively using the Common Mechanism. https://doi.org/10.48550/ARXIV.2212.00135
[71] Jia Xu, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, Ge Yu, and Marianne Winslett. 2013. Differentially private histogram publication. *The VLDB journal* 22, 6 (2013), 797–822.
[72] Ganzhao Yuan, Yin Yang, Zhenjie Zhang, and Zhifeng Hao. 2016. Convex Optimization for Linear Query Processing under Approximate Differential Privacy. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
[73] Ganzhao Yuan, Zhenjie Zhang, Marianne Winslett, Xiaokui Xiao, Yin Yang, and Zhifeng Hao. 2012. Low-Rank Mechanism: Optimizing Batch Queries under Differential Privacy. *Proc. VLDB Endow.* 5, 11 (July 2012), 1352–1363. https://doi.org/10.14778/2350229.2350252
[74] Ganzhao Yuan, Zhenjie Zhang, Marianne Winslett, Xiaokui Xiao, Yin Yang, and Zhifeng Hao. 2015. Optimizing batch linear queries under exact and approximate differential privacy. *ACM Transactions on Database Systems (TODS)* 40, 2 (2015), 1–47.
[75] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2017. PrivBayes: Private Data Release via Bayesian Networks. *ACM Trans. Database Syst.* 42, 4, Article 25 (oct 2017), 41 pages.
[76] Xiaojian Zhang, Rui Chen, Jianliang Xu, Xiaofeng Meng, and Yingtao Xie. 2014. Towards accurate histogram publication under differential privacy. In *Proceedings of the 2014 SIAM international conference on data mining*. SIAM, 587–595.

A PROOFS FROM SECTION 3

Lemma A.1. *For any real matrix* $\mathbf{A} \in \mathbb{R}^{m\times n}$*, if* $\mathbf{A}\mathbf{A}^T \preceq \mathbf{I}_m$*, then* $\mathbf{A}^T\mathbf{A} \preceq \mathbf{I}_n$*.*

Proof of Lemma A.1. Let r be the rank of $\mathbf{A}$. Using the singular value decomposition, represent $\mathbf{A}$ as $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}$, where $\mathbf{U}$ is a $m \times m$ orthogonal matrix, $\mathbf{V}$ is a $n \times n$ orthogonal matrix, and $\mathbf{D}$ is an $m \times n$ matrix with the singular values $(s_1, s_2, \cdots, s_r)$ of $\mathbf{A}$ along its main diagonal (the other entries are 0). First, we are given that:

$$\begin{aligned}
\mathbf{A}\mathbf{A}^T &= \mathbf{U}\mathbf{D}\mathbf{D}^T\mathbf{U}^T \\
&= \mathbf{U}\ Diag_{m\times m}(s_1^2, s_2^2, \cdots, s_r^2, 0, \cdots, 0)\ \mathbf{U}^T \\
&\preceq \mathbf{I}_m
\end{aligned}$$

Noting that $\mathbf{U}^T\mathbf{U} = \mathbf{I}_m$, we have

$$\begin{aligned}
\mathbf{U}^T\mathbf{A}\mathbf{A}^T\mathbf{U} &= \mathbf{U}^T(\mathbf{U}\ Diag_{m\times m}(s_1^2, s_2^2, \cdots, s_r^2, 0, \cdots, 0)\ \mathbf{U}^T)\mathbf{U} \\
&= Diag_{m\times m}(s_1^2, s_2^2, \cdots, s_r^2, 0, \cdots, 0) \\
&\preceq \mathbf{U}^T\mathbf{I}_m\mathbf{U} = \mathbf{I}_m
\end{aligned}$$

which means that $s_i^2 \le 1$ for $i = 1, 2, \cdots, r$. Therefore,

$$\begin{aligned}
\mathbf{A}^T\mathbf{A} &= \mathbf{V}^T\mathbf{D}^T\mathbf{D}\mathbf{V} \\
&= \mathbf{V}^T Diag_{n\times n}(s_1^2, s_2^2, \cdots, s_r^2, 0, \cdots, 0)\mathbf{V} \\
&\preceq \mathbf{V}^T\mathbf{I}_n\mathbf{V} = \mathbf{V}^T\mathbf{V} = \mathbf{I}_n
\end{aligned}$$

since $\mathbf{I}_n - Diag_{n\times n}(s_1^2, s_2^2, \cdots, s_r^2, 0, \cdots, 0)$ is a diagonal matrix with nonnegative diagonals, and hence positive semidefinite. □

Lemma A.2. *Let* $r(\mathbf{B})$ *denote the row space of a matrix* $\mathbf{B}$*. If* $\mathbf{A}$ *and* $\mathbf{B}$ *are two symmetric positive semi-definite matrices and* $\mathbf{B} \preceq \mathbf{A}$*, then* $r(\mathbf{B}) \subseteq r(\mathbf{A})$*.*

Proof of Lemma A.2. Consider the linear space of vectors that are orthogonal to all of the rows of $\mathbf{A}$ and let $\mathbf{C}$ be a matrix whose columns consist of a full basis for this space (hence the columns are independent). Then $\mathbf{A}\mathbf{C} = \mathbf{0}$ and a vector $\mathbf{y}^T$ belongs to the rowspace of $\mathbf{A}$ if and only if $\mathbf{y}^T\mathbf{C} = \mathbf{0}$.

If $\mathbf{A} - \mathbf{B}$ is positive semidefinite, then so is $\mathbf{C}^T\mathbf{A}\mathbf{C} - \mathbf{C}^T\mathbf{B}\mathbf{C}$. But $\mathbf{C}^T\mathbf{A}\mathbf{C} - \mathbf{C}^T\mathbf{B}\mathbf{C} = -\mathbf{C}^T\mathbf{B}\mathbf{C}$ which is negative semidefinite, and so $-\mathbf{C}^T\mathbf{B}\mathbf{C}$ is both positive and negative semidefinite, meaning that its eigenvalues are all nonnegative and nonpositive, meaning that they are 0 and $\mathbf{C}^T\mathbf{B}\mathbf{C} = \mathbf{0}$.

Since $\mathbf{B}$ is a symmetric positive semidefinite matrix of rank (say) r, it can be written as $\mathbf{B} = \mathbf{V}\mathbf{D}\mathbf{V}^T$ where $\mathbf{D}$ is an $r \times r$ diagonal matrix containing the r positive eigenvalues of $\mathbf{B}$ (if all eigenvalues are 0 the lemma is trivially true) and $\mathbf{V}$ is an $m \times r$ matrix whose columns are orthogonal.

Thus if $\mathbf{C}^T\mathbf{B}\mathbf{C} = \mathbf{0}$ then $\mathbf{C}^T\mathbf{V}\mathbf{D}\mathbf{V}^T\mathbf{C} = \mathbf{0}$ and so for any column vector $\mathbf{c}_i$ of $\mathbf{C}$,

$$\begin{aligned}
0 &= \mathbf{c}_i^T\mathbf{V}\mathbf{D}\mathbf{V}^T\mathbf{c}_i \\
&= \sum_j (\mathbf{c}_i \cdot \mathbf{v}_j)^2\mathbf{D}[j, j] \quad \text{where } \mathbf{v}_j \text{ is the } j^{\text{th}} \text{ column of } \mathbf{V}
\end{aligned}$$

Noting that the diagonals of $\mathbf{D}$ are all positive, this means that $\mathbf{v}_j^T\mathbf{c}_i = 0$ for all i and j. Thus $\mathbf{B}\mathbf{C} = \mathbf{V}\mathbf{D}\mathbf{V}^T\mathbf{C} = \mathbf{0}$ and therefore the rows of $\mathbf{B}$ belong to the row space of $\mathbf{A}$. □

Lemma A.3. *Let* $r(\mathbf{B})$ *denote the row space of the matrix* $\mathbf{B}$*. If* $r(\mathbf{B}) \subseteq r(\mathbf{A})$*, then* $\mathbf{B} = \mathbf{B}\mathbf{A}^+\mathbf{A}$*, where* $\mathbf{A}^+$ *is the Moore-Penrose pseudoinverse of* $\mathbf{A}$*.*

Proof of Lemma A.3. If $r(\mathbf{B}) \subset r(\mathbf{A})$ then $\mathbf{B} = \mathbf{C}\mathbf{A}$ for some matrix $\mathbf{C}$. Then

$$\mathbf{B}\mathbf{A}^+\mathbf{A} = \mathbf{C}\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{C}\mathbf{A} = \mathbf{B}$$

by the property of pseudoinverses ($\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A}$). □

Theorem 3.5. *Lets* $\mathcal{M}_a(\mathbf{x}) = \mathbf{B}_a\mathbf{x} + N(\mathbf{0}, \Sigma_a)$ *and* $\mathcal{M}_b(\mathbf{x}) = \mathbf{B}_b\mathbf{x} + N(\mathbf{0}, \Sigma_b)$ *be linear Gaussian mechanisms.* $\mathcal{M}_b$ *is exactly answerable from* $\mathcal{M}_a$ *if and only if* $\mathbf{B}_b^T\Sigma_b^{-1}\mathbf{B}_b \preceq \mathbf{B}_a^T\Sigma_a^{-1}\mathbf{B}_a$ *(i.e.,* $\mathbf{B}_a^T\Sigma_a^{-1}\mathbf{B}_a - \mathbf{B}_b^T\Sigma_b^{-1}\mathbf{B}_b$ *is positive semidefinite, and hence its eigenvalues are nonnegative).*

Proof of Theorem 3.5. First, we prove that "if" direction. Suppose $\mathcal{M}_b$ is exactly answerable from $\mathcal{M}_a$. By definition, there exist matrices $\mathbf{A}$ and $\mathbf{C}$ such that

$$\begin{aligned}
\mathbf{B}_b &= \mathbf{A}\mathbf{B}_a \\
\Sigma_b &= \mathbf{A}\Sigma_a\mathbf{A}^T + \mathbf{C}\mathbf{C}^T \\
&\text{and so} \\
\mathbf{A}\Sigma_a\mathbf{A}^T &\preceq \Sigma_b
\end{aligned}$$

Let $\mathbf{B}_{a\star} = \Sigma_a^{-1/2}\mathbf{B}_a$ (where $\Sigma_a^{1/2}$ is the inverse of a symmetric positive definite matrix square root of Σ_a) and $\mathbf{B}_{b\star} = \Sigma_b^{-1/2}\mathbf{B}_b$. Then let $\mathbf{A}_\star = \Sigma_b^{-1/2}\mathbf{A}\Sigma_a^{1/2}$, then we have

$$\begin{aligned}
\mathbf{B}_{b\star} &= \Sigma_b^{-1/2}\mathbf{B}_b \\
&= \Sigma_b^{-1/2}\mathbf{A}\mathbf{B}_a \\
&= \Sigma_b^{-1/2}\mathbf{A}\Sigma_a^{1/2}\Sigma_a^{-1/2}\mathbf{B}_a \\
&= \mathbf{A}_\star\mathbf{B}_{a\star} \\
\mathbf{A}_\star\mathbf{A}_\star^T &= \left(\Sigma_b^{-1/2}\mathbf{A}\Sigma_a^{1/2}\right)\left(\Sigma_b^{-1/2}\mathbf{A}\Sigma_a^{1/2}\right)^T \\
&= \Sigma_b^{-1/2}\mathbf{A}\Sigma_a\mathbf{A}^T\Sigma_b^{-1/2} \\
&\preceq \Sigma_b^{-1/2}\Sigma_b\Sigma_b^{-1/2} \quad (\text{ since } \mathbf{A}\Sigma_a\mathbf{A}^T \preceq \Sigma_b) \\
&= \mathbf{I}_{m_b} \quad (\text{where } m_b \text{ is the number of rows of } \mathbf{B}_b)
\end{aligned}$$

This means that $\mathbf{A}_\star\mathbf{A}_\star^T \preceq \mathbf{I}_{m_b}$ and so by Lemma A.1 we know that $\mathbf{A}_\star^T\mathbf{A}_\star \preceq \mathbf{I}_{m_a}$, where m_a is the number of rows of $\mathbf{B}_a$. Thus

$$\begin{aligned}
\mathbf{B}_b^T\Sigma_b^{-1}\mathbf{B}_b &= \mathbf{B}_a^T\mathbf{A}^T\Sigma_b^{-1}\mathbf{A}\mathbf{B}_a \\
&= (\mathbf{B}_a^T\Sigma_a^{-1/2})\Sigma_a^{1/2}\mathbf{A}^T\Sigma_b^{-1}\mathbf{A}\Sigma_a^{1/2}(\Sigma_a^{-1/2}\mathbf{B}_a) \\
&= (\mathbf{B}_a^T\Sigma_a^{-1/2})(\Sigma_a^{1/2}\mathbf{A}^T\Sigma_b^{-1/2})(\Sigma_b^{-1/2}\mathbf{A}\Sigma_a^{1/2})(\Sigma_a^{-1/2}\mathbf{B}_a) \\
&= \mathbf{B}_{a\star}^T\mathbf{A}_\star^T\mathbf{A}_\star\mathbf{B}_{a\star} \\
&\preceq \mathbf{B}_{a\star}^T\mathbf{I}_{m_a}\mathbf{B}_{a\star} \quad (\text{since } \mathbf{A}_\star^T\mathbf{A}_\star \preceq \mathbf{I}_{m_a}) \\
&= \mathbf{B}_{a\star}^T\mathbf{B}_{a\star} \\
&= \mathbf{B}_a^T\Sigma_a^{-1}\mathbf{B}_a
\end{aligned}$$

so if $\mathcal{M}_b$ is exactly answerable from $\mathcal{M}_a$, we have $\mathbf{B}_b^T\Sigma_b^{-1}\mathbf{B}_b \preceq \mathbf{B}_a^T\Sigma_a^{-1}\mathbf{B}_a$

Now we must prove the other direction (i.e., if $\mathbf{B}_b^T\Sigma_b^{-1}\mathbf{B}_b \preceq \mathbf{B}_a^T\Sigma_a^{-1}\mathbf{B}_a$, we must show $\mathcal{M}_b$ is exactly answerable from $\mathcal{M}_a$).

As before, let $\mathbf{B}_{a\star} = \Sigma_a^{-1/2}\mathbf{B}_a$ (where $\Sigma_a^{1/2}$ is the inverse of a symmetric positive definite matrix square root of Σ_a) and $\mathbf{B}_{b\star} = \Sigma_b^{-1/2}\mathbf{B}_b$. Consider the mechanisms:

$$\mathcal{M}_{a\star}(x) = \mathbf{B}_{a\star} + N(0, \mathbf{I}_{m_a})$$
$$\mathcal{M}_{b\star}(x) = \mathbf{B}_{b\star} + N(0, \mathbf{I}_{m_b})$$

We have

$$\begin{aligned}\mathbf{B}_{b\star}^T\mathbf{B}_{b\star} &= \mathbf{B}_b^T\Sigma_b^{-1}\mathbf{B}_b \\ &\preceq \mathbf{B}_a^T\Sigma_a^{-1}\mathbf{B}_a \\ &= \mathbf{B}_{a\star}^T\mathbf{B}_{a\star}\end{aligned}$$

Let $r(\mathbf{C})$ denote the row space of a matrix $\mathbf{C}$. From Lemma A.2 we know that

$$r(\mathbf{B}_{b\star}) = r(\mathbf{B}_{b\star}^T\mathbf{B}_{b\star}) \subseteq r(\mathbf{B}_{a\star}^T\mathbf{B}_{a\star}) = r(\mathbf{B}_{a\star})$$

Let $\mathbf{A}_\star = \mathbf{B}_{b\star}\mathbf{B}_{a\star}^+$, then from lemma A.3 we have $\mathbf{B}_{b\star} = \mathbf{A}_\star\mathbf{B}_{a\star}$.

Now, the dimensions of $\mathbf{B}_a$ and $\mathbf{B}_b$ (and hence $\mathbf{B}_{a*}$ and $\mathbf{B}_{b*}$ are $m_a \times d$ and $m_b \times d$ (they have the same number of columns because they are defined for mechanisms whose input is a vector $\mathbf{x}$ with d components).

Taking the SVD, we can get the following representation:

$$\mathbf{B}_{b\star} = \mathbf{U}_b\mathbf{D}_b\mathbf{V}_b$$
$$\mathbf{B}_{a\star} = \mathbf{U}_a\mathbf{D}_a\mathbf{V}_a$$

and so

$$\mathbf{B}_{a\star}^+ = \mathbf{V}_a^T\mathbf{D}_a^+\mathbf{U}_a^T$$

where $\mathbf{U}_a$ (resp., $\mathbf{U}_b$)is an $m_a \times m_a$ (resp., $m_b \times m_b$) orthogonal matrix, $\mathbf{D_a}$ (resp., $\mathbf{D}_b$) is an $m_a \times d$ (resp., $m_b \times d$) matrix, and both $\mathbf{V}_a$ and $\mathbf{V}_b$ are $d \times d$ orthogonal matrices.

Since $\mathbf{B}_{b\star}^T\mathbf{B}_{b\star} \preceq \mathbf{B}_{a\star}^T\mathbf{B}_{a\star}$ we know that

$$\mathbf{V}_b^T\mathbf{D}_b^T\mathbf{D}_b\mathbf{V}_b \preceq \mathbf{V}_a^T\mathbf{D}_a^T\mathbf{D}_a\mathbf{V}_a$$

and

$$\begin{aligned}\mathbf{A}_\star^T\mathbf{A}_\star &= \left(\mathbf{B}_{a\star}^{+T}\mathbf{B}_{b\star}^T\right)\left(\mathbf{B}_{b\star}\mathbf{B}_{a\star}^+\right) \\ &= \left(\mathbf{U}_a\mathbf{D}_a^{+T}\mathbf{V}_a\mathbf{V}_b^T\mathbf{D}_b^T\mathbf{U}_b^T\right)\left(\mathbf{U}_b\mathbf{D}_b\mathbf{V}_b\mathbf{V}_a^T\mathbf{D}_a^+\mathbf{U}_a^T\right) \\ &= \mathbf{U}_a\mathbf{D}_a^{+T}\mathbf{V}_a\underline{\mathbf{V}_b^T\mathbf{D}_b^T\mathbf{D}_b\mathbf{V}_b}\mathbf{V}_a^T\mathbf{D}_a^+\mathbf{U}_a^T \\ &\preceq \mathbf{U}_a\mathbf{D}_a^{+T}\mathbf{V}_a\underline{\mathbf{V}_a^T\mathbf{D}_a^T\mathbf{D}_a\mathbf{V}_a}\mathbf{V}_a^T\mathbf{D}_a^+\mathbf{U}_a^T \\ &= \mathbf{U}_a\mathbf{D}_a^{+T}\mathbf{D}_a^T\mathbf{D}_a\mathbf{D}_a^+\mathbf{U}_b^T\end{aligned}$$

Because $\mathbf{D}_a$ is a $m_a \times d$ diagonal matrix, where the elements of its main diagonal are the singular values s_i of $\mathbf{B}_a$, then $\mathbf{D}_a^+$ is a $d \times m_a$ diagonal matrix, where the elements of its main diagonal are $\frac{1}{s_i}$ for the nonzero singular values (and 0 otherwise). So, that $\mathbf{D}_a\mathbf{D}_a^+$ is a $m_a \times m_a$ diagonal matrix where the elements of its diagonal are either 1 or 0. The same thing holds for $\mathbf{D}_a^{+T}\mathbf{D}_a^T$. Therefore, $\mathbf{D}_a\mathbf{D}_a^+\mathbf{D}_a^{+T}\mathbf{D}_a^T \preceq \mathbf{I}_{m_a}$ and

$$\begin{aligned}\mathbf{A}_\star^T\mathbf{A}_\star &\preceq \mathbf{U}_a\mathbf{I}_{m_a}\mathbf{U}_a^T \\ &= \mathbf{U}_a\mathbf{U}_a^T = \mathbf{I}_{m_a}\end{aligned}$$

From Lemma A.1 we have

$$\mathbf{A}_\star\mathbf{A}_\star^T \preceq \mathbf{I}_{m_b}$$

Let $\mathbf{A} = \Sigma_b^{1/2}\mathbf{A}_\star\Sigma_a^{-1/2}$, we can see that

$$\begin{aligned}\mathbf{B}_b &= \Sigma_b^{1/2}\mathbf{B}_{b\star} \\ &= \Sigma_b^{1/2}\mathbf{A}_\star\mathbf{B}_{a\star} \quad \text{which we showed previously} \\ &= \Sigma_b^{1/2}\mathbf{A}_\star\Sigma_a^{-1/2}\mathbf{B}_a \\ &= \mathbf{A}\mathbf{B}_a \\ \Sigma_b &= \Sigma_b^{1/2}\mathbf{I}_{m_b}\Sigma_b^{1/2} \\ &\succeq \Sigma_b^{1/2}\mathbf{A}_\star\mathbf{A}_\star^T\Sigma_b^{1/2} \quad \text{since we showed} \mathbf{A}_\star\mathbf{A}_\star^T \preceq \mathbf{I}_{m_b} \\ &= \mathbf{A}\Sigma_a\mathbf{A}^T\end{aligned}$$

Then, for the purposes of exact answerability in Definition 3.1, we have defined the matrix $\mathbf{A}$. The matrix $\mathbf{C}$ can be obtained by noting that $\mathbf{B}_b - \mathbf{A}\Sigma_a\mathbf{A}^T$ is positive semidefinite and so has a symmetric matrix square root. Setting $\mathbf{C}$ to be this square root, we have $\mathbf{B}_b = \mathbf{A}\Sigma_a\mathbf{A}^T + \mathbf{C}\mathbf{C}^T$ and therefore $\mathcal{M}_b$ is answerable from $\mathcal{M}_a$ by Lemma 3.2. ◻

Lemma 3.11. *Suppose that $\mathcal{M}_1(\mathbf{x}) = \mathbf{B}_1\mathbf{x} + N(\mathbf{0}, \Sigma_1)$ and $\mathcal{M}_2(\mathbf{x}) = \mathbf{B}_2\mathbf{x} + N(\mathbf{0}, \Sigma_2)$ are linear Gaussian mechanisms and that $\mathcal{M}_*(\mathbf{x}) = \mathbf{B}_*\mathbf{x} + N(\mathbf{0}, \Sigma_*)$ is their maximally common mechanism. Then $\mathcal{M}_1'(\mathbf{x}) = \mathbf{B}_1'\mathbf{x} + N(\mathbf{0}, \Sigma_1')$ and $\mathcal{M}_2'(\mathbf{x}) = \mathbf{B}_2'\mathbf{x} + N(\mathbf{0}, \Sigma_2')$ are residual mechanisms if and only if:*

$$\mathbf{B}_*{}^T\Sigma_*{}^{-1}\mathbf{B}_* \;+\; (\mathbf{B}_1')^T(\Sigma_1')^{-1}\mathbf{B}_1' \;=\; \mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1$$
$$\mathbf{B}_*{}^T\Sigma_*{}^{-1}\mathbf{B}_* \;+\; (\mathbf{B}_2')^T(\Sigma_2')^{-1}\mathbf{B}_2' \;=\; \mathbf{B}_2^T\Sigma_2^{-1}\mathbf{B}_2$$

In which case $(\mathcal{M}_*, \mathcal{M}_1')$ is equivalent to $\mathcal{M}_1$ and $(\mathcal{M}_*, \mathcal{M}_2')$ is equivalent to $\mathcal{M}_2$.

Proof of Lemma 3.11. We prove the result for $\mathcal{M}_2$ as the result for $\mathcal{M}_2$ is similar. The combined mechanism $(\mathcal{M}_*, \mathcal{M}_1')$ can be expressed in matrix form as follows:

$$\begin{bmatrix}\mathbf{B}_* \\ \mathbf{B}_1'\end{bmatrix}\mathbf{x} + N\left(\mathbf{0}, \begin{bmatrix}\Sigma_* & \mathbf{0} \\ \mathbf{0} & \Sigma_1'\end{bmatrix}\right)$$

Thus $(\mathcal{M}_*, \mathcal{M}_1')$ is a linear Gaussian mechanism with $\mathbf{B} = \begin{bmatrix}\mathbf{B}_* \\ \mathbf{B}_1'\end{bmatrix}$ and $\Sigma = \begin{bmatrix}\Sigma_* & \mathbf{0} \\ \mathbf{0} & \Sigma_1'\end{bmatrix}$. The privacy cost is

$$\mathbf{B}^T\Sigma^{-1}\mathbf{B} = \mathbf{B}_*{}^T\Sigma_*{}^{-1}\mathbf{B}_* \;+\; (\mathbf{B}_1')^T(\Sigma_1')^{-1}\mathbf{B}_1'$$

and the rest follows directly from Theorem 3.5. ◻

B PROOFS FROM SECTION 5

Lemma 5.1. *Let $\mathcal{M}_{orig}$ be a linear Gaussian mechanism with query matrix $\mathbf{B}_{orig}$ and covariance matrix Σ_{orig}. Let $\mathcal{M}$ be a linear Gaussian mechanism with identity covariance and query matrix $\mathbf{B}$ obtained by running Algorithm 2 on $\mathcal{M}_{orig}$. Then $\mathcal{M}_{orig}$ and $\mathcal{M}$ are equivalent.*

Proof of Lemma 5.1. Using the eigenvalue decomposition of Algorithm 2, we have the privacy cost matrix of $\mathcal{M}_{orig}$ is $\lambda_1\mathbf{v}_1\mathbf{v}_1^T + \cdots + \lambda_k\mathbf{v}_k\mathbf{v}_k^T + \cdots + \lambda_d\mathbf{v}_d\mathbf{v}_d^T$ which equals $\lambda_1\mathbf{v}_1\mathbf{v}_1^T + \cdots + \lambda_k\mathbf{v}_k\mathbf{v}_k^T$ since the eigenvalues $\lambda_{k+1}, \ldots, \lambda_d$ are all 0. But that is the same as

$\mathbf{B}^T\mathbf{I}^{-1}\mathbf{B}$, the privacy cost matrix of $\mathcal{M}$, due Line 3 in the algorithm, and so they are equivalent by Theorem 3.5. □

Lemma B.1. *Let $\mathcal{M}(\mathbf{x}) = \mathbf{B}\mathbf{x} + N(\mathbf{0}, \Sigma)$ be a linear Gaussian mechanism. Let $\mathbf{B}'$ be a matrix with linearly independent rows such that rowspace$(\mathbf{B})$ = rowspace$(\mathbf{B}')$. Then there exists a mechanism that is equivalent to $\mathcal{M}$ and uses the query matrix $\mathbf{B}'$.*

Proof of Lemma B.1. Let r be the number of rows of $\mathbf{B}$ (so it is a $r \times d$) matrix. This means that Σ has size $r \times r$. Let r' be the number of rows of $\mathbf{B}^p rime$.

Define $\mathbf{S}$ to be a $r \times r$ symmetric matrix square root of Σ (which is possible because Σ is positive semi-definite). Let $\mathbf{A}$ be a $r \times r'$ matrix such that $\mathbf{B} = \mathbf{A}\mathbf{B}'$ (which is possible because all the rows of $\mathbf{B}$ are in the row space of $\mathbf{B}'$). Note that the rank of $\mathbf{B}$ and $\mathbf{B}'$ is the same, which equals r' and is the number of rows of $\mathbf{B}'$. This means that the rank of $\mathbf{A}$ is also r'.

Now consider the matrix $\mathbf{A}^T\Sigma^{-1}\mathbf{A}$ which is a symmetric positive semidefinite matrix of dimension $r' \times r'$. Since $\mathbf{A}$ has rank r' and Σ has rank $r \geq r'$, this matrix is invertible and therefore positive definite. So we can define $\Sigma' = \left(\mathbf{A}^T\Sigma^{-1}\mathbf{A}\right)^{-1}$.

Consider the mechanism $\mathcal{M}'(\mathbf{x}) = \mathbf{B}'\mathbf{x} + N(\mathbf{0}, \Sigma')$. We have

$$\begin{aligned}(\mathbf{B}')^T(\Sigma')^{-1}\mathbf{B}' &= (\mathbf{B}')^T\mathbf{A}^T\Sigma^{-1}\mathbf{A}\mathbf{B}' \\ &= \mathbf{B}^T\Sigma^{-1}\mathbf{B}'\end{aligned}$$

and therefore $\mathcal{M}$ and $\mathcal{M}'$ are equivalent by Theorem 3.5 and Definition 3.6. □

Theorem 5.2. *Let $\mathcal{M}_1(\mathbf{x}) = \mathbf{B}_1\mathbf{x} + N(\mathbf{0}, \Sigma_1)$ and $\mathcal{M}_2(\mathbf{x}) = \mathbf{B}_2\mathbf{x} + N(\mathbf{0}, \Sigma_2)$ be linear Gaussian mechanisms.*

- *If $\mathcal{M}_c(\mathbf{x}) = \mathbf{B}_c\mathbf{x} + N(\mathbf{0}, \Sigma_c)$ is common to $\mathcal{M}_1$ and $\mathcal{M}_2$ then rowspace$(\mathbf{B}_c) \subseteq$ rowspace$(\mathbf{B}_1) \cap$ rowspace$(\mathbf{B}_2)$.*
- *If $\mathcal{M}_c$ is* maximally *common then rowspace$(\mathbf{B}_c)$ = rowspace$(\mathbf{B}_1) \cap$ rowspace$(\mathbf{B}_2)$*
- *The choice for basis of rowspace$(\mathbf{B}_1) \cap$ rowspace$(\mathbf{B}_2)$ does not matter. If $\mathcal{M}_c$ is maximally common and if $\mathbf{B}_* \neq \mathbf{B}_c$ is any matrix whose rows form a linearly independent basis, then there exists a common mechanism that is equivalent to $\mathcal{M}_c$ and has query matrix $\mathbf{B}_*$.*

Proof of Theorem 5.2. **First Item**. In order for $\mathcal{M}_c$ to be common to $\mathcal{M}_1$ and $\mathcal{M}_2$, by Lemma 3.5, there must exist matrices $\mathbf{A}_1$, $\mathbf{A}_2$, such that

$$\begin{aligned}\mathbf{B}_c &= \mathbf{A}_1\mathbf{B}_1 \\ \mathbf{B}_c &= \mathbf{A}_2\mathbf{B}_2\end{aligned}$$

and therefore rowspace$(\mathbf{B}_c) \subseteq$ rowspace$(\mathbf{B}_1) \cap$ rowspace$(\mathbf{B}_2)$.

Second Item. If $\mathcal{M}_c$ is maximally common but rowspace$(\mathbf{B}_c) \subset$ rowspace$(\mathbf{B}_1) \cap$ rowspace$(\mathbf{B}_2)$ then there exists a unit (column) vector $\mathbf{v} \in$ rowspace$(\mathbf{B}_2)$ that is orthogonal to rowspace$(\mathbf{B}_c)$. Define $\frac{1}{\sigma^2} = \min(\mathbf{v}^T\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1)\mathbf{v},\ \mathbf{v}^T\mathbf{B}_2^T\Sigma_2^{-1}\mathbf{B}_2\mathbf{v})$ and note that $\frac{1}{\sigma^2}$ is nonzero because $\mathbf{v}$ is in the row space of these matrices and the covariance matrices are nonsingular. Define

$$\begin{aligned}\mathbf{B}' &= \begin{bmatrix}\mathbf{B}_c \\ \mathbf{v}^T\end{bmatrix} \\ \Sigma' &= \begin{bmatrix}\Sigma_c & \mathbf{0} \\ \mathbf{0} & \sigma^2\end{bmatrix}\end{aligned}$$

and consider the mechanism $\mathcal{M}'(\mathbf{x}) = \mathbf{B}'\mathbf{x} + N(\mathbf{0}, \Sigma')$. Clearly, $\mathcal{M}_c$ is answerable from $\mathcal{M}'$ simply by dropping the last query answer from $\mathcal{M}$ (corresponding to the last row of $\mathbf{B}'$). On the other hand, it is easy to see that $\mathcal{M}'$ is not answerable from $\mathcal{M}_c$.

By Lemma B.1 we can, without loss of generality, assume that $\mathbf{B}_1$ has linearly independent rows, that one of the rows is $\mathbf{v}^T$ and the rest of the rows of $\mathbf{B}_1$ are orthogonal to $\mathbf{v}^T$ (since $\mathbf{v}^T$ is in the row space of $\mathbf{B}_1$). The same can be assumed of $\mathbf{B}_2$, so that we can write $\mathbf{B}_1 = \begin{bmatrix}\mathbf{B}_1^\circ \\ \mathbf{v}^T\end{bmatrix}$ for some $\mathbf{B}_1^\circ$ whose rows are orthogonal to $\mathbf{v}^T$ and similarly $\mathbf{B}_2 = \begin{bmatrix}\mathbf{B}_2^\circ \\ \mathbf{v}^T\end{bmatrix}$

Now let $\mathbf{w}$ be any vector orthogonal to $\mathbf{v}$. Since $\mathcal{M}_c$ is common to $\mathcal{M}_1$ and $\mathcal{M}_2$, then by Theorem 3.5, we must have:

$$\begin{aligned}0 &\leq \mathbf{w}^T(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1 - \mathbf{B}_c^T\Sigma_c^{-1}\mathbf{B}_c)\mathbf{w}^T \\ &\mathbf{w}^T\left(\begin{bmatrix}\mathbf{B}_1^\circ \\ \mathbf{v}^T\end{bmatrix}^T\Sigma_1^{-1}\begin{bmatrix}\mathbf{B}_1^\circ \\ \mathbf{v}^T\end{bmatrix} - \mathbf{B}_c^T\Sigma_c^{-1}\mathbf{B}_c\right)\mathbf{w}^T \\ &= \mathbf{w}^T\left(\begin{bmatrix}\mathbf{B}_1^\circ \\ \mathbf{0}^T\end{bmatrix}^T\Sigma_1^{-1}\begin{bmatrix}\mathbf{B}_1^\circ \\ \mathbf{0}^T\end{bmatrix} - \mathbf{B}_c^T\Sigma_c^{-1}\mathbf{B}_c\right)\mathbf{w}^T \qquad (2) \\ 0 &\leq \mathbf{w}^T(\mathbf{B}_2^T\Sigma_2^{-1}\mathbf{B}_2 - \mathbf{B}_c^T\Sigma_c^{-1}\mathbf{B}_c)\mathbf{w}^T \\ &= \mathbf{w}^T\left(\begin{bmatrix}\mathbf{B}_2^\circ \\ \mathbf{0}^T\end{bmatrix}^T\Sigma_2^{-1}\begin{bmatrix}\mathbf{B}_2^\circ \\ \mathbf{0}^T\end{bmatrix} - \mathbf{B}_c^T\Sigma_c^{-1}\mathbf{B}_c\right)\mathbf{w}^T \qquad (3)\end{aligned}$$

Also, by construction of σ^2, we have:

$$\mathbf{v}^T\left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1 - \frac{1}{\sigma^2}\mathbf{v}^T\mathbf{v}\right)\mathbf{v}^T = \mathbf{v}^T\left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1\right)\mathbf{v} - \frac{1}{\sigma^2} \geq 0 \qquad (4)$$

$$\mathbf{v}^T\left(\mathbf{B}_2^T\Sigma_2^{-1}\mathbf{B}_2 - \frac{1}{\sigma^2}\mathbf{v}^T\mathbf{v}\right)\mathbf{v}^T \geq 0 \qquad (5)$$

Now we show that $\mathcal{M}'$ is also common to $\mathcal{M}_1$ and $\mathcal{M}_2$.

Any vector $\mathbf{z}$ can be represented as a linear combination $\mathbf{w} + b\mathbf{v}$ where $\mathbf{w}$ is a vector that is orthogonal to $\mathbf{v}$ (and the choice of $b, \mathbf{w}$ depends on $\mathbf{z}$).

$$\begin{aligned}&\mathbf{z}^T\left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1 - (\mathbf{B}')^T(\Sigma')^{-1}(\mathbf{B}')\right)\mathbf{z} \\ &= (\mathbf{w} + b\mathbf{v})^T\left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1 - (\mathbf{B}')^T(\Sigma')^{-1}(\mathbf{B}')\right)(\mathbf{w} + b\mathbf{v}) \\ &= (\mathbf{w} + b\mathbf{v})^T\left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1 - \begin{bmatrix}\mathbf{B}_c \\ \mathbf{v}^T\end{bmatrix}^T\begin{bmatrix}\Sigma_c & \mathbf{0} \\ \mathbf{0} & \sigma^2\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{B}_c \\ \mathbf{v}^T\end{bmatrix}\right)(\mathbf{w} + b\mathbf{v}) \\ &= (\mathbf{w} + b\mathbf{v})^T\left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1 - \begin{bmatrix}\mathbf{B}_c^T, & \mathbf{v}\end{bmatrix}\begin{bmatrix}\Sigma_c^{-1} & \mathbf{0} \\ \mathbf{0} & 1/\sigma^2\end{bmatrix}\begin{bmatrix}\mathbf{B}_c \\ \mathbf{v}^T\end{bmatrix}\right)(\mathbf{w} + b\mathbf{v}) \\ &= (\mathbf{w} + b\mathbf{v})^T\left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1 - \mathbf{B}_c^T\Sigma_c^{-1}\mathbf{B}_c - \frac{1}{\sigma^2}\mathbf{v}\mathbf{v}^T\right)(\mathbf{w} + b\mathbf{v}) \\ &= \mathbf{w}^T\left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1 - \mathbf{B}_c^T\Sigma_c^{-1}\mathbf{B}_c\right)\mathbf{w}^T + b^2\mathbf{v}^T\left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1 - \frac{1}{\sigma^2}\mathbf{v}^T\mathbf{v}\right)\mathbf{v}^T \\ &\quad + 2b\mathbf{w}^T\left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1\right)\mathbf{v}\end{aligned}$$

Since $\mathbf{v}^T$ is orthogonal to $\mathbf{w}$ and the rows of $\mathbf{B}_c$

$$= \mathbf{w}^T \left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1 - \mathbf{B}_c^T\Sigma_c^{-1}\mathbf{B}_c\right)\mathbf{w}^T + b^2\mathbf{v}^T\left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1 - \frac{1}{\sigma^2}\mathbf{v}^T\mathbf{v}\right)\mathbf{v}^T$$
$$+ 2b\mathbf{w}^T\left(\left[\,(\mathbf{B}_1^\circ)^T,\ \mathbf{v}\,\right]\Sigma_1^{-1}\begin{bmatrix}\mathbf{B}_1^\circ\\ \mathbf{v}^T\end{bmatrix}\right)\mathbf{v}$$
$$= \mathbf{w}^T \left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1 - \mathbf{B}_c^T\Sigma_c^{-1}\mathbf{B}_c\right)\mathbf{w}^T + b^2\mathbf{v}^T\left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1 - \frac{1}{\sigma^2}\mathbf{v}^T\mathbf{v}\right)\mathbf{v}^T$$
$$+ 2b\mathbf{w}^T\left(\left[\,(\mathbf{B}_1^\circ)^T,\ 0\,\right]\Sigma_1^{-1}\begin{bmatrix}0\\ \mathbf{v}^T\end{bmatrix}\right)\mathbf{v}$$

Since $\mathbf{v}^T$ is orthogonal to $\mathbf{w}$ and the rows of $\mathbf{B}_1^\circ$

$$= \mathbf{w}^T \left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1 - \mathbf{B}_c^T\Sigma_c^{-1}\mathbf{B}_c\right)\mathbf{w}^T + b^2\mathbf{v}^T\left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1 - \frac{1}{\sigma^2}\mathbf{v}^T\mathbf{v}\right)\mathbf{v}^T$$

and both terms are nonnegative because of Equations 2 and 4. A similar result holds for $\mathbf{B}_2$. Thus, by Theorem 3.5, $\mathcal{M}'$ is common to $\mathcal{M}_1$ and $\mathcal{M}_2$.

Since $\mathcal{M}_c$ is answerable from $\mathcal{M}'$ (but not vice versa) then $\mathcal{M}_c$ is not maximal. This contradiction arose from the assumption that the row space of $\mathbf{B}_c$ was a strict subset of the intersection of the row spaces of $\mathbf{B}_1$ and $\mathbf{B}_2$.

Third item. This follows directly from Lemma B.1.

□

Theorem 5.3. *Let $\mathcal{M}_1(\mathbf{x}) = \mathbf{B}_1\mathbf{x}+N(\mathbf{0},\mathbf{I})$ and $\mathcal{M}_2(\mathbf{x}) = \mathbf{B}_2\mathbf{x}+N(\mathbf{0},\mathbf{I})$ be linear Gaussian mechanisms that are standardized (i.e., produced by Algorithm 2). Let $\mathbf{B}_*$ be a matrix whose rows are linearly independent and spans rowspace$(\mathbf{B}_1)\cap$ rowspace$(\mathbf{B}_2)$. Then one can obtain a maximally common mechanism by using the Σ_* that optimizes the following problem (here † represents the Moore-Penrose Pseudoinverse operation):*

$$\Sigma_* \leftarrow \min_{\Sigma}\ trace(\Sigma)\ s.t.\ \Sigma \geq \mathbf{B}_*\mathbf{B}_1^\dagger(\mathbf{B}_1^\dagger)^T\mathbf{B}_*{}^T \qquad (1)$$
$$\Sigma \geq \mathbf{B}_*\mathbf{B}_2^\dagger(\mathbf{B}_2^\dagger)^T\mathbf{B}_*{}^T$$

Proof of Theorem 5.3. In order for a mechanism $\mathcal{M}_*(\mathbf{x}) = \mathbf{B}_*\mathbf{x} + N(\mathbf{0},\Sigma_*)$ to be answerable from $\mathcal{M}_1$ and $\mathcal{M}_2$, by Lemma 3.1, there must exist matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ such that

$$\mathbf{B}_* = \mathbf{A}_1\mathbf{B}_1$$
$$\mathbf{B}_* = \mathbf{A}_2\mathbf{B}_2$$
$$\Sigma_*\succeq\mathbf{A}_1\mathbf{A}_1^T$$
$$\Sigma_*\succeq\mathbf{A}_2\mathbf{A}_2^T$$

since $\Sigma_1 = \mathbf{I}$ and $\Sigma_2 = \mathbf{I}$. Since the rows of $\mathbf{B}_1$ are linearly independent (and similarly for $\mathbf{B}_2$), the matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ are unique and can be obtained by multiplying by the right inverses (and therefore Moore-Penrose Pseudoinverses) $\mathbf{B}_1^\dagger, \mathbf{B}_2^\dagger$, so that we can set:

$$\mathbf{A}_1 = \mathbf{B}_*\mathbf{B}_1^\dagger$$
$$\mathbf{A}_2 = \mathbf{B}_*\mathbf{B}_2^\dagger$$

Then we can minimize Σ_* over the Loewner partial order subject to the constraints $\Sigma_*\succeq\mathbf{A}_1\mathbf{A}_1^T$ and $\Sigma_*\succeq\mathbf{A}_2\mathbf{A}_2^T$. This will ensure that we get a maximally common mechanism because (1) any other common mechanism can be rewritten so that its query matrix is also $\mathbf{B}_*$ (by Lemma B.1) and its covariance matrix $\Sigma_*{}'$ must also be larger than $\mathbf{A}_1\mathbf{A}_1^T$ and $\mathbf{A}_1\mathbf{A}_1^T$ in the Loewner order; (2) if this other mechanism with covariance $\Sigma_*{}'$ answers the one with covariance Σ_* then by Lemma 3.2 we must have $\Sigma_*\succeq\Sigma_*{}'$, (3) but by minimality of Σ_*, we therefore have $\Sigma_* = \Sigma_*{}'$.

However, the Loewner partial order means that there could be multiple minimal solutions, and so we are free to pick one by using an objective function that respects the Loewner partial order. One such objective function is trace(Σ) and one can easily see that $\Sigma_a\preceq\Sigma_b$ implies trace$(\Sigma_a)\leq$ trace(Σ_b) since:

$$\text{trace}(\Sigma_b - \Sigma_a) = \sum_i \mathbf{e}_i^T(\Sigma_b - \Sigma_a)\mathbf{e}_i$$

where $\mathbf{e}_i$ has a 1 in position i and 0 everywhere else

$$\geq 0 \text{ by the Loewner order on } \Sigma_a \text{ and } \Sigma_b$$

Putting all these facts together results in the optimization problem in Equation 1. □

Theorem 5.4. *[Stott [61–63]] The solution to the optimization problem in Equation 1 is $\Sigma_* = \frac{\mathbf{A}_1\mathbf{A}_1^T+\mathbf{A}_2\mathbf{A}_2^T}{2} + \frac{|\mathbf{A}_2\mathbf{A}_2^T-\mathbf{A}_1\mathbf{A}_1^T|}{2}$, where $\mathbf{A}_1 = \mathbf{B}_*\mathbf{B}_1^\dagger$, $\mathbf{A}_2 = \mathbf{B}_*\mathbf{B}_2^\dagger$, and $|\cdot|$ is the operator that replaces negative eigenvalues with positive eigenvalues (i.e., if the eigendecomposition of $\mathbf{V} = \mathbf{P}^T Diag(\lambda)\mathbf{P}$ then $|\mathbf{V}| = \mathbf{P}^T Diag(|\lambda|)\mathbf{P}$).*

Theorem 5.5. *Let $\mathcal{M}_1(\mathbf{x}) = \mathbf{B}_1\mathbf{x} + N(\mathbf{0},\Sigma_1)$ and $\mathcal{M}_2(\mathbf{x}) = \mathbf{B}_2\mathbf{x} + N(\mathbf{0},\Sigma_2)$ be linear Gaussian mechanisms. Let $\mathcal{M}_*(\mathbf{x}) = \mathbf{B}_*\mathbf{x}+N(\mathbf{0},\Sigma_*)$ be their maximally common mechanism and let $\mathcal{M}_1'(\mathbf{x}) = \mathbf{B}_1'\mathbf{x} + N(\mathbf{0},\Sigma_1')$ be the residual mechanism for $\mathcal{M}_1$. Define:*

- *$\Sigma_1^{1/2}$ to be the symmetric matrix square root of Σ_1,*
- *$\Sigma_1^{-1/2}$ to be the inverse of $\Sigma_1^{1/2}$,*
- *$\mathbf{A}_* = \Sigma_1^{1/2}(\mathbf{B}_1^T\Sigma_1^{-1/2})^\dagger\mathbf{B}_*{}^T\Sigma_*{}^{-1}$, where † is the Moore-Penrose pseudo-inverse,*
- *$\mathbf{A}_1' = \Sigma_1^{1/2}(\mathbf{B}_1^T\Sigma_1^{-1/2})^\dagger\mathbf{B}_1'^T(\Sigma_1')^{-1}$.*

Then $\mathcal{M}_1(\mathbf{x}) = \mathbf{A}_*\mathcal{M}_*(\mathbf{x}) + \mathbf{A}_1'\mathcal{M}_1'(\mathbf{x}) + N(\mathbf{0},\Sigma_1 - \mathbf{A}_*\Sigma_*\mathbf{A}_*^T - \mathbf{A}_1'\Sigma_1'\mathbf{A}_1'^T)$ and $\mathcal{M}_1(\mathbf{x})$ is equivalent to $\mathbf{A}_*\mathcal{M}_*(\mathbf{x}) + \mathbf{A}_1'\mathcal{M}_1'(\mathbf{x})$.

Proof of Theorem 5.5. **Step 1**. First we need to show that for any $\mathbf{x}$, $E[\mathcal{M}_1(\mathbf{x})] = \mathbf{A}_*E[\mathcal{M}_*(\mathbf{x})] + \mathbf{A}_1'E[\mathcal{M}_1'(\mathbf{x})]$, which is the same as showing $\mathbf{B}_1 = \mathbf{A}_*\mathbf{B}_* + \mathbf{A}_1'\mathbf{B}_1'$.

Express $\mathbf{B}_1^T\Sigma_1^{-1/2}$ in terms of its singular value decomposition:

$$\mathbf{B}_1^T\Sigma_1^{-1/2} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

where $\mathbf{D}$ is an invertible $m\times m$ diagonal matrix (for some m), $\mathbf{U}$ is an $d\times m$ matrix with orthogonal columns such that $\mathbf{U}^T\mathbf{U} = \mathbf{I}_m$, and $\mathbf{V}$ is a $d\times m$ orthogonal matrix, so that $\mathbf{V}^T\mathbf{V} = \mathbf{I}_m$. Then we have:

$$(\mathbf{B}_1^T\Sigma_1^{-1/2})^\dagger = \mathbf{V}\mathbf{D}^{-1}\mathbf{U}^T$$

Recall that both common and residual mechanisms are constructed so that: $\mathbf{B}_*{}^T\Sigma_*{}^{-1}\mathbf{B}_* + (\mathbf{B}_1')^T(\Sigma_1')^{-1}\mathbf{B}_1' = \mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1$, we have

$$\begin{aligned}
\mathbf{A}_*\mathbf{B}_* + \mathbf{A}_1'\mathbf{B}_1' &= \Sigma_1^{1/2}(\mathbf{B}_1^T\Sigma_1^{-1/2})^\dagger\left(\mathbf{B}_*{}^T\Sigma_*{}^{-1}\mathbf{B}_* + \mathbf{B}_1'^T(\Sigma_1')^{-1}\mathbf{B}_1'\right)\\
&= \Sigma_1^{1/2}(\mathbf{B}_1^T\Sigma_1^{-1/2})^\dagger\left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1\right)\\
&= \Sigma_1^{1/2}(\mathbf{B}_1^T\Sigma_1^{-1/2})^\dagger\left(\mathbf{B}_1^T\Sigma_1^{-1/2}\right)\left(\Sigma_1^{-1/2}\mathbf{B}_1\right)\\
&= \Sigma_1^{1/2}\mathbf{V}\mathbf{D}^{-1}\mathbf{U}^T\mathbf{U}\mathbf{D}\mathbf{V}^T\mathbf{V}\mathbf{D}\mathbf{U}^T\\
&= \Sigma_1^{1/2}\mathbf{V}\mathbf{D}^{-1}\mathbf{D}\mathbf{D}\mathbf{U}^T\\
&= \Sigma_1^{1/2}\mathbf{V}\mathbf{D}\mathbf{U}^T\\
&= \Sigma_1^{1/2}\left(\Sigma_1^{-1/2}\mathbf{B}_1\right)\\
&= \mathbf{B}_1
\end{aligned}$$

Step 2: Now, by Lemma 3.2 we need to show that the covariance matrix of $\mathbf{A}_*E[\mathcal{M}_*(\mathbf{x})] + \mathbf{A}_1'E[\mathcal{M}_1'(\mathbf{x})]$ is $\preceq \Sigma_1$ in the Loewner partial order. This covariance matrix is:

$$\begin{aligned}
&\mathbf{A}_*\Sigma_*\mathbf{A}_*^T + \mathbf{A}_1'\Sigma_1'\mathbf{A}_1'^T\\
&= \Sigma_1^{1/2}(\mathbf{B}_1^T\Sigma_1^{-1/2})^\dagger\Big(\mathbf{B}_*{}^T\Sigma_*{}^{-1}\Sigma_*\Sigma_*{}^{-1}\mathbf{B}_*\\
&\qquad + \mathbf{B}_1'^T(\Sigma_1')^{-1}\Sigma_1'(\Sigma_1')^{-1}\mathbf{B}_1'\Big)(\mathbf{B}_1^T\Sigma_1^{-1/2})^{\dagger T}\Sigma_1^{1/2}\\
&= \Sigma_1^{1/2}(\mathbf{B}_1^T\Sigma_1^{-1/2})^\dagger\left(\mathbf{B}_*{}^T\Sigma_*{}^{-1}\mathbf{B}_* + \mathbf{B}_1'^T(\Sigma_1')^{-1}\mathbf{B}_1'\right)(\mathbf{B}_1^T\Sigma_1^{-1/2})^{\dagger T}\Sigma_1^{1/2}\\
&= \Sigma_1^{1/2}(\mathbf{B}_1^T\Sigma_1^{-1/2})^\dagger\left(\mathbf{B}_1^T\Sigma_1^{-1}\mathbf{B}_1\right)(\mathbf{B}_1^T\Sigma_1^{-1/2})^{\dagger T}\Sigma_1^{1/2}\\
&= \Sigma_1^{1/2}\mathbf{V}\mathbf{D}^{-1}\mathbf{U}^T\left(\mathbf{U}\mathbf{D}\mathbf{V}^T\mathbf{V}\mathbf{D}\mathbf{U}^T\right)\mathbf{U}\mathbf{D}^{-1}\mathbf{V}^T\Sigma_1^{1/2}\\
&= \Sigma_1^{1/2}\mathbf{V}\mathbf{D}^{-1}\mathbf{D}\mathbf{D}\mathbf{D}^{-1}\mathbf{V}^T\Sigma_1^{1/2}\\
&= \Sigma_1^{1/2}\mathbf{V}\mathbf{V}^T\Sigma_1^{1/2}\\
&= \Sigma_1^{1/2}\left(\sum_{i=1}^{m}\mathbf{v}_i\mathbf{v}_i^T\right)\Sigma_1^{1/2}
\end{aligned}$$

(where the $\mathbf{v}_i$ are the orthogonal columns of $\mathbf{V}$)

$$\preceq\Sigma_1^{1/2}\left(\sum_{i=1}^{m}\mathbf{v}_i\mathbf{v}_i^T\right)\Sigma_1^{1/2} + \Sigma_1^{1/2}\left(\sum_{i=m+1}^{d}\mathbf{v}_i'\mathbf{v}_i'^T\right)\Sigma_1^{1/2}$$

(where the $\mathbf{v}_i'$ are orthogonal and complete the bases for $\mathbb{R}^d$)

$$= \Sigma_1^{1/2}\Sigma_1^{1/2} = \Sigma_1$$

Step 3: The equivalence claimed in the theorem follows from the fact that $\mathcal{M}_1$ is answerable from $\mathbf{A}_*\mathcal{M}_* + \mathbf{A}_1'\mathcal{M}_1'$ (which was just proven) and $(\mathcal{M}_*, \mathcal{M}_1')$ is answerable from $\mathcal{M}_1$, by definition of common and residual mechanism, and $\mathbf{A}_*\mathcal{M}_*+\mathbf{A}_1'\mathcal{M}_1'$ is answerable from $(\mathcal{M}_*, \mathcal{M}_1')$. □