Relaxed Marginal Consistency for Differentially Private Query Answering

Ryan McKenna, Siddhant Pradhan, Daniel Sheldon, Gerome Miklau
College of Information and Computer Sciences
University of Massachusetts
Amherst, MA 01002
{rmckenna, sspradhan, sheldon, miklau}@cs.umass.edu

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 of differentially private algorithms is answering 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, 12–15]. A recently-proposed method, PRIVATE-PGM [16], 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 [16]. PRIVATE-PGM was used in the winning entry of the 2018 NIST differential privacy synthetic data contest [17, 18], as well as in *both* the first and second-place entry of the follow-up 2020 NIST differential privacy temporal map contest [19, 20].

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

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 underlying graphical model, which in turn is determined by which marginals the mechanism depends on. When the mechanism depends on a modest number of carefully chosen marginals, PRIVATE-PGM is extremely efficient. But, as the number of required marginals increases, the underlying graphical model becomes intractably large, and PRIVATE-PGM eventually fails 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 [21–31] 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 imposing strict constraints on the selected measurements. We show that APPGM permits efficient reconstruction for HDMM [32], while also improving its accuracy, allows MWEM [5] to scale to far more measurements, and improves the accuracy of FEM [33].

2 Background

We first review background on our data model, marginals, and differential privacy, following [16].

Data Our input data represents a population of individuals, each contributing a single record $\mathbf{x} = (x_1, \dots, x_d)$ where x_i is the i^{th} attribute belonging to a discrete finite domain Ω_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 $\mathbf{X} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)})$. 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.

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 Ω_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 on r can be computed from the full data vector or the marginal for any superset of attributes by summing over variables that are not in r. 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 any $r \subseteq s$. 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 [34]). *A randomized algorithm $\mathcal{A}$ satisfies (ϵ, δ)-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 ϵ-differential privacy.

3 Private-PGM

In this section we describe PRIVATE-PGM [16], 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 (ϵ, δ)-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 $\mathcal{C}$ [35].*

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 Ω 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

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

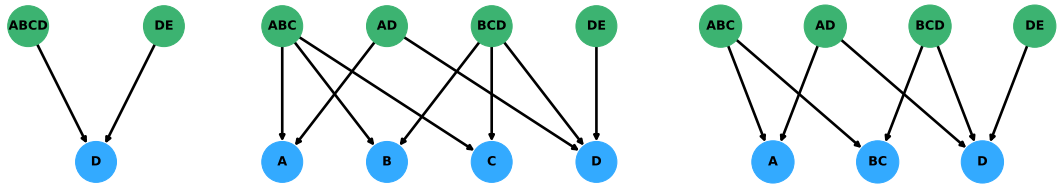
Algorithm 1 PROX-PGM [16]

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 [35]:

- 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 [36]. 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 [16]. 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-occurrence 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 [36]. 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 [35]. In two state-of-the-art mechanisms for synthetic data, MST [18] and PrivMRF [20], $\mathcal{C}$ was specifically chosen to limit the treewidth and ensure tractability of Private-PGM. In other mechanisms agnostic to the limitations of* PRIVATE-PGM*, like HDMM and MWEM, the set $\mathcal{C}$ can often lead to graphs with intractable treewidths.*

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 [36])**.** *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 [36]). *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 [35]. 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 [36, 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 $\mathcal{C}$ [36].

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 [37]. 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 [36]). *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 [35], 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 κ [21–31].

Theorem 1 (Algorithm for Approximate Free Energy Minimization [25])**.** *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 [25] 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 κ), 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 [38] 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 [39]. Each iteration of mirror descent requires solving subproblems of the form:

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

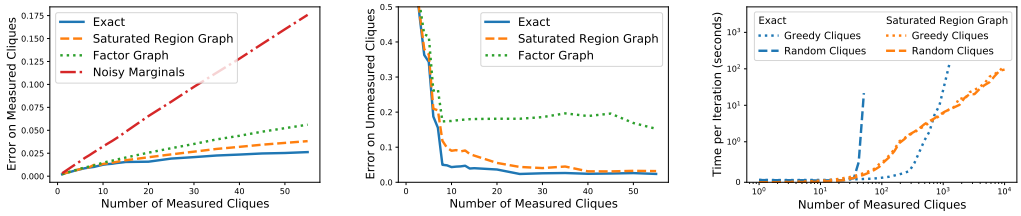
Here, D is a Bregman distance measure and ψ is some strongly convex and continuously differentiable function. Setting $\psi = -H_\kappa$, a negative weighted entropy with any (strongly) convex counting numbers κ, we arrive at the following update equation:

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

□

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 κ_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 constraints. Second, the idealized case uses κ_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 [39].

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.

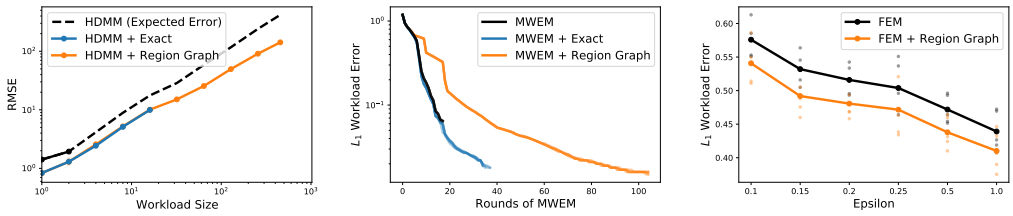
5 Experiments

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 [32] 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 [16], 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 [40], 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 [33]. The setup is similar to the DualQuery experiment in [16]:

[3] Scalability experiments were conducted on two cores of a 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 ϵ (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 ϵ. We additionally compare PRIVATE-PGM and APPGM to the recent Relaxed Projection method [14], and found that APPGM performs better for $\epsilon > 0.1$, although is outperformed for $\epsilon \leq 0.1$.

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 it is better to use PRIVATE-PGM over APPGM in the context of an MWEM-style algorithm, even though PRIVATE-PGM is more limited in the number of rounds it can run for. This performance difference can be traced back to our method for out-of-model inference, where utilization of local information only can lead to poor estimates. 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 [41, 11, 42, 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 concurrent work, Aydore et al. [14] and Liu et al. [15] proposed scalable instantiations of the MWEM algorithm, both avoiding the data vector representation in favor of novel compact representations. Although originally described in the context of MWEM-style algorithms, the key ideas presented in these works can be abstracted to the more general setting considered in this work. Specifically, these methods can be seen as alternatives to APPGM for overcoming the scalability limitations of PRIVATE-PGM; each of these methods make different approximations to overcome the inherent hardness of Problem 1. A direct comparison between PRIVATE-PGM or APPGM, and these alternatives remains an interesting question for future research.

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.

Acknowledgements

This work was supported by the National Science Foundation under grant IIS-1749854, by DARPA and SPAWAR under contract N66001-15-C-4067, and by Oracle Labs, part of Oracle America, through a gift to the University of Massachusetts Amherst in support of academic research.

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 Steven Wu. Leveraging public data for practical private query release. In *Proceedings of the 38th International Conference on Machine Learning*, pages 6968–6977, 2021.

[14] Sergul Aydore, William Brown, Michael Kearns, Krishnaram Kenthapadi, Luca Melis, Aaron Roth, and Ankit A Siva. Differentially private query release through adaptive projection. In *Proceedings of the 38th International Conference on Machine Learning*, pages 457–467, 2021.

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

[16] 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.

[17] 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`.

[18] Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the nist contest: A scalable and general approach to differentially private synthetic data. *Journal of Privacy and Confidentiality*, 2021.

[19] www.nist.gov. 2020 differential privacy temporal map challenge, 2020. URL `https://www.nist.gov/ctl/pscr/open-innovation-prize-challenges/current-and-upcoming-prize-challenges/2020-differential`.

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

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

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

[30] 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.

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

[32] 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.

[33] 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.

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

[35] 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.

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

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

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

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

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

[41] 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.

[42] 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.

[43] 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.

[44] Ninghui Li, Zhikun Zhang, and Tianhao Wang. Dpsyn: Experiences in the nist differential privacy data synthesis challenges. *Journal of Privacy and Confidentiality*, 11(2), 2021.

A Approximate Marginal Inference Algorithm

Algorithm 2 CONVEX-GBP: Convex Generalized Belief Propagation [25]

Input: Region Graph $G = (\mathcal{V}, \mathcal{E})$, parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_r)_{r \in \mathcal{V}}$, convex counting numbers $\kappa_r > 0$
Output: Model marginals $\boldsymbol{\tau} = (\boldsymbol{\tau}_r)_{r \in \mathcal{C}}$
$\kappa_{r,t} = \kappa_r / (\kappa_t + \sum_{p \to t} \kappa_p)$
Initialize $m_{r \to t}(\mathbf{x}_t) = 0$ and $\lambda_{t \to r}(\mathbf{x}_t) = 0$
for $i = 1, \dots$ **do**
  for $r \to t$ **do**
    $m_{r \to t}(\mathbf{x}_t) = \kappa_r \log \Big(\sum_{\mathbf{x}_r \setminus \mathbf{x}_t} \exp \Big((\boldsymbol{\theta}_r(\mathbf{x}_r) + \sum_{c \neq r} \lambda_{c \to r}(\mathbf{x}_c) - \sum_p \lambda_{r \to p}(\mathbf{x}_p)) / \kappa_r \Big) \Big)$
    $\lambda_{t \to r}(\mathbf{x}_t) = \kappa_{r,t} \Big(\boldsymbol{\theta}_t(\mathbf{x}_t) + \sum_c \lambda_{c \to t}(\mathbf{x}_t) + \sum_p m_{p \to t}(\mathbf{x}_t) \Big) - m_{t \to r}(\mathbf{x}_t)$
for $r \in \mathcal{C}$ **do**
  $\boldsymbol{\tau}_r(\mathbf{x}_r) \propto \exp \Big((\boldsymbol{\theta}_r(\mathbf{x}_r) + \sum_t \lambda_{t \to r}(\mathbf{x}_t) - \sum_p \lambda_{r \to p}(\mathbf{x}_r)) / \kappa_r \Big)$
return $\boldsymbol{\tau} = (\boldsymbol{\tau}_r)_{r \in \mathcal{V}}$

Lemma 1. *Let G be a region graph, let κ be positive counting numbers, and suppose $\hat{\boldsymbol{\tau}} =$ CONVEX-GBP$(G, \hat{\boldsymbol{\theta}}, \kappa)$ for parameters $\hat{\boldsymbol{\theta}}$. Then, for any vector $\mathbf{z}$:*

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

For the remainder of this section, let S be the linear subspace parallel to the affine hull of $\mathcal{L}(G)$, and let $S_\perp$ be the orthogonal complement of S. That is, if we write $\mathcal{L}(G) = \{\boldsymbol{\tau} \geq \mathbf{0} : A\boldsymbol{\tau} = \mathbf{b}\}$ using the constraint matrix A, then S is the null space of A. This means that for any $\mathbf{0} < \boldsymbol{\tau} \in \mathcal{L}(G)$ and $\mathbf{z} \in S$, there is some $\lambda > 0$ such that $\boldsymbol{\tau} + \lambda \mathbf{z} \in \mathcal{L}(G)$. On the other hand, if $\mathbf{z} \in \mathcal{L}(G)$ and $\mathbf{z} \notin S$, there is no $\lambda > 0$ such that $\boldsymbol{\tau} + \lambda \mathbf{z} \in \mathcal{L}(G)$.

Proof. Because $\hat{\boldsymbol{\tau}} =$ CONVEX-GBP$(G, \hat{\boldsymbol{\theta}}, \kappa)$ we know that $\hat{\boldsymbol{\tau}}$ minimizes $-\tilde{\boldsymbol{\tau}}^\top \hat{\boldsymbol{\theta}} - H_\kappa(\tilde{\boldsymbol{\tau}})$ over all $\tilde{\boldsymbol{\tau}} \in \mathcal{L}(G)$, and it is easy to see from the final line of CONVEX-GBP that $\hat{\boldsymbol{\tau}} > \mathbf{0}$. Therefore, we can apply Lemma 2 below to conclude that there are vectors $\mathbf{v}_\perp, \mathbf{v}'_\perp \in S_\perp$ such that

$$-\nabla H_\kappa(\hat{\boldsymbol{\tau}}) = \mathbf{u}(\hat{\boldsymbol{\tau}}) + \mathbf{v}_\perp$$
$$\hat{\boldsymbol{\theta}} = \mathbf{u}(\hat{\boldsymbol{\tau}}) + \mathbf{v}'_\perp$$

where $\mathbf{u}(\hat{\boldsymbol{\tau}})$ is the projection of $-\nabla H_\kappa(\hat{\boldsymbol{\tau}})$ onto S.

We will now show that the linear parts of the objectives of the two minimization problems in the lemma statement differ by only a constant. Since the nonlinear part is the same, this will prove that the objectives as a whole differ by only a constant, so the problems have the same minimizers, as stated.

Let $\mathbf{z} = \mathbf{z}_\parallel + \mathbf{z}_\perp$ where $\mathbf{z}_\parallel \in S$ and $\mathbf{z}_\perp \in S_\perp$. Then, for any $\boldsymbol{\tau} \in \mathcal{L}(G)$, the linear component of the first objective is

$$-\boldsymbol{\tau}^\top(-\nabla H_\kappa(\boldsymbol{\tau}) + \mathbf{z}) = -\boldsymbol{\tau}^\top(\mathbf{u}(\hat{\boldsymbol{\tau}}) + \mathbf{v}_\perp + \mathbf{z}_\parallel + \mathbf{z}_\perp) = -\boldsymbol{\tau}^\top(\mathbf{u}(\hat{\boldsymbol{\tau}}) + \mathbf{z}_\parallel) + c \tag{3}$$

where $c = -\boldsymbol{\tau}^\top(\mathbf{v}_\perp + \mathbf{z}_\perp)$ is a constant that does not depend on $\boldsymbol{\tau}$, since, for any $\boldsymbol{\tau}, \boldsymbol{\tau}' \in \mathcal{L}(G)$ we have

$$\boldsymbol{\tau}'^\top(\mathbf{v}_\perp + \mathbf{z}_\perp) - \boldsymbol{\tau}^\top(\mathbf{v}_\perp + \mathbf{z}_\perp) = (\boldsymbol{\tau}' - \boldsymbol{\tau})^\top(\mathbf{v}_\perp + \mathbf{z}_\perp) = 0,$$

since $\boldsymbol{\tau}' - \boldsymbol{\tau} \in S$ and $\mathbf{v}_\perp + \mathbf{z}_\perp \in S_\perp$.

Similarly, the linear component of the second objective is

$$-\boldsymbol{\tau}^\top(\hat{\boldsymbol{\theta}} + \mathbf{z}) = -\boldsymbol{\tau}^\top(\mathbf{u}(\hat{\boldsymbol{\tau}}) + \mathbf{v}'_\perp + \mathbf{z}_\parallel + \mathbf{z}_\perp) = -\boldsymbol{\tau}^\top(\mathbf{u}(\hat{\boldsymbol{\tau}}) + \mathbf{z}_\parallel) + c' \tag{4}$$

where $c' = -\boldsymbol{\tau}^T(\mathbf{v}'_\perp + \mathbf{z}_\perp)$ is a (different) constant independent of $\boldsymbol{\tau}$.

This shows that the objectives differ by a constant, and completes the proof.

Equation (3) and Equation (4) show that the objectives differ by a constant, which completes the proof. □

Lemma 2. *Let G be a region graph, let κ be positive counting numbers, and let $\boldsymbol{\tau} \in \mathcal{L}(G)$ with $\boldsymbol{\tau} > \mathbf{0}$. Define $\Theta(\boldsymbol{\tau}) = \{\boldsymbol{\theta} : \boldsymbol{\tau} = \operatorname{argmin}_{\tilde{\boldsymbol{\tau}} \in \mathcal{L}(G)} -\tilde{\boldsymbol{\tau}}^\top \boldsymbol{\theta} - H_\kappa(\tilde{\boldsymbol{\tau}})\}$ to be the set of all $\boldsymbol{\theta}$ such that $\textsc{Convex-GBP}(G, \boldsymbol{\theta}, \kappa) = \boldsymbol{\tau}$. Then*

$$\Theta(\boldsymbol{\tau}) = \mathbf{u}(\boldsymbol{\tau}) + S_\perp$$

where $\mathbf{u}(\boldsymbol{\tau})$ is the projection of $-\nabla H_\kappa(\boldsymbol{\tau})$ onto S.

Proof. This follows fairly standard arguments in convex analysis after noting that the objective of the optimization problem coincides with the convex conjugate of $-H_\kappa$ (e.g., see Rockafellar, 2015[1]; Bertsekas, 2009[2]), but with some specialization to our setting.

Define $f(\boldsymbol{\tau})$ to be the extended real-valued function that takes value $-H_\kappa(\boldsymbol{\tau})$ for $\boldsymbol{\tau} \in \mathcal{L}(G)$ and $+\infty$ for $\boldsymbol{\tau} \notin \mathcal{L}(G)$. Let $\partial f(\boldsymbol{\tau})$ be the subdifferential of f at $\boldsymbol{\tau} \in \mathcal{L}(G)$.

We will first show that $\Theta(\boldsymbol{\tau}) = \partial f(\boldsymbol{\tau})$.

By the definition of a subgradient, for $\boldsymbol{\tau} \in \mathcal{L}(G)$,

$$\begin{aligned}
\boldsymbol{\theta} \in \partial f(\boldsymbol{\tau}) &\iff f(\tilde{\boldsymbol{\tau}}) \geq f(\boldsymbol{\tau}) + \boldsymbol{\theta}^\top(\tilde{\boldsymbol{\tau}} - \boldsymbol{\tau}) \quad \forall \tilde{\boldsymbol{\tau}} \in \mathbb{R}^d \\
&\iff f(\tilde{\boldsymbol{\tau}}) \geq f(\boldsymbol{\tau}) + \boldsymbol{\theta}^\top(\tilde{\boldsymbol{\tau}} - \boldsymbol{\tau}) \quad \forall \tilde{\boldsymbol{\tau}} \in \mathcal{L}(G) \\
&\iff \boldsymbol{\tau}^\top \boldsymbol{\theta} - f(\boldsymbol{\tau}) \geq \tilde{\boldsymbol{\tau}}^\top \boldsymbol{\theta} - f(\tilde{\boldsymbol{\tau}}) \quad \forall \tilde{\boldsymbol{\tau}} \in \mathcal{L}(G) \\
&\iff \boldsymbol{\tau} = \operatorname*{argmax}_{\tilde{\boldsymbol{\tau}} \in \mathcal{L}(G)} \tilde{\boldsymbol{\tau}}^\top \boldsymbol{\theta} - f(\tilde{\boldsymbol{\tau}}) \\
&\iff \boldsymbol{\tau} = \operatorname*{argmin}_{\tilde{\boldsymbol{\tau}} \in \mathcal{L}(G)} -\tilde{\boldsymbol{\tau}}^\top \boldsymbol{\theta} - H_\kappa(\tilde{\boldsymbol{\tau}}) \\
&\iff \boldsymbol{\theta} \in \Theta(\boldsymbol{\tau}).
\end{aligned}$$

In the second line, we used the fact that the inequality always holds for $\tilde{\boldsymbol{\tau}} \notin \mathcal{L}(G)$ because $f(\tilde{\boldsymbol{\tau}}) = +\infty$ and the other quantities are finite. In the third line, we used the fact that $f(\tilde{\boldsymbol{\tau}})$, which coincides with $-H_\kappa(\tilde{\boldsymbol{\tau}})$ on $\mathcal{L}(G)$, is strictly convex, so $\boldsymbol{\tau}$ is a *unique* maximizer of $\tilde{\boldsymbol{\tau}}^\top \boldsymbol{\theta} - f(\tilde{\boldsymbol{\tau}})$.

Now, we will show that $\partial f(\boldsymbol{\tau}) = \mathbf{u}(\boldsymbol{\tau}) + S_\perp = \{\mathbf{u}(\boldsymbol{\tau}) + \mathbf{v} : \mathbf{v} \in S_\perp\}$, which will conclude the proof.

We use the following characterization of the subdifferential (Rockafellar, 2015, Theorem 23.2):

$$\boldsymbol{\theta} \in \partial f(\boldsymbol{\tau}) \iff \boldsymbol{\theta}^\top \mathbf{z} \leq f'(\boldsymbol{\tau}; \mathbf{z}) \quad \forall \mathbf{z} \in \mathbb{R}^d \tag{5}$$

where $f'(\boldsymbol{\tau}; \mathbf{z})$ is the directional derivative of f along direction $\mathbf{z}$. Since f is the restriction of the differentiable function $-H_\kappa$ to $\mathcal{L}(G)$, its directional derivatives coincide with those of $-H_\kappa$ for points $\boldsymbol{\tau} \in \mathcal{L}(G)$ with $\boldsymbol{\tau} > \mathbf{0}$ and directions in $\mathbf{z} \in S$ (so that $\boldsymbol{\tau} + \lambda \mathbf{z} \in \mathcal{L}(G)$ for small enough $\lambda > 0$), and are equal to $+\infty$ for points $\boldsymbol{\tau} \in \mathcal{L}(G)$ and directions $\mathbf{z} \notin S$ (so that $\boldsymbol{\tau} + \lambda \mathbf{z} \notin \mathcal{L}(G)$ for any $\lambda > 0$). That is, for $\boldsymbol{\tau} \in \mathcal{L}(G)$ with $\boldsymbol{\tau} > 0$,

$$f'(\boldsymbol{\tau}; \mathbf{z}) = \begin{cases} -\nabla H(\boldsymbol{\tau})^\top \mathbf{z} & \mathbf{z} \in S \\ +\infty & \mathbf{z} \notin S \end{cases} \tag{6}$$

Therefore, by Equation (5) and Equation (6), for any $\boldsymbol{\tau} \in \mathcal{L}(G)$ with $\boldsymbol{\tau} > \mathbf{0}$,

$$\begin{aligned}
\boldsymbol{\theta} \in \partial f(\boldsymbol{\tau}) &\iff \boldsymbol{\theta}^\top \mathbf{z} \leq f'(\boldsymbol{\tau}; \mathbf{z}) && \forall \mathbf{z} \in \mathbb{R}^d \\
&\iff \boldsymbol{\theta}^\top \mathbf{z} \leq f'(\boldsymbol{\tau}; \mathbf{z}) && \forall \mathbf{z} \in S \\
&\iff \boldsymbol{\theta}^\top \mathbf{z} \leq -\nabla H(\boldsymbol{\tau})^\top \mathbf{z} && \forall \mathbf{z} \in S \\
&\iff \boldsymbol{\theta}^\top \mathbf{z} = -\nabla H(\boldsymbol{\tau})^\top \mathbf{z} && \forall \mathbf{z} \in S \\
&\iff \boldsymbol{\theta} = \mathbf{u}(\boldsymbol{\tau}) + \mathbf{v} && \mathbf{v} \in S_\perp,
\end{aligned}$$

where $\mathbf{u}(\boldsymbol{\tau})$ is the projection of $-\nabla H(\boldsymbol{\tau})$ onto S. In the second line, we used the fact that the inequality always holds for $\mathbf{z} \notin S$ because $f'(\boldsymbol{\tau}; \mathbf{z}) = +\infty$ and the other quantities are finite. In the fourth line, we observed that $\mathbf{z} \in S$ iff $-\mathbf{z} \in S$ (since S is a linear subspace) and

$$\boldsymbol{\theta}^\top(-\mathbf{z}) \leq -\nabla H(\boldsymbol{\tau})^\top(-\mathbf{z}) \iff \boldsymbol{\theta}^\top \mathbf{z} \geq -\nabla H(\boldsymbol{\tau})^\top \mathbf{z},$$

[1] Rockafellar, R. T. (2015). *Convex analysis*. Princeton university press.

[2] Bertsekas, Dimitri P. Convex optimization theory. Belmont: Athena Scientific, 2009.

so the third line is equivalent to both inequalities holding for all $\mathbf{z} \in S$. The equivalence of the final line to the penultimate line is a straightforward exercise by breaking both $\boldsymbol{\theta}$ and $-\nabla H(\boldsymbol{\tau})$ into their orthogonal components along S and $S_\perp$, respectively, and observing that the component of $-\nabla H(\boldsymbol{\tau})$ along S is $\mathbf{u}(\boldsymbol{\tau})$. □

B Out-of-Model Inference

We now turn our attention to the problem of out-of-model inference; i.e., estimating $\boldsymbol{\tau}_r$ where $r \notin \mathcal{V}$. There are many approaches for this problem that seem natural on the surface, but upon close inspection each one has it's problems. In Section 4, we proposed one approach that had certain desirable properties, but we considered many alternatives which enumerate below and discuss in detail.

B.1 Variable Elimination in $\mathbf{p}_{\boldsymbol{\theta}}$

In PRIVATE-PGM, out-of-model inference was done by performing variable elimination in the graphical model $\mathbf{p}_{\boldsymbol{\theta}}$. There are two problems with applying that idea here. First, variable elimination will not in general be tractable for the graphical models we may encounter, since it is an exact inference method. Second, if we run variable elimination to estimate in-model marginals from $\boldsymbol{\theta}$ produced by PROX-PGM, it will give a different answer than the pseudo-marginals $\boldsymbol{\tau}$ produced by PROX-PGM (even if $\boldsymbol{\tau} \in \mathcal{M}(\mathcal{V})$ is a realizable marginal). In this case, the pseudo-marginals estimated by PROX-PGM are the ones that should be trusted, and the parameters $\boldsymbol{\theta}$ are only useful in the context of our approximate marginal oracle CONVEX-GBP. In summary, this approach is not viable, and even if it was, it has undesirable properties.

Before moving on, we make note of an alternate way to perform exact out-of-model inference that will motivate our first approach to approximate out-of-model inference. They key idea is to add a new zero log-potentials $\boldsymbol{\theta}_r = \mathbf{0}$ for the new clique whose marginal we are interested in estimating. Clearly, the introduction of this zero log-potential does not change the distribution $\mathbf{p}_{\boldsymbol{\theta}}$ or it's in-model marginals $\boldsymbol{\mu}_{\boldsymbol{\theta}}$. However, when we run an exact MARGINAL-ORACLE with these new parameters, it will produce all in-model marginals, and the new out-of-model marginal as well.

B.2 Running CONVEX-GBP on an Expanded Region Graph

Using the idea above, one approach to out-of-model inference is to expand the region graph to include the region r whose pseudo-marginal we are interested in. This will require adding at least one new vertex r to the region graph. Edges and additional vertices could be added depending on the structure of the existing region graph, and the desired local consistency constraints that $\boldsymbol{\tau}_r$ should obey. With this new region graph, we can set $\boldsymbol{\theta}_r = \mathbf{0}$ (and do the same for any additional vertices we added as well), and run CONVEX-GBP on the new graph. This is an interesting idea, but it leaves open several questions:

1. What nodes and edges should be included in the augmented region graph?
2. What counting numbers should be assigned to those nodes?
3. What formal guarantees can we make about this approach?
4. Can we analyze the message-passing equations in CONVEX-GBP to perform an equivalent computation without re-running the algorithm in its entirety?

For question (1) above, a natural choice is to use the same structure as the original region graph. For example, if the original region graph is a factor graph, then we can simply add one new vertex corresponding to the new one, and add edges connecting to the singleton cliques. If the original region graph is saturated, then we can build a new saturated region graph that includes the new clique.

For question (2) above, a natural choice is to use $\kappa'_r = 1$ for all regions r (including the new region), since that is the scheme used to set κ within PROX-PGM. Unfortunately, the new pseudo-marginals $\boldsymbol{\tau}' = \text{CONVEX-GBP}(\boldsymbol{\theta}', \kappa')$ may not agree with the originally optimized pseudo-marginals $\boldsymbol{\tau} = \text{CONVEX-GBP}(\boldsymbol{\theta}, \kappa)$ on the in-model cliques. Specifically, $\boldsymbol{\tau}_r$ need not equal $\boldsymbol{\tau}'_r$ when $r \in \mathcal{V}$. This is clearly undesirable, and would be a consistency violation. A better choice of the counting

numbers would be $\kappa'_r = \kappa_r$ for $r \in \mathcal{V}$ and $\kappa_{r'} = 0$ otherwise. As we show below, this approach has a compelling theoretical guarantee.

Theorem 3. *Let $G = (\mathcal{V}, \mathcal{E})$ be a region graph, $\boldsymbol{\theta}$ be parameters, κ be positive counting numbers and let $\boldsymbol{\tau} = \text{CONVEX-GBP}(G, \boldsymbol{\theta}, \kappa)$. Now let $G' = (\mathcal{V}', \mathcal{E}')$ be a region graph that extends G (i.e, $\mathcal{V} \subseteq \mathcal{V}'$ and $\mathcal{E} \subseteq \mathcal{E}'$), $\boldsymbol{\theta}'_r = \boldsymbol{\theta}_r$ if $r \in \mathcal{V}$ and $\boldsymbol{\theta}'_r = \mathbf{0}$ if $r \notin \mathcal{V}$, $\kappa'_r = \kappa_r$ for $r \in \mathcal{V}$ and $\kappa'_r = \varepsilon$ otherwise.*

$$\boldsymbol{\tau}' = \lim_{\varepsilon \to 0^+} \text{CONVEX-GBP}(G', \boldsymbol{\theta}', \kappa')$$

If $S = \{\boldsymbol{\tau}' \in L(G') \mid \boldsymbol{\tau}'_r = \boldsymbol{\tau}_r \forall r \in \mathcal{V}\} \neq \varnothing$, then

$$\boldsymbol{\tau}' = \operatorname*{argmax}_{\boldsymbol{\tau}' \in S} \sum_{r \in \mathcal{V}' \setminus \mathcal{V}} H(\boldsymbol{\tau}'_r)$$

Proof. We begin by restating the free energy minimization problem solved by CONVEX-GBP.

$$\begin{aligned}
\boldsymbol{\mu}' &= \operatorname*{argmin}_{\boldsymbol{\tau}' \in \mathcal{L}(G')} -\boldsymbol{\theta}^\top \boldsymbol{\tau}' - H_{\kappa'}(\boldsymbol{\tau}') \\
&= \operatorname*{argmin}_{\boldsymbol{\tau}' \in \mathcal{L}(G')} -\Big[\sum_{r \in \mathcal{V}} \boldsymbol{\theta}_r^\top \boldsymbol{\tau}' + \kappa_r H(\boldsymbol{\tau}'_r)\Big] - \Big[\sum_{r \in \mathcal{V}' \setminus \mathcal{V}} \mathbf{0}^\top \boldsymbol{\tau}' + \varepsilon H(\boldsymbol{\tau}'_r)\Big] \\
&= \operatorname*{argmin}_{\boldsymbol{\tau}' \in \mathcal{L}(G')} -\Big[\sum_{r \in \mathcal{V}} \boldsymbol{\theta}_r^\top \boldsymbol{\tau}' + \kappa_r H(\boldsymbol{\tau}'_r)\Big] - \varepsilon \sum_{r \in \mathcal{V}' \setminus \mathcal{V}} H(\boldsymbol{\tau}'_r)
\end{aligned}$$

Note that as $\varepsilon \to 0$, the objective only depends on $\boldsymbol{\tau}_r$ for $r \in \mathcal{V}$ (and not $r \in \mathcal{V}' \setminus \mathcal{V}$). Thus, $\boldsymbol{\tau}_r$ only affect the problem via the constraints they impose on the problem. Since $\boldsymbol{\tau}$ is the optimizer of the relaxed problem when $L(G')$ is replaced by $L(G)$ (which includes a subset of the constraints), if $\boldsymbol{\tau}$ is feasible in the larger problem (which it is by assumption $S \neq \varnothing$), it is also optimal in this problem. Moreover, since we are taking the limit as $\varepsilon \to 0$ from the right, there will be an infinitesimally small entropy penalty, which will force $\boldsymbol{\mu}'_r$ to have maximum entropy among marginals that are consistent with $\boldsymbol{\mu}$, as desired. □

Theorem 3 is a compelling reason to use this approach, namely running CONVEX-GBP with zero counting numbers for the new cliques whose marginals we are estimating. One subtle detail to this theorem is that it is certainly possible that $S = \varnothing$, which means that there aren't any pseudo-marginals in the expanded region graph that are consistent with the pseudo-marginals in the original region graph. In this case, it is not immediately clear how to characterize the behavior of this approach.

Remark 2 (Special Case: Factor Graph)**.** *In the special case when both the original and expanded region graphs are factor graphs, we can guarantee that $S \neq \varnothing$ and we can efficiently estimate the new pseudo-marginal without rerunning* CONVEX-GBP *over the full graph. Since factor graphs only require each pseudo-marginal to be internally consistent with respect to the one-way marginals, we can always find higher-order marginals by multiplying the one-way marginals. Clearly, this gives the maximum entropy estimate for the new pseudo-marginal that is internally consistent with the existing ones. This is a computationally cheap estimate: it simply requires multiplying one-way marginals and does not require any iterative message passing scheme.*

For more complex region graphs, things do not work out so nicely. Since we are mainly interested in saturated region graphs in this work, this nice result for factor graphs is not particularly useful for our purposes. In practice, there is a problem with running CONVEX-GBP with a zero or near-zero counting numbers. We observed empirically that using small counting numbers severely deteriorates the convergence rate of CONVEX-GBP, and for that reason, this is not an ideal approach.

B.3 Minimizing Constraint Violation and Maximizing Entropy

While the method described above has some drawbacks in practice, the principles underlying the approach are still sound: namely, we should find the maximum entropy distribution for the new

marginal that is consistent with the existing marginals (for some natural notion of consistency). However, for complex region graphs, it is certainly possible that no such marginals exist. In that case, a natural alternative would be to find a pseudo-marginal that minimizes the constraint violation, and among all minimizers, has maximum entropy. This is the approach that we evaluated empirically, and described in Section 4.

It requires solving a quadratic minimization problem over the probability simplex. This problem can be readily solved with iterative proximal algorithms like entropic mirror descent [39]. Entropic mirror descent guarantees the solution found will have maximum entropy among all minimizers of the objective. Thus, when $S \neq \varnothing$, this method gives the same answer as Theorem 3. However, it is more general, and also does something principled when $S = \varnothing$. Additionally, this method does not require any information about attributes not in r, and even though it is an iterative algorithm, each iteration runs much faster than an iteration of CONVEX-GBP.

B.4 Running PROX-PGM over expanded local polytope.

While the idea above is more principled than the alternatives that preceded it, it is still not ideal because it does not guarantee perfect consistency between the in-model pseudo-marginals and the out-of-model pseudo marginals. When perfect consistency is not achievable, it settles for minimizing the constraint violation.

We can overcome this limitation by running PROX-PGM on an *over-saturated* region graph. That is the region graph will contain vertices for every region necessary to define the loss function, *and* every region whose pseudo-marginal we are interested in estimated. The additional regions do not affect the loss function (the log-potentials will always remain $\mathbf{0}$), but it does impact the constraints. In particular, upon convergence, the estimated pseudo-marginals will all be locally consistent. This comes at a cost, however. Since the region graph contains more vertices and edges, each iteration of PROX-PGM requires more time, and the algorithm as a whole is slower. Whether it makes sense to use this strategy depends on how important perfect consistency is, as well as how many new marginals must be answered. In our empirical evaluation of this approach, we found that it did produce better estimates than the previous idea, but also took considerably longer.

B.5 Incorporating Global Information

As we saw empirically in Section 5, our approach to out-of-model inference did not perform particularly well compared to the exact method used in PRIVATE-PGM. We conducted more experiments to verify this in Appendix C.2. In this subsection, we explore in greater detail why it did not perform well in all cases.

Consider a simple graphical model with cliques $\mathcal{C} = \{\{A, B\}, \{B, C\}\}$, and suppose that A is highly correlated with B and B is highly correlated with C. Then clearly, A and C should also be highly correlated. When performing exact inference in this model, we preserve this correlation between A and C. However, when we only require local consistency for the new clique, we will assume that A and C are independent, and lose the correlation between A and C.

Note that all methods described thus far suffer from this problem, not just the one method we evaluated in this paper. To correctly preserve the correlation between A and C, we would have to first estimate the $\{A, B, C\}$ marginal then derive the $\{A, C\}$ marginal from it. This could be accomplished by adding an $\{A, B, C\}$ region to the region graph and using any of the methods described above. In this toy problem, it is easy enough to do and feasible, but for larger region graphs, it is not immediately obvious how to generalize the idea.

Since exact marginal inference is not feasible for the graphs we are interested in, it is clear that we must make some approximation. It is not clear what the nature of the approximation should be, however. We showed that only using local information in the approximation has problems in some cases, and utilizing some global information may give better results in some cases. We leave this as an interesting open problem.

C Additional Experiments

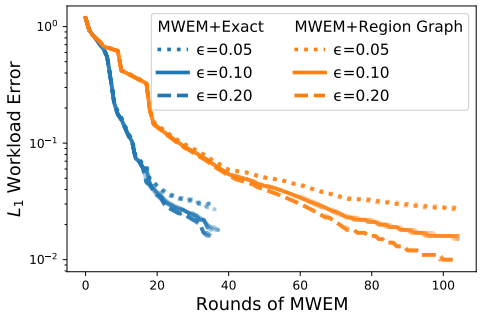
C.1 Synthetic Data used in Experiments

Given a domain size $(n_1, \ldots, n_d)$ and a number of records m, we generate synthetic data to use in experiments as follows:

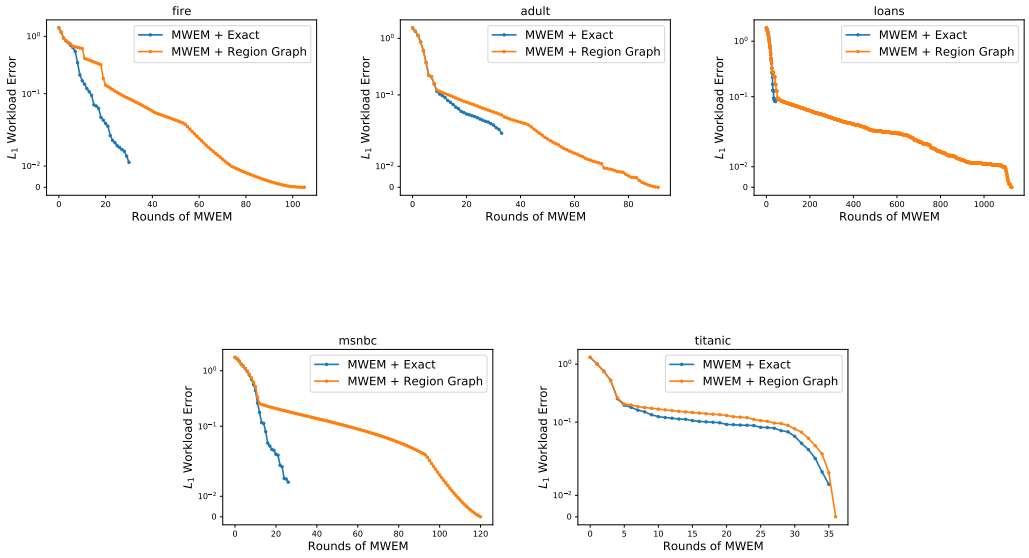
1. Compute a random spanning tree of the complete graph with nodes $1, \ldots, d$. The edges in this tree will correspond to the cliques in our model.
2. For each edge r in the tree, set $\boldsymbol{\theta}_r \sim N(0, \sigma^2)^{n_r}$. Here σ is a "temperature" parameter that determines the strength of the parameters.
3. Sample m records from the graphical model $\mathbf{p}_{\boldsymbol{\theta}}$.

C.2 MWEM Experiments

In Figure 3b of Section 5 we observed that integrating APPGM into MWEM can enable the mechanism to run for more rounds, but the approximation resulted in much worse error for the same number of rounds. When run to completion, MWEM+APPGM did achieve lower error than the minimum error achieved by MWEM+PRIVATE-PGM, but it required running for $3\times$ as many rounds and thus spending $3\times$ as much privacy budget. In the figure below, we include additional lines for different privacy levels $\epsilon = 0.05, 0.1, 0.2$ *per round*. As shown in the figure, it would be better to run MWEM + Exact for 35 round at $\epsilon = 0.1$ than it would be to run MWEM + Region Graph for 70 rounds at $\epsilon = 0.05$.



As hinted at in the main text, the main reason Region Graph performs poorly here is because it only incorporates local information when conducting out-of-model inference, which is problematic for this dataset. To demonstrate that this is really the problem, we repeat the experiment with $\epsilon = \infty$. That is, in each round of MWEM, we exactly select the worse approximated clique, and measure the corresponding marginal with no noise. Since no noise is added, the measured marginals solve Problem 2 and there is no need to run PROX-PGM. Thus, the only difference between Exact and Region Graph is in the handling of out-of-model marginals. We run the experiment for five datasets and plot the results below. The additional error for Region Graph is particularly large for the fire and msnbc dataset but not as much for adult, loans, and titanic. msnbc is a click stream dataset and is thus naturally modeled as a Markov chain. Once the 2-way marginals corresponding to the edges in this Markov chain are measured, MWEM + Exact achieves very low error. MWEM + Exact preserves the long range dependencies between the first and last node in the chain, whereas MWEM + Region Graph only preserves the local dependencies, which explains the difference in this case. Some datasets (like adult, loans, and titanic) do not have strong dependency chains as msnbc does, and in these cases there is a smaller difference in error for out-of-model marginals.



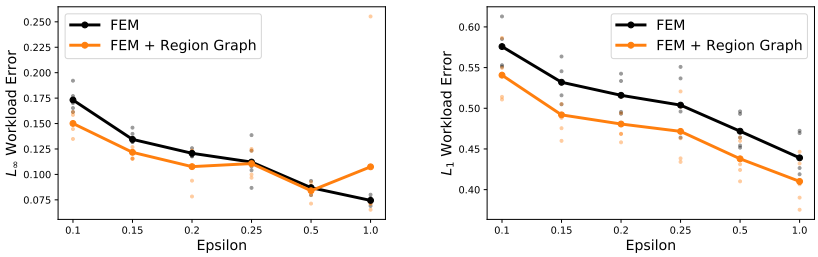
C.3 FEM Experiments

As hinted at in Section 5, our method can be integrated into FEM as well [33]. The integration is similar to how PRIVATE-PGM is used to improve DualQuery [16, 43]. Like MWEM, FEM runs for a specified number of rounds, and maintains an estimate of the data distribution (in tabular format) throughout the execution. In each round, FEM selects a query from the workload that is poorly approximated under the current estimate of the data distribution using the exponential mechanism. This is the only way in which FEM interacts with the sensitive data (Unlike MWEM, it does not measure this query with Laplace noise). It then adds records to the estimated dataset that could explain the previous measurement, in the hopes of reducing the error on that query.

To integrate into FEM, we first note that the mechanism only depends on the data through the answers to the workload. If the workload consists of marginal queries, then our methods apply. We derive an expression for the (negative) log-likelihood of the observations, which are the samples from the exponential mechanism in each round, and use this as our objective function for Problem 2. After solving Problem 2 with PROX-PGM, we can use the estimated pseudo-marginals to answer the workload in place of the synthetic dataset generated by FEM.

In this experiment, we use the adult dataset, as that was one of the main datasets considered in FEM. We note that FEM has a number of hyper-parameters, and it is not obvious how to select them, and selecting them incorrectly can result in very poor performance. However, in the authors open source implementation, they provided a set of tuned hyper-parameters a particular dataset/workload pair: the adult dataset and the workload of 64 random 3-way marginals. For a fair comparison, this is the experimental setting we consider.

We run FEM and FEM + Region Graph and note that FEM + Exact failed to run here, because the underlying junction tree necessary to perform exact marginal inference is too large. We report the L_∞ workload error (which is what FEM is designed to minimize), as well as the L_1 workload error (which better captures the overall error. The results are shown below. In general, FEM + Region Graph achieves slightly lower error than regular FEM in both L_∞ and L_1 error. There is one outlier for L_∞ error when $\epsilon = 1$ that skews the results, and there was negligible improvement at $\epsilon = 0.25$ and $\epsilon = 0.5$ as well. There was consistent improvement in L_1 error for every value of ϵ, although the magnitude of the improvement is somewhat small.



C.4 Comparison with PriView and Relaxed Projection

As discussed in Section 7, PriView proposed a method for resolving inconsistencies in noisy marginals that can be seen as a less general competitor to us. We compare against that competitor here. We use the implementation of this method available from team DPSyn in the 2018 NIST synthetic data competition [44]. In addition, we compare against a variant of the Relaxed Projection algorithm from [14]. We describe the modifications made to this algorithm in the next section.

To compare these methods with our proposed method, we consider the adult dataset and measure 32 random 2-way marginals using the Gaussian mechanism with privacy parameters $\epsilon \in [0.01, 100]$ and $\delta = 10^{-6}$. In this particular case, PRIVATE-PGM can also run, so we include that as a competitor as well. We report the L_1 error of the estimated marginals, averaged over all measured marginals and 5 trials for each method in the table below. All four methods for resolving inconsistencies provide significantly better error than the original noisy marginals.

Ignoring Relaxed Projection, PROX-PGM (Exact) is the best method in every setting except $\epsilon = 100.0$. The second best method is PROX-PGM (Region Graph) in every setting except $\epsilon = 0.01$ and $\epsilon = 100.0$. At the smallest value of ϵ, our method is likely overfitting to the noise, and the estimated pseudo-marginals are likely far from the set of realizable marginals. At the largest value of ϵ, both variants of PROX-PGM simply didn't run for enough iterations (10000 was used in this experiment). Due to the small amount of noise, the true solution to Problem 2 likely does not contain any negatives, and the PriView approach solves the relaxed problem without the non-negativity constraints in closed form. PROX-PGM should eventually converge to the same solution but it would require more than 10000 iterations.

Relaxed Projection (RP) performs slightly better than even PROX-PGM (Exact) for $\epsilon \leq 0.1$, an interesting and surprising observation. We conjecture that this is because RP essentially restricts the search space to distributions which are a mixture of products (as described in the next section). This can be seen as a form of regularization, which can help in the high-privacy / high-noise regime. For $\epsilon > 0.1$, RP is worse than both PROX-PGM (Exact) and PROX-PGM (Region Graph). Moreover, it is the only method whose error does not tend towards 0 as ϵ gets larger. We suspect this is due to the non-convexity in the problem formulation for RP: it is finding a local minimum to the problem that does not have 0 error. Alternatively, it could be possible that the restricted search space does not include a distribution with near-zero error, although we believe this is a less likely explanation.

ϵ	PROX-PGM (Exact)	PROX-PGM (Region Graph)	PriView Consistency	Relaxed Projection	Noisy Marginals
0.0100	0.4375 ± 0.0245	0.5630 ± 0.0344	0.5229 ± 0.0202	0.4189 ± 0.0275	28.050 ± 0.1249
0.0316	0.2848 ± 0.0081	0.3277 ± 0.0100	0.3525 ± 0.0078	0.2567 ± 0.0045	8.8782 ± 0.0254
0.1000	0.1724 ± 0.0032	0.1788 ± 0.0025	0.1965 ± 0.0051	0.1620 ± 0.0036	2.8091 ± 0.0101
0.3162	0.0908 ± 0.0009	0.0931 ± 0.0018	0.1007 ± 0.0016	0.1031 ± 0.0025	0.8919 ± 0.0030
1.0000	0.0433 ± 0.0008	0.0447 ± 0.0006	0.0510 ± 0.0003	0.0746 ± 0.0009	0.2853 ± 0.0007
3.1622	0.0187 ± 0.0001	0.0198 ± 0.0002	0.0229 ± 0.0003	0.0617 ± 0.0007	0.0934 ± 0.0003
10.000	0.0074 ± 0.0001	0.0087 ± 0.0001	0.0095 ± 0.0001	0.0582 ± 0.0011	0.0324 ± 0.0001
31.622	0.0037 ± 0.0000	0.0045 ± 0.0000	0.0040 ± 0.0000	0.0579 ± 0.0017	0.0125 ± 0.0000
100.00	0.0027 ± 0.0000	0.0032 ± 0.0000	0.0018 ± 0.0000	0.0574 ± 0.0012	0.0054 ± 0.0000

C.5 Implementation Details for Relaxed Projection

The authors of the Relaxed Projection method released their code on GitHub. They provided code to run their end-to-end MWEM-style algorithm, but did not expose the subroutine for performing the relaxed projection in a way that can easily be tested in isolation. For that reason, we compare against a faithful reimplementation of their approach. This reimplementation is available in the open source PRIVATE-PGM repository.

One way to view RP is as optimizing over the set of distributions which are mixtures of products. That is, each row of the relaxed tabualr format can be viewed as a product distribution (if the values for each feature are non-negative and sum to one). For multiple rows, this translates to a format that has capacity to represent a mixture of product distributions. While the authors do not propose restricting the feature values to satisfy the aforementioned constraints, in our reimplemenation, we apply softmax transformations to the table to ensure this invariant holds. This is related to $\text{RAP}^{\text{softmax}}$ as described by Liu et al. [15], although the interpretation as a mixture of products was not mentioned in that work. For the experiment above, we consider distributions with 100 mixture components.