# Relaxed Marginal Consistency for Differentially Private Query Answering

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Many differentially private algorithms for answering database queries involve a step that reconstructs a discrete data distribution from noisy measurements. This provides consistent query answers and reduces error, but often requires space that grows exponentially with dimension. PRIVATE-PGM is a recent approach that uses graphical models to represent the data distribution, with complexity proportional to that of exact marginal inference in a graphical model with structure determined by the co-occurrence of variables in the noisy measurements. PRIVATE-PGM is highly scalable for sparse measurements, but may fail to run in high dimensions with dense measurements. We overcome the main scalability limitation of PRIVATE-PGM through a principled approach that relaxes consistency constraints in the estimation objective. Our new approach works with many existing private query answering algorithms and improves scalability or accuracy with no privacy cost.

## 1 Introduction

A central problem in the design differentially private algorithms is answering complex sets of counting queries from a database. Many proposed algorithms follow the select-measure-reconstruct paradigm: they *select* a set of measurement queries, they privately *measure* them (using Gaussian or Laplace noise addition), and then they *reconstruct* the data or query answers from the noisy measurements. When done in a principled manner, the reconstruct phase serves a number of critical functions: it combines the noisy evidence provided by the measurement queries, it allows new unmeasured queries to be answered (with no additional privacy cost), and it resolves inconsistencies in the noisy measurements to produce consistent estimates, which often have lower error. In this paper, we propose a novel, scalable, and general-purpose approach to the reconstruct step. With a principled approach to this problem, future research can focus on the challenging open problem of query selection.

Most existing *general-purpose* methods for reconstruction cannot scale to high-dimensional data, as they operate over a vectorized representation of the data, whose size is exponential in the dimensionality [1–6]. Some special purpose methods exist that have better scalability, but are only applicable within a particular mechanism or in certain special cases [7–11, 5, 12–14]. A recently-proposed method, PRIVATE-PGM [15], offers the scalability of these special purpose methods and retains much of the generality of the general-purpose methods. PRIVATE-PGM can be used for the reconstruction phase whenever the measurements only depend on the data through its low-dimensional marginals. PRIVATE-PGM avoids the data vector representation in favor of a more compact graphical model representation, and was shown to dramatically improve the scalability of a number of popular mechanisms while also improving accuracy [15]. PRIVATE-PGM was used in the winning entry of the 2018 NIST differential privacy contest [16].

While PRIVATE-PGM is far more scalable than operating over a vector representation of the data, it is still limited. In particular, its required memory and runtime depend on the structure of the

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

underlying graphical model, which is defined by co-occurrence of variables in marginals selected for measurement by the privacy mechanism. When the graph is sparse, PRIVATE-PGM is extremely efficient. But, with more measurements, the graphical model typically becomes dense and it eventually becomes intractable for PRIVATE-PGM to run. This is due to the inherent hardness of exact marginal inference in a graphical model.

In this paper, we overcome the scalability limitations of PRIVATE-PGM by proposing a natural relaxation of the estimation objective that enforces specified *local* consistency constraints among marginals, instead of global ones, and can be solved efficiently. Our technical contributions may be of broader interest. We develop an efficient algorithm to solve a generic convex optimization problem over the local polytope of a graphical model, which uses a body of prior work on generalized belief propagation [17–27] and can scale to problems with millions of optimization variables. We also propose a variational approach to predict "out-of-model" marginals given estimated pseudo-marginals, which gives a completely variational formulation for both estimation and inference: the results are invariant to optimization details, including the approximate inference methods used as subroutines.

Our new approach, APPROX-PRIVATE-PGM (APPGM), offers many of the same benefits as PRIVATE-PGM, but can be deployed in far more settings, allowing effective reconstruction to be performed without constraining the selected measurements. We show that APPGM permits efficient reconstruction for the state-of-the-art HDMM [28] algorithm, with little impact on accuracy, allows the popular MWEM [5] algorithm to scale to far more measurements, and results in better accuracy for the recent FEM algorithm [29].

## 2 Background

We first review background on our data model, marginals, and differential privacy, following [15].

**Data** Our input data represents a population of individuals, each contributing a single record $\mathbf{x} = (x_1, \ldots, x_d)$ where $x_i$ is the $i^{th}$ attribute belonging to a discrete finite domain $\Omega_i$ of $n_i$ possible values. The full domain is $\Omega = \prod_{i=1}^d \Omega_i$ and its size $n = \prod_{i=1}^d n_i$ is exponential in the number of attributes. A dataset $\mathbf{X}$ consists of $m$ such records $\mathbf{X} = (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)})$.

It is often convenient to work with an alternate representation of $\mathbf{X}$: the *data vector* or *data distribution* $\mathbf{p}$ is a vector of length $n$, indexed by $\mathbf{x} \in \Omega$ such that $\mathbf{p}(\mathbf{x})$ counts the fraction of individuals with record equal to $\mathbf{x}$. That is, $\mathbf{p}(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^m \mathbb{I}\{\mathbf{x}^{(i)} = \mathbf{x}\}, \forall \mathbf{x} \in \Omega$, where $\mathbb{I}\{\cdot\}$ is an indicator function. Note that the vector $m\mathbf{p}$ is sometimes called the full contingency table for the data set.

**Marginals** When dealing with high-dimensional data, it is common to work with *marginals* defined over a subset of attributes. Let $r \subseteq [d]$ be a *region* or *clique* that identifies a subset of attributes and, for $\mathbf{x} \in \Omega$, let $\mathbf{x}_r = (x_i)_{i \in r}$ be the sub-vector of $\mathbf{x}$ restricted to $r$. Then the marginal vector (or simply "marginal on $r$") $\boldsymbol{\mu}_r$, is defined by:

$$\boldsymbol{\mu}_r(\mathbf{x}_r) = \frac{1}{m}\sum_{i=1}^m \mathbb{I}\{\mathbf{x}_r^{(i)} = \mathbf{x}_r\}, \quad \forall \mathbf{x}_r \in \Omega_r := \prod_{i \in r} \Omega_i. \tag{1}$$

This marginal is the data vector on the sub-domain $\Omega_r$ corresponding to the attribute set $r$. Its size is $n_r := |\Omega_r| = \prod_{i \in r} n_i$, which is exponential in $|r|$ but may be considerably smaller than $n$. A marginal can be computed from the full data vector or the marginal for any superset of attributes by summing over variables that are not in $s$. We denote these (linear) operations by $M_r$ and $P_{s \to r}$, so $\boldsymbol{\mu}_r = M_r \mathbf{p} = P_{s \to r} \boldsymbol{\mu}_s$ for $r \subseteq s$. We will also consider vectors $\boldsymbol{\mu}$ that combine marginals for each region in a collection $\mathcal{C}$, and let $M_{\mathcal{C}}$ be the linear operator such that $\boldsymbol{\mu} = (\boldsymbol{\mu}_r)_{r \in \mathcal{C}} = M_{\mathcal{C}}\mathbf{p}$.

**Differential Privacy** Differential privacy protects individuals by bounding the impact any one individual can have on the output of an algorithm.

**Definition 1** (Differential Privacy; Dwork et al., 2006)**.** *A randomized algorithm $\mathcal{A}$ satisfies $(\epsilon, \delta)$-differential privacy if, for any input $\mathbf{X}$, any $\mathbf{X}' \in$ nbrs$(\mathbf{X})$, and any subset of outputs $S \subseteq$ Range$(\mathcal{A})$,*

$$\Pr[\mathcal{A}(\mathbf{X}) \in S] \leq \exp(\epsilon) \Pr[\mathcal{A}(\mathbf{X}') \in S] + \delta$$

Above, nbrs$(\mathbf{X})$ denotes the set of datasets formed by replacing any $\mathbf{x}^{(i)} \in \mathbf{X}$ with an arbitrary new record $\mathbf{x}'^{(i)} \in \Omega$. When $\delta = 0$ we say $\mathcal{A}$ satisfies $\epsilon$-differential privacy.

# 3 Private-PGM

In this section we describe PRIVATE-PGM [15], a general-purpose reconstruction method applied to differentially private measurements of a discrete dataset. There are two steps to PRIVATE-PGM: (1) *estimate* a representation of the data distribution given noisy measurements, and (2) *infer* answers to new queries given the data distribution representation.

In particular, suppose an arbitrary $(\epsilon, \delta)$-differentially private algorithm $\mathcal{A}$ is run on a discrete dataset with data vector $\mathbf{p}_0$, where $\mathcal{A}$ only depends on $\mathbf{p}_0$ through its low-dimensional marginals $\boldsymbol{\mu}_0 = M_{\mathcal{C}}\mathbf{p}_0$ for some collection of cliques $\mathcal{C}$. The sample $\mathbf{y} \sim \mathcal{A}(\boldsymbol{\mu}_0)$ typically reveals noisy high-level aggregate information about the data. PRIVATE-PGM will first estimate a compact representation of a distribution $\hat{\mathbf{p}}$ that explains $\mathbf{y}$ well, and then answer new queries using $\hat{\mathbf{p}}$.

**Estimation: Finding a Data Distribution Representation** PRIVATE-PGM first estimates a data vector by finding $\hat{\mathbf{p}}$ to solve the inverse problem $\min_{\mathbf{p}} L(M_{\mathcal{C}}\mathbf{p})$, where $L(\boldsymbol{\mu})$ is a convex loss function that measures how well $\boldsymbol{\mu}$ explains the observations $\mathbf{y}$. Since $L(M_{\mathcal{C}}\mathbf{p})$ only depends on $\mathbf{p}$ through its marginals, it is clear we can find the optimal *marginals* by instead solving the following problem.

**Problem 1** (Convex Optimization over the Marginal Polytope)**.** *Given a clique set $\mathcal{C}$ and convex loss function $L(\boldsymbol{\mu})$, solve*

$$\hat{\boldsymbol{\mu}} \in \operatorname*{argmin}_{\boldsymbol{\mu} \in \mathcal{M}(\mathcal{C})} L(\boldsymbol{\mu}),$$

*where $\mathcal{M}(\mathcal{C}) = \{\boldsymbol{\mu} : \exists\ \mathbf{p}$ s.t. $M_{\mathcal{C}}\mathbf{p} = \boldsymbol{\mu}\}$ is the set of realizable marginals, known as the* marginal polytope *of $\mathcal{C}$ [31].*

The solution to this problem gives marginals that are *consistent* with some underlying data vector and, therefore, typically provide a better estimate of the true marginals than $\mathbf{y}$. In the general case, the loss function $L$ can simply be set to the *negative log likelihood*, i.e., $L(\boldsymbol{\mu}) = -\log \Pr[\mathcal{A}(\boldsymbol{\mu}) = \mathbf{y}]$,[1] however other choices are also possible. As a concrete motivating application, consider the case where the mechanism $\mathcal{A}$ adds Gaussian noise directly to the data marginals $\boldsymbol{\mu}_0$, i.e., $\mathcal{A}(\boldsymbol{\mu}_0) = \mathbf{y}$ where $\mathbf{y}_r = \boldsymbol{\mu}_{0,r} + \mathcal{N}(0, \sigma^2 I_{n_r})$. In this case, the log-likelihood is proportional to the squared Euclidean distance and gives the loss function $L(\boldsymbol{\mu}) = \|\boldsymbol{\mu} - \mathbf{y}\|_2^2$, so the problem at hand is an $L_2$ minimization problem. The theory for PRIVATE-PGM focuses on convex loss functions, but the algorithms are also used to seek local minima of Problem 1 when $L$ is non-convex.

**Graphical models** Two remaining issues are how to solve Problem 1 and how to recover a full data vector from $\hat{\boldsymbol{\mu}}$. PRIVATE-PGM addresses both with *graphical models*. A graphical model with clique set $\mathcal{C}$ is a distribution over $\Omega$ where the unnormalized probability is a product of factors involving only subsets of variables, one for each clique in $\mathcal{C}$. It has the form

$$\mathbf{p}_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_{r \in \mathcal{C}} \boldsymbol{\theta}_r(\mathbf{x}_r)\right).$$

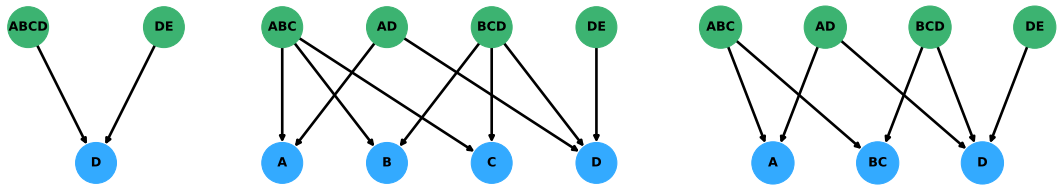
**Algorithm 1** PROX-PGM [15]

**Input:** Convex loss function $L(\boldsymbol{\mu})$
**Output:** Marginals $\hat{\boldsymbol{\mu}}$, parameters $\hat{\boldsymbol{\theta}}$
$\hat{\boldsymbol{\theta}} = \mathbf{0}$
**for** $t = 1, \ldots, T$ **do**
  $\hat{\boldsymbol{\mu}} =$MARGINAL-ORACLE$(\hat{\boldsymbol{\theta}})$
  $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} - \eta_t \nabla L(\hat{\boldsymbol{\mu}})$
**return** $\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\theta}}$

The real numbers $\boldsymbol{\theta}_r(\mathbf{x}_r)$ are *log-potentials* or *parameters*. The full parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_r(\mathbf{x}_r))_{r \in \mathcal{C}, \mathbf{x}_r \in \Omega_r}$ matches the marginal vector $\boldsymbol{\mu}$ in size and indexing, and the relationship between these two vectors is central to graphical models [31]:

- A parameter vector $\boldsymbol{\theta}$ determines a unique marginal vector $\boldsymbol{\mu}_{\boldsymbol{\theta}} \in \mathcal{M}(\mathcal{C})$, defined as $\boldsymbol{\mu}_{\boldsymbol{\theta}} = M_{\mathcal{C}}\mathbf{p}_{\boldsymbol{\theta}}$, the marginals of $\mathbf{p}_{\boldsymbol{\theta}}$. *Marginal inference* is the problem of (efficiently) computing $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ from $\boldsymbol{\theta}$. It can be solved exactly by algorithms such as *variable elimination* or *belief propagation* with a junction tree [32]. We denote by MARGINAL-ORACLE an algorithm that outputs $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ on input $\boldsymbol{\theta}$.
- For every $\boldsymbol{\mu} \in \mathcal{M}(\mathcal{C})$ with positive entries, there is a unique distribution $\mathbf{p}_{\boldsymbol{\theta}}$ in the family of graphical models with cliques $\mathcal{C}$ that has marginals $\boldsymbol{\mu}$, and $\mathbf{p}_{\boldsymbol{\theta}}$ has *maximum entropy among all distributions with marginals $\boldsymbol{\mu}$*.

[1] For mechanisms with continuous output values, interpret this as a negative log-density.




(a) Junction Tree (b) Factor Graph (c) Saturated Region Graph

Figure 1: Comparison of different region graph structures defined over a domain with attributes $\{A, B, C, D, E\}$ that support the cliques $\mathcal{C} = \{\{A, B, C\}, \{A, D\}, \{B, C, D\}, \{D, E\}\}$.

PROX-PGM (Algorithm 1) is a proximal algorithm to solve Problem 1 [15]. It returns a marginal vector $\hat{\boldsymbol{\mu}} \in \mathcal{M}(\mathcal{C})$ that minimizes $L(\boldsymbol{\mu})$ *and* parameters $\hat{\boldsymbol{\theta}}$ such that $\mathbf{p}_{\hat{\boldsymbol{\theta}}}$ has marginals $\hat{\boldsymbol{\mu}}$. The core of the computation is repeated calls to MARGINAL-ORACLE. The estimated graphical model $\mathbf{p}_{\hat{\boldsymbol{\theta}}}$ has cliques $\mathcal{C}$ that coincide with the marginals measured by the privacy mechanism, and has maximum entropy among all distributions whose marginals minimize $L(\boldsymbol{\mu})$. In general, there will be infinitely many distributions that have marginals $\hat{\boldsymbol{\mu}}$ (and hence are equally good from the perspective of the loss function $L$). PRIVATE-PGM chooses the distribution with maximum entropy, which is an appealing way to break ties that falls out naturally from the graphical model.

**Inference: Answering New Queries** With $\mathbf{p}_{\hat{\boldsymbol{\theta}}}$ in hand, PRIVATE-PGM can readily estimate new marginals $\boldsymbol{\mu}_r$. There are two separate cases. If $r$ is contained in a clique of $C$, we say $r$ is "in-model", and we can readily calculate $\boldsymbol{\mu}_r$ from the output of PROX-PGM. The more interesting case occurs when $r$ is out-of-model (is not contained in a clique of $\mathcal{C}$): in this case, the standard way to compute $\boldsymbol{\mu}_r$ is to perform variable elimination in the graphical model $\mathbf{p}_{\boldsymbol{\theta}}$.

**Remark 1.** *The complexity of* PRIVATE-PGM *depends on that of* MARGINAL-ORACLE*, which depends critically on the structure of the cliques $\mathcal{C}$ measured by the privacy mechanism. In general, running time is exponential in the treewidth of the graph $G$ induced by attribute co-occurence within a clique of $\mathcal{C}$. When $G$ is tree-like,* PRIVATE-PGM *can be highly efficient and exponentially faster than working with a full data vector. When $G$ is dense,* PRIVATE-PGM *may fail to run due to time or memory constraints. This limitation is not specific to the* PROX-PGM *algorithm: Problem 1 is as hard as marginal inference, which can be solved by minimizing the (convex) variational free energy over the marginal polytope [32]. A primary difficulty is the intractability of $\mathcal{M}(\mathcal{C})$, which is a convex set, but in general requires a very large number of constraints to represent explicitly [31].*

# 4 Our Approach

In this section we describe our approach to overcome the main scalability limitations of PRIVATE-PGM, by introducing suitable approximations and new algorithmic techniques. Our innovations allow us to scale significantly better than PRIVATE-PGM with respect to the size of $\mathcal{C}$. A high-level idea is to use approximate marginal inference in the PROX-PGM algorithm, but doing so naively would make it unclear what, if any, formal problem is being solved. We will develop a principled approach that uses approximate inference to *exactly* solve a relaxed optimization problem.

**Region Graphs** Central to our approach is the notion of a *region graph*, which is a data structure that encodes constraints between cliques in a natural graphical format and facilitates message passing algorithms for approximate marginal inference.

**Definition 2** (Region Graph)**.** *A region graph $G = (\mathcal{V}, \mathcal{E})$ is a directed graph where every vertex $r \in \mathcal{V}$ is an attribute clique and for any edge $r \to s \in \mathcal{E}$ we have that $r \supseteq s$. We say that $G$ supports a clique set $\mathcal{C}$ if for every clique $r \in \mathcal{C}$, there exists some $r' \in \mathcal{V}$ such that $r \subseteq r'$.*

For any region graph, there is a corresponding set of constraints that characterize the *local polytope* of internally consistent *pseudo-marginals*, defined below.

**Definition 3** (Local Polytope)**.** *The local polytope of pseudo-marginals associated with a region graph $G = (\mathcal{V}, \mathcal{E})$ is:*

$$\mathcal{L}(G) = \left\{ \boldsymbol{\tau} \geq 0 \;\middle|\; \begin{array}{ll} \mathbf{1}^\top \boldsymbol{\tau}_r = 1 & \forall r \in \mathcal{V} \\ P_{r \to s} \boldsymbol{\tau}_r = \boldsymbol{\tau}_s & \forall r \to s \in \mathcal{E} \end{array} \right\}. \tag{2}$$

The nodes in the region graph correspond to the cliques in the pseudo-marginal vector, while the edges in the region graph dictate which internal consistency constraints we expect to hold between two cliques. These constraints are necessary, but not sufficient, for a given set of pseudo-marginals to be realizable, i.e., $\mathcal{M}(\mathcal{V}) \subseteq \mathcal{L}(G)$. In the special case when $G$ is a junction tree, these constraints are also sufficient, and we have $\mathcal{M}(\mathcal{V}) = \mathcal{L}(G)$.We use the notation $\boldsymbol{\tau}$ in place of $\boldsymbol{\mu}$ to emphasize that $\boldsymbol{\tau}$ is not necessarily a valid marginal vector, even though we will generally treat it as such. This notational choice is standard in the graphical models literature [31]. The general idea is to relax problems involving the intractable marginal polytope to use the local polytope instead, since $\mathcal{L}(G)$ is straightforward to characterize using the linear constraints in Equation (2).

Region graphs can encode different structures, including junction trees and factor graphs as special cases. For example, Figure 1 shows three different region graphs that support $\mathcal{C} = \{\{A, B, C\}, \{A, D\}, \{B, C, D\}, \{D, E\}\}$. At one extreme is the Junction Tree, shown in Figure 1a, which is obtained by merging cliques $\{A, B, C\}$, $\{A, D\}$, and $\{B, C, D\}$ into a super-clique $\{A, B, C, D\}$. Here, $\mathcal{M}(\mathcal{V}) = \mathcal{L}(G)$, and ordinary belief propagation in this graph corresponds to exact marginal inference. At the other end of the extreme is the Factor Graph, shown in Figure 1b. This graph contains one vertex for every clique $r \in \mathcal{C}$, plus additional vertices for the singleton cliques. It encodes constraints that all cliques must agree on common one-way marginals. For example, $\boldsymbol{\tau}_{ABC}$ and $\boldsymbol{\tau}_{BCD}$ must agree on the shared one-way marginals $\boldsymbol{\tau}_B$ and $\boldsymbol{\tau}_C$, but not necessarily on the shared two-way marginal $\boldsymbol{\tau}_{BC}$. A natural middle ground is the fully Saturated Region Graph, shown in Figure 1c. This graph includes every clique $r \in \mathcal{C}$ as a vertex, and includes additional vertices to capture intersections between those cliques. Unlike the factor graph, this graph *does* require that $\boldsymbol{\tau}_{ABC}$ is consistent with $\boldsymbol{\tau}_{BCD}$ with respect to the $\boldsymbol{\tau}_{BC}$ marginal. Unlike the junction tree, this graph does not require forming super-cliques whose size grow quickly with $|\mathcal{C}|$. For more details about the concepts above, please refer to [32, Section 11.3].

The methods we describe in this paper apply for any region graph that supports $\mathcal{C}$. By default, we simply use the fully saturated region graph, which is the smallest region graph that encodes all internal consistency constraints, and can easily be constructed given the cliques $\mathcal{C}$ [32].

**Estimation: Finding an Approximate Data Distribution Representation** We begin by introducing a very natural relaxation of the problem we seek to solve.

**Problem 2** (Convex Optimization over the Local Polytope)**.** *Given a region graph $G$ and a convex loss function $L(\boldsymbol{\tau})$ where $\boldsymbol{\tau} = (\boldsymbol{\tau}_r)_{r \in \mathcal{V}}$, solve:*

$$\hat{\boldsymbol{\tau}} = \operatorname*{argmin}_{\boldsymbol{\tau} \in \mathcal{L}(G)} L(\boldsymbol{\tau}).$$

In the problem above, we simply replaced the marginal polytope $\mathcal{M}(\mathcal{V})$ from our original problem[2] with the local polytope $\mathcal{L}(G)$. Since this is a convex optimization problem with linear constraints, it can be solved with a number of general purpose techniques, including interior point and active set methods [33]. However, these methods do nothing to exploit the special structure in the constraint set $\mathcal{L}(G)$, and as such, they have trouble running on large-scale problems.

Our first contribution is to show that we can solve Problem 2 efficiently by instantiating PROX-PGM with a carefully chosen approximate marginal oracle. To do so, it is useful to view approximate marginal inference through the lens of the free energy minimization problem, stated below.

**Problem 3** (Approximate Free Energy Minimization)**.** *Let $G$ be a region graph, $\boldsymbol{\theta} = (\boldsymbol{\theta}_r)_{r \in \mathcal{V}}$ be real-valued parameters, and $H_\kappa(\boldsymbol{\tau}) = \sum_{r \in \mathcal{V}} \kappa_r H(\boldsymbol{\tau}_r)$, where $\kappa_r \in \mathbb{R}$ are counting numbers, and $H(\boldsymbol{\tau}_r) = -\sum_{\mathbf{x}_r \in \Omega_r} \boldsymbol{\tau}_r(\mathbf{x}_r) \log \boldsymbol{\tau}_r(\mathbf{x}_r)$ is the Shannon entropy of $\boldsymbol{\tau}_r$, solve:*

$$\hat{\boldsymbol{\tau}} = \operatorname*{argmin}_{\boldsymbol{\tau} \in \mathcal{L}(G)} -\boldsymbol{\tau}^\top \boldsymbol{\theta} - H_\kappa(\boldsymbol{\tau})$$

[2] If $G$ supports $\mathcal{C}$, we can assume without loss of generality that Problem 1 was defined on $\mathcal{M}(\mathcal{V})$ instead of $\mathcal{M}(\mathcal{C})$. In particular, the loss function $L$ can be written to depend on marginals $(\boldsymbol{\mu}_{r'})_{r' \in \mathcal{V}}$ instead of $(\boldsymbol{\mu}_r)_{r \in \mathcal{C}}$, because the latter can be computed from the former.

This problem approximates the (intractable) variational free energy minimization problem [31], for which the optimum gives the true marginals of $\mathbf{p}_{\boldsymbol{\theta}}$, by using $\mathcal{L}(G)$ instead of $\mathcal{M}(\mathcal{V})$, and using $H_\kappa(\boldsymbol{\tau})$ as an approximation to the full Shannon entropy. Many algorithms for approximate marginal inference can be seen as solving variants of this free energy minimization problem under different assumptions about $G$ and $\kappa$ [17–27].

**Theorem 1** (Algorithm for Approximate Free Energy Minimization [21])**.** *Given a region graph $G$, parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_r)_{r\in\mathcal{V}}$, and any positive counting numbers $\kappa_r > 0$ for $r \in \mathcal{V}$, the convex generalized belief propagation (*CONVEX-GBP*) algorithm of [21] solves the approximate free energy minimization problem of Problem 3.*

CONVEX-GBP is listed in Appendix A (Algorithm 2) and is a message-passing algorithm in the region graph that uses the counting numbers as weights. Importantly, the complexity of CONVEX-GBP depends mainly on the size of the largest clique in the region graph. In many cases of practical interest, this will be exponentially smaller in the saturated region graph than in a junction tree.

**Theorem 2.** *When* PROX-PGM *uses* CONVEX-GBP *as the* MARGINAL-ORACLE *(with **any** positive counting numbers $\kappa$), it solves the convex optimization problem over the local polytope of Problem 2.*

This result is remarkable in light of previous work, where different counting number schemes are used with the goal of tightly approximating the true entropy, and form the basis for different approximate inference methods. In our setting, all methods with *convex* counting numbers are equivalent: they may lead to different *parameters* $\hat{\boldsymbol{\theta}}$, but the corresponding pseudo-marginals $\hat{\boldsymbol{\tau}} = \text{CONVEX-GBP}(\hat{\boldsymbol{\theta}})$ are invariant. Indeed, the optimal $\hat{\boldsymbol{\tau}}$ depends only on the estimation objective $L(\boldsymbol{\tau})$ and the structure of the local polytope. We conjecture that a similar invariance holds for traditional marginal-based learning objectives with approximate inference [34] when message-passing algorithms based on convex free-energy approximations are used as the approximate inference method.

*Proof.* Since $L$ is a convex function and $\mathcal{L}$ is a convex constraint set, this problem can be solved with mirror descent [35]. Each iteration of mirror descent requires solving subproblems of the form:

$$\boldsymbol{\tau}^{t+1} = \operatorname*{argmin}_{\boldsymbol{\tau}\in\mathcal{L}} \boldsymbol{\tau}^\top \nabla L(\boldsymbol{\tau}^t) + \frac{1}{\eta_t} D(\boldsymbol{\tau}, \boldsymbol{\tau}^t), \qquad D(\boldsymbol{\tau}, \boldsymbol{\tau}^t) = \psi(\boldsymbol{\tau}) - \psi(\boldsymbol{\tau}^t) - (\boldsymbol{\tau} - \boldsymbol{\tau}^t)^\top \nabla\psi(\boldsymbol{\tau}^t).$$

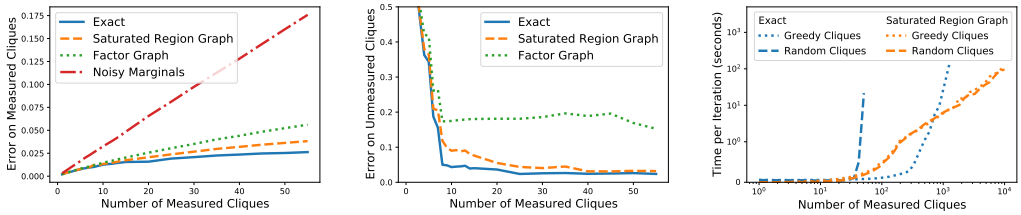
Here, $D$ is a Bregman distance measure and $\psi$ is some strongly convex and continuously differentiable function. Setting $\psi = -H_\kappa$, a negative weighted entropy with any (strongly) convex counting numbers $\kappa$, we arrive at the following update equation:

$$\begin{aligned}
\boldsymbol{\tau}^{t+1} &= \operatorname*{argmin}_{\boldsymbol{\tau}\in\mathcal{L}(G)} \boldsymbol{\tau}^\top \nabla L(\boldsymbol{\tau}^t) + \frac{1}{\eta_t}\Big( - H_\kappa(\boldsymbol{\tau}) + H_\kappa(\boldsymbol{\tau}^t) + (\boldsymbol{\tau} - \boldsymbol{\tau}^t)^\top \nabla H_\kappa(\boldsymbol{\tau}^t)\Big) \\
&= \operatorname*{argmin}_{\boldsymbol{\tau}\in\mathcal{L}(G)} \boldsymbol{\tau}^\top \Big(\eta_t \nabla L(\boldsymbol{\tau}^t) + \nabla H_\kappa(\boldsymbol{\tau}^t)\Big) - H_\kappa(\boldsymbol{\tau}) && \text{(algebraic manipulation)} \\
&= \operatorname*{argmin}_{\boldsymbol{\tau}\in\mathcal{L}(G)} \boldsymbol{\tau}^\top \Big(\eta_t \nabla L(\boldsymbol{\tau}^t) - \boldsymbol{\theta}^t\Big) - H_\kappa(\boldsymbol{\tau}) && \text{(Lemma 1; Appendix A)} \\
&= \text{CONVEX-GBP}\big(G,\, \boldsymbol{\theta}^t - \eta_t \nabla L(\boldsymbol{\tau}^t),\, \kappa\big) && \text{(Theorem 1)}
\end{aligned}$$

□

**Inference: Answering New Queries** We now turn our attention to the problem of inference. The central challenge is to estimate out-of-model marginals. Let $\hat{\boldsymbol{\tau}}$ and $\hat{\boldsymbol{\theta}}$ be the estimated pseudo-marginals and corresponding parameters after running PROX-PGM with CONVEX-GBP and region graph $G$. We have $\hat{\boldsymbol{\tau}} \approx \boldsymbol{\mu}_0$, and want an estimate $\hat{\boldsymbol{\tau}}_r \approx \boldsymbol{\mu}_{0,r}$ where $r \notin \mathcal{V}$.

CONVEX-GBP is the mapping such that $\hat{\boldsymbol{\tau}} = \text{CONVEX-GBP}(\hat{\boldsymbol{\theta}}) \approx \boldsymbol{\mu}_0$. Thus, it is appropriate to use CONVEX-GBP with estimated parameters $\hat{\boldsymbol{\theta}}$ as the basis for estimating new pseudo-marginals. This requires selecting an expanded region graph $G'$ that supports $r$ and new counting number $\kappa_r$. In Appendix B, we analyze this approach for an idealized setting and find that it leads to estimates $\hat{\boldsymbol{\tau}}_r$ that maximize the entropy $H(\hat{\boldsymbol{\tau}}_r)$ subject to $\hat{\boldsymbol{\tau}}_r$ being consistent with $\hat{\boldsymbol{\tau}}$ on overlapping marginals. However, there are two practical difficulties with the idealized setting. First, there may be *no* $\hat{\boldsymbol{\tau}}_r$ that is consistent with $\hat{\boldsymbol{\tau}}$ on overlapping marginals: this is because $\hat{\boldsymbol{\tau}}$ satisfies only local consistency contraints. Second, the idealized case uses $\kappa_r$ very close to zero, and CONVEX-GBP performs poorly in this case. Instead, we design an optimization algorithm to mimic the idealized setting:




(a) In-Model Marginals (b) Out-of-Model Marginals (c) Scalability

Figure 2: Comparison between PROX-PGM with exact and approximate inference (for different region graph structures): (a) error on in-model (measured) marginals, (b) error on out-of-model (unmeasured) marginals, and (c) scalability of PROX-PGM vs. number of measured marginals.

**Problem 4** (Maximize Entropy Subject to Minimizing Constraint Violation)**.** *Let* $\hat{\boldsymbol{\tau}} = (\hat{\boldsymbol{\tau}}_u)_{u \in \mathcal{V}}$ *and let* $r \notin \mathcal{V}$*. Solve:*

$$\max_{\hat{\boldsymbol{\tau}}_r} H(\hat{\boldsymbol{\tau}}_r) \text{ subject to } \hat{\boldsymbol{\tau}}_r \in \operatorname*{argmin}_{\boldsymbol{\tau}_r \in \mathcal{S}} \sum_{u \in \mathcal{V}, s = u \cap r} \|P_{r \to s} \boldsymbol{\tau}_r - P_{u \to s} \hat{\boldsymbol{\tau}}_u\|_2^2.$$

This relaxes the constraint that $\hat{\boldsymbol{\tau}}_r$ agrees with $\hat{\boldsymbol{\tau}}$ on overlapping marginals, to instead minimize the $L_2$ constraint violation. The inner problem is a quadratic minimization problem over the probability simplex $\mathcal{S}$. We show in Appendix B that a maximizer of Problem 4 is obtained by solving the inner problem once using entropic mirror descent [35].

The advantages of Problem 4 are that it is low-dimensional, only requires information from $\hat{\boldsymbol{\tau}}$ about attributes that are in $r$, can be solved much more quickly than running CONVEX-GBP, and can be solved in parallel for different marginals $r, r'$. This also gives a *fully* variational approach: both estimation and inference are fully defined through convex optimization problems that can be solved efficiently, and whose solutions are invariant to details of the approximate inference routines such as counting numbers.

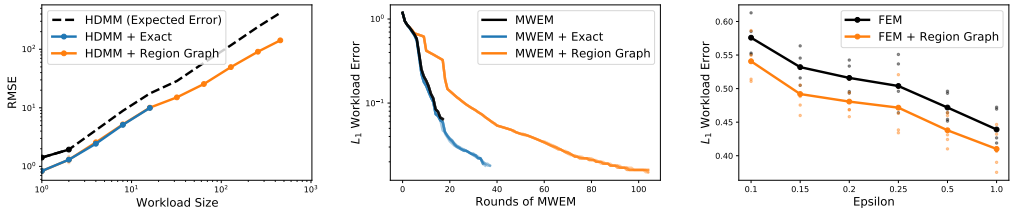
## 5 Experiments

**Comparison to PRIVATE-PGM in a Simple Mechanism** We begin by comparing the accuracy and scalability of APPGM and PRIVATE-PGM for estimating a fixed workload of marginal queries from noisy measurements of those marginals made by the Laplace mechanism. We use synthetic data to control the data domain and distribution (details in Appendix C.1) and measure $k$ different 3-way marginals with $\epsilon = 1$, for different settings of $k$. We run PROX-PGM with different versions of MARGINAL-ORACLE: Exact, Saturated Region Graph, and Factor Graph, where the first corresponds to PRIVATE-PGM, and the latter two to APPGM with the corresponding region graph (Figure 1).

*Accuracy.* We first show that when exact inference is tractable, some accuracy is lost by using approximate inference, but the estimated pseudo-marginals are much better than the noisy ones. We use eight-dimensional data with $n_1 = \cdots = n_8 = 4$, which is small enough so exact inference is always tractable, and measure random 3-way marginals for $k$ from 1 to $\binom{8}{3} = 56$. We then run 10000 iterations of PROX-PGM using differing choices for MARGINAL-ORACLE, and report the $L_1$ error on in- and out-of-model marginals, averaged over all cliques and across five trials.

In Figure 2a, we see that the error of all PROX-PGM variants is always lower than the error of the noisy measurements themselves. Exact (PRIVATE-PGM) always has lowest error, followed by Saturated Region Graph, then Factor Graph. This matches expectations: richer region graph structures encode more of the actual constraints and hence provide better error. The trend is similar for out-of-model marginals (Figure 2b). Factor Graph, which only enforces consistency with respect to the one-way marginals, performs poorly on unmeasured cliques, while Saturated Region Graph and Exact, which enforce more constraints, do substantially better. The difference between Saturated Region Graph and Exact is smaller, but meaningful.

*Scalability.* Next we consider high-dimensional data and compare the scalability of Exact and Saturated Region Graph on 100-dimensional data with $n_1 = \cdots = n_{100} = 10$. We vary $k$ from 1




(a) HDMM – FIRE (b) MWEM – FIRE (c) FEM – ADULT

Figure 3: Three examples of using PROX-PGM to improve scalability and accuracy of other algorithms.

to $10^4$ and calculate the per-iteration time of PROX-PGM.[3] We consider two schemes for selecting measured cliques: *random* selects triples of attributes uniformly at random, and *greedy* selects triples to minimize the junction tree size in each iteration. As shown in Figure 2c, Exact can handle about 50 random measured cliques or 1000 greedy ones before the per-iteration time becomes too expensive (the growth rate is exponential). In contrast, Saturated Region Graph runs with 10000 measured cliques for either strategy and could run on larger cases (the growth rate is linear).

**Improving Scalability and Accuracy in Sophisticated Mechanisms** We show next how PROX-PGM can be used to improve the performance of two sophisticated mechanisms for answering complex workloads of linear queries, including marginals. HDMM [28] is a state-of-the-art algorithm that first selects a "strategy" set of (weighted) marginal queries to be measured, and then reconstructs answers to workload queries. MWEM [5] is another competitive mechanism that iteratively measures poorly approximated queries to improve a data distribution approximation. In each algorithm, the bottleneck in high dimensions is estimating a data distribution $\hat{\mathbf{p}}$, which is prohibitive to do explicitly. PRIVATE-PGM can extend each algorithm to run in higher dimensions [15], but still becomes infeasible with enough dimensions and measurements. HDMM is a "batch" algorithm and either can or cannot run for a particular workload. Because MWEM iteratively selects measurements, even in high dimensions it can run for some number of iterations before the graphical model structure becomes too complex. By using APPGM instead, HDMM can run for workloads that were previously impossible, and MWEM can run for any number of rounds. We use the FIRE dataset from the 2018 NIST synthetic data competition [36], which includes 15 attributes and $m \approx 300{,}000$ individuals.

For HDMM, we consider workloads of $k$ random 3-way marginals, for $k = 1, 2, 4, 8, \ldots, 256, 455$, run five trials, and report root mean squared error, the objective function that HDMM optimizes. Figure 3a shows the results. HDMM with a full data vector cannot run for $k > 2$, but we can still analytically compute the *expected error* if it were able to run. HDMM + Exact fails to run beyond $k = 16$, while HDMM + Region Graph is able to run in every setting, substantially expanding the range of settings in which HDMM can be used. Both variants offer the significant error improvements of PGM-style inference, because they impose non-negativity constraints that reduce error. For example, when $k = 455$, there is a $3\times$ reduction in RMSE.

For MWEM, we consider the workload of all 2-way marginals, use a privacy budget of $\epsilon = 0.1$ *per round*, and run for as may rounds as possible, until MWEM has measured all 2-way marginals or exceeds a generous time/memory limit of 24 hours and 16GB. Figure 3b shows the results. As expected, MWEM + Exact runs successfully in early iterations, but exceeds resource limits by 35–40 rounds. In comparison, MWEM + Region Graph can run to completion and eventually measure all 2-way marginals. For a fixed number of rounds, Exact has lower error, in this case, substantially so, but results are data-dependent (we evaluate with other datasets in Appendix C). The difference can largely be traced to Exact's better performance on out-of-model cliques. In contrast, HDMM's measurements support all workload queries, so no out-of-model inference is required, and we see little gap between exact and approximate inference. Improving performance of approximate inference for out-of-model marginals is an area to be considered for future work; see Appendix B.

**Additional experiments** We also apply APPGM to improve the accuracy of FEM, a recent state-of-the-art query-answering mechanism [29]. The setup is similar to the DualQuery experiment in [15]:

[3] Scalability experiments were conducted on a 2 core machine with a 2.4GHz CPU and 16 GB of RAM.

we run FEM to completion to release $\mathbf{y}$, but instead of answering queries directly with $\mathbf{y}$, we use APPGM to estimate pseudo-marginals, from which we compute query answers. This leads to a modest error reduction for all $\epsilon$ (Figure 3c; details in Appendix C). In Appendix C we also compare PRIVATE-PGM and APPGM directly to a method proposed in PriView [9] for estimating consistent marginals, and find that PGM-based methods are more accurate for almost all values of $\epsilon$.

## 6 Practical Considerations and Limitations

When using our approach in practice, there are several implementation issues to consider. First note that CONVEX-GBP is an iterative algorithm that potentially requires many iterations to solve Problem 3, and this marginal inference routine is called within each iteration of PROX-PGM. Our theory requires running CONVEX-GBP until convergence, but in practice we only need to run it for a fixed number of iterations. In fact, we find that by *warm starting* the messages in CONVEX-GBP to the values from the previous call, we can actually run only one inner iteration of CONVEX-GBP within each outer iteration of PROX-PGM. This approach works remarkably well and makes much faster progress on reducing the objective than using more inner iterations. The main drawback of this is that we can no longer rely on a line search to find an appropriate step size within PROX-PGM, since one iteration of CONVEX-GBP with warm starting is not necessarily a descent direction. Using a constant step size works well most of the time, but selecting that step size can be tricky. We utilize a simple heuristic that seems to work well in most settings, but may require manual tuning in some cases. Second, we observed that introducing damping into CONVEX-GBP improved its stability and robustness, especially for very dense region graphs. Finally our MWEM experiment revealed that our estimates of out-of-model marginals can sometimes be poor, as the estimate only utilizes local information. We discuss alternative approaches for this problem in Appendix B.

## 7 Related Work

A number of recent approaches support reconstruction for measurements on high-dimensional data. As part of the PriView algorithm [9], the authors describe a method for resolving inconsistencies in noisy measured marginals, which has since been incorporated into other mechanisms [37, 11, 38, 10]. Like APPGM, their method only guarantees local consistency. In Appendix C, we show empirically that it achieves similar (but slightly worse) error than APPGM. In addition, the method is less general, as measured queries may only be marginals, while APPGM allows an arbitrary convex loss function to be specified over the marginals. This extra generality is critical for integrating with mechanisms like HDMM, FEM, and MWEM when the workload contains more complex linear queries.

In a very recent preprint, Aydore et al. [14] propose the *relaxed projection* (RP) reconstruction algorithm, which avoids representing the data vector by optimizing over a tabular data representation. This bypasses the curse of dimensionality but introduces non-convexity, so it is not guaranteed to find a global optimum. While APPGM can be seen as using an *outer approximation* of the marginal polytope, RP uses an *inner approximation*: the estimated marginals are always realizable, but not all realizable marginals can be represented. Favorable error bounds are given for "in-model" queries if optimization is solved exactly. For out-of-model queries, the estimation objective shared by these methods is under-determined: there are generally infinitely many data distributions that minimize it. RP does not specify how to choose among these, in contrast to PRIVATE-PGM, which relies on the principle of maximum entropy. Their work was published while this manuscript was in preparation; we were unable to get an implementation to evaluate it empirically.

To avoid the data vector representation, Liu et al. [13] restrict the support of the data vector to the domain elements that appear in a public dataset. This is much more scalable, but could result in poor performance if the public domain and the input domain differ substantially.

Lastly, Dwork et al. [12] propose an algorithm similar to APPGM. Their approach also projects onto an outer approximation of the marginal polytope, using the Frank Wolfe algorithm. The outer approximation is constructed via geometric techniques and is different from the local polytope we consider. They prove favorable error bounds with polynomial running time, but leave open the implementation and evaluation of their approach. By using the local polytope and message-passing algorithms, our method can scale in practice to problems with millions of variables.

# References


[1] Chao Li, Michael Hay, Vibhor Rastogi, Gerome Miklau, and Andrew McGregor. Optimizing linear counting queries under differential privacy. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 123–134. ACM, 2010.

[2] Jaewoo Lee, Yue Wang, and Daniel Kifer. Maximum likelihood postprocessing for differential privacy under consistency constraints. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 635–644. ACM, 2015.

[3] Dan Zhang, Ryan McKenna, Ios Kotsogiannis, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. Ektelo: A framework for defining differentially-private computations. In *Conference on Management of Data (SIGMOD)*, 2018.

[4] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: the sparse and approximate cases. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 351–360, 2013.

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

[6] Boaz Barak, Kamalika Chaudhuri, Cynthia Dwork, Satyen Kale, Frank McSherry, and Kunal Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 273–282, 2007.

[7] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially private histograms through consistency. *Proceedings of the VLDB Endowment*, 3 (1-2):1021–1032, 2010.

[8] Bolin Ding, Marianne Winslett, Jiawei Han, and Zhenhui Li. Differentially private data cubes: optimizing noise sources and consistency. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 217–228. ACM, 2011.

[9] Wahbeh Qardaji, Weining Yang, and Ninghui Li. PriView: Practical differentially private release of marginal contingency tables. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1435–1446. ACM, 2014.

[10] Zhikun Zhang, Tianhao Wang, Ninghui Li, Jean Honorio, Michael Backes, Shibo He, Jiming Chen, and Yang Zhang. PrivSyn: Differentially private data synthesis. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

[11] Zhikun Zhang, Tianhao Wang, Ninghui Li, Shibo He, and Jiming Chen. CALM: Consistent adaptive local marginal for marginal release under local differential privacy. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 212–229, 2018.

[12] Cynthia Dwork, Aleksandar Nikolov, and Kunal Talwar. Efficient algorithms for privately releasing marginals via convex relaxations. *Discrete & Computational Geometry*, 53(3):650–673, 2015.

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

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

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

[16] www.nist.gov. 2018 differential privacy synthetic data challenge, 2018. URL https://www.nist.gov/communications-technology-laboratory/pscr/funding-opportunities/open-innovation-prize-challenges-1.

[17] Tom Heskes. On the uniqueness of loopy belief propagation fixed points. *Neural Computation*, 16(11):2379–2413, 2004.

[18] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudo-moment matching. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 3, page 3, 2003.

[19] Wim Wiegerinck. Approximations with reweighted generalized belief propagation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2005.

[20] Po-Ling Loh and Andre Wibisono. Concavity of reweighted kikuchi approximation. In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, pages 3473–3481, 2014.

[21] Tamir Hazan, Jian Peng, and Amnon Shashua. Tightening fractional covering upper bounds on the partition function for high-order region graphs. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 356–366, 2012.

[22] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on information theory*, 51(7):2282–2312, 2005.

[23] Tamir Hazan and Amnon Shashua. Convergent message-passing algorithms for inference over general graphs with convex free energies. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, pages 264–273, 2008.

[24] Talya Meltzer, Amir Globerson, and Yair Weiss. Convergent message passing algorithms: a unifying view. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 393–401, 2009.

[25] Tom Heskes. Convexity arguments for efficient minimization of the bethe and kikuchi free energies. *Journal of Artificial Intelligence Research*, 26:153–190, 2006.

[26] Tom Heskes and Onno Zoeter. Generalized belief propagation for approximate inference in hybrid bayesian networks. In *International Workshop on Artificial Intelligence and Statistics*, pages 132–140. PMLR, 2003.

[27] Payam Pakzad and Venkat Anantharam. Estimation and marginalization using the kikuchi approximation methods. *Neural Computation*, 17(8):1836–1873, 2005.

[28] Ryan McKenna, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. Optimizing error of high-dimensional statistical queries under differential privacy. *Proceedings of the VLDB Endowment*, 11(10):1206–1219, 2018.

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

[30] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Third Theory of Cryptography Conference*, 2006.

[31] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

[32] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[33] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[34] Justin Domke. Learning graphical model parameters with approximate marginal inference. *IEEE transactions on pattern analysis and machine intelligence*, 35(10):2454–2467, 2013.

[35] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[36] Diane Ridgeway, Mary Theofanos, Terese Manley, Christine Task, et al. Challenge design and lessons learned from the 2018 differential privacy challenges. 2021.

[37] Rui Chen, Qian Xiao, Yu Zhang, and Jianliang Xu. Differentially private high-dimensional data publication via sampling-based inference. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 129–138. ACM, 2015.

[38] Amrita Roy Chowdhury, Theodoros Rekatsinas, and Somesh Jha. Data-dependent differentially private parameter learning for directed graphical models. In *International Conference on Machine Learning*, pages 1939–1951. PMLR, 2020.

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

[40] URL `https://github.com/usnistgov/PrivacyEngCollabSpace/tree/master/tools/de-identification/Differential-Privacy-Synthetic-Data-Challenge-Algorithms/DPSyn/lib_view`.

## Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Proofs omitted from the main text are in the supplemental material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code appears in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5 and Appendix C.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Repeated 5 trials in every plot, most of the time the variation was too small to be perceptible, although you can see slight variation if you zoom in closely
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] For scalability plots, provided relevant details about computational time and hardware used.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] Licenses of open-source code assets we used are available in repositories of the cited work. We will release our code with a suitable open-source license upon publication.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]