# Differentially private algorithms for linear queries via stochastic convex optimization

**Giorgio Micali**
University of Twente

**Clement Lezane**
University of Twente

**Annika Betken**
University of Twente

## Abstract

This article establishes a method to answer a finite set of linear queries on a given dataset while ensuring differential privacy. To achieve this, we formulate the corresponding task as a saddle-point problem, i.e. an optimization problem whose solution corresponds to a distribution minimizing the difference between answers to the linear queries based on the true distribution and answers from a differentially private distribution. Against this background, we establish two new algorithms for corresponding differentially private data release: the first is based on the differentially private Frank-Wolfe method, the second combines randomized smoothing with stochastic convex optimization techniques for a solution to the saddle-point problem. While previous works assess the accuracy of differentially private algorithms with reference to the empirical data distribution, a key contribution of our work is a more natural evaluation of the proposed algorithms' accuracy with reference to the true data-generating distribution.

## 1 INTRODUCTION

As data analysis plays a pivotal role in modern society, the need to protect private information on individuals becomes increasingly important. Ensuring privacy of sensitive information requires robust and effective methods. Among various approaches, differential privacy, introduced by Dwork et al. [2006], emerged as a widely recognized and adopted framework. In fact, today, many public data releases, such as the 2020 US Census (seeAbowd [2018]), are required to adhere to differential privacy standards.

This article studies private synthetic data generation for query release, i.e. it aims at the construction of synthetic datasets as replacement for datasets containing sensitive information, while balancing the accuracy of answers to a predefined set of data queries and the protection of sensitive information in the original dataset.

Unlike commonly known standard privatization methods that are based on the principle of data protection obtained through the incorporation of noise at output level, we propose algorithms that generate a synthetic data distribution that is subject to differential privacy, while close to the data-generating distribution. One fundamental advantage of this method is that once a differentially private distribution is obtained, a synthetic, private dataset can be generated by sampling independently from this distribution.

### Related Works

Generation of private synthetic data for query release is at the heart of research within the field of differential privacy. Many established approaches aiming at (approximate) accuracy in answering a high number of statistical queries through synthetic data release, while, at the same time preserving privacy, are based on the Private Multiplicative Weights (PMW) mechanism introduced by Hardt and Rothblum [2010]. Starting from a uniform distribution on the data universe, PMW iteratively updates a synthetic data distribution based on the errors of all previously answered queries. Recent works improved the computational efficiency of PMW: The Multiplicative Weights Exponential Mechanism (MWEM), introduced in Hardt et al. [2012], builds upon PMW by using the exponential mechanism to focus on the queries with the largest errors, optimizing a surrogate loss function. This allows MWEM to achieve better accuracy and faster convergence, making it more efficient than PMW for practical purposes. While MWEM main-

Proceedings of the 28$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. 

tains a full distribution over the data domain, which leads to high computational complexity especially in high-dimensional settings, DualQuery (established in Gaboardi et al. [2014]) instead maintains a distribution over the query class, which is typically much smaller than the data domain. Moreover, Vietri et al. [2020] proposes three new algorithms for synthetic data release: FEM, sepFEM and DQRS. FEM (Follow-the-Perturbed-Leader with Exponential Mechanism) selects queries using the Exponential Mechanism (like MWEM), but differs by updating through a perturbed optimization problem with stochastic updates. sepFEM is a variant of FEM which achieves better error rates for some classes of queries. The algorithm, however, is less general as it relies on the assumption that for any two distinct pairs of elements in the data domain, there must exist a query for which their answers differ. DQRS is a variant of DualQuery, which implements rejection sampling to reuse previous samples, reducing the number of oracle calls and queries that had not been sampled before. An overview of these algortihms' error bounds can be found in Table 1 (see also Vietri et al. [2020]).

Another two methods for private, synthetic data release have been proposed by Liu et al. [2021]: Private Entropy Projection (PEP) and Generative Networks with the Exponential Mechanism (GEM). PEP builds upon MWEM by adaptively reusing past query measurements and optimizing learning rates resulting in improved accuracy and faster convergence than MWEM. It, however, retains the computational overhead of iterating over the full data distribution. GEM addresses this issue by training neural networks to compactly represent data distributions. While GEM shows strong empirical performance in high-dimensional settings on the one hand, it raises new challenges related to neural network training and hyperparameter tuning on the other. Through the Private-Probabilistic Graphical Model (Private-PGM) McKenna et al. [2019] improve scalability of established algorithms such as MWEM to high-dimensional data universes through entropic mirror descent over a marginal polytope induced by graphical model assumptions on the data distribution. Building upon Private-PGM McKenna et al. [2022] propose an adaptive and iterative mechanism for differentially private synthetic data release (AIM), a wholesome method for generating differentially private synthetic data which iteratively chooses the most useful queries, taking into account how well these approximate the input data and their contribution to answering queries in the current iteration.

Other relevant work in the context of differentially private query release include Bassily et al. [2020] which studies the private and public sample complexities of the Private Query Release Assisted by Public Data (PAP) algorithms and gives upper and lower bounds on both. Another key contribution is Aydore et al. [2021] which introduces the Relaxed Adaptive Projection (RAP) Mechanism, involving differentially private optimization techniques to address high-dimensional query release problems. Lastly, we highlight recent work by Lin et al. [2024] which presents an alternative approach to private synthetic data generation for images through the use of foundation models.

### Our contributions

We revisit the query release problem formulation introduced by Liu et al. [2021], where the goal is to find a distribution within a specified family that minimizes the maximum error across all queries. Unlike previous approaches that rely on the Adaptive Measurements algorithm Hardt and Rothblum [2010]—which minimizes a surrogate loss at each iteration to solve an easier problem—we directly tackle the core objective by optimizing a regularized version of the maximum error among all query answers within the framework of Differentially Private Stochastic Convex Optimization (DP-SCO). Instead of employing adaptive measurements, we solve the problem using differentially private optimization techniques, proposing two algorithms: a private Frank-Wolfe method (DPFW) and a randomized smoothing mirror descent approach (DPAM). By selecting an appropriate regularization constant, we minimize accuracy bounds for both methods.

Furthermore, we assess the accuracy of our methods DPFW and DPAM by providing guarantees with respect to the true data-generating distribution.To the best of our knowledge, this is the first work that offers such guarantees. In fact, and in contrast to our analysis, prior works typically analyze the empirical risk only, i.e. the risk computed from the dataset's empirical distribution.

Notably, the accuracy of DPFW with respect to the population loss aligns with its accuracy on the empirical counterpart. The latter is comparable to most established state-of-the-art accuracy bounds for the aforementioned methods. Furthermore, the upper bound for the population loss of our second algorithm coincides with its empirical counterpart, and for a large query class, DPAM achieves higher accuracy compared to all previously discussed methods.

## 2 PRELIMINARIES

In the following, we establish notations and assumptions that will be used consistently throughout this

manuscript. Let $\mathcal{Z}$ be the data universe, be a finite sample space and $\mathcal{P}$ a probability distribution supported on $\mathcal{Z}$. Without loss of generality we assume that $\mathcal{Z} = \{1, \dots, k\}$, where $|\mathcal{Z}| = k$. We refer to $\mathcal{P}$ as *target distribution*. Due to finiteness of $\mathcal{Z}$, $\mathcal{P}$ can be represented as $(\mathcal{P}(z))_{z\in\mathcal{Z}}$. We define the set of probability distributions on $\mathcal{Z}$ by $\Delta_k := \{(\mathcal{D}(z))_{z\in\mathcal{Z}} \in \mathbb{R}_+^k : \sum_{z\in\mathcal{Z}} \mathcal{D}(z) = 1\}$. Let us denote with $S_n = \{z_1, \dots, z_n\}$ a dataset of $n$ points obtained by sampling independently from $\mathcal{Z}$ according to $\mathcal{P}$. From $S_n$ we retrieve the empirical distribution $\mathcal{P}_n$, i.e. $\mathcal{P}_n(z) = |\{i \in [n] : z_i = z\}|/n$. We refer to any function $q : \mathcal{Z} \to [-1, 1]$ as a *query* of $\mathcal{Z}$, and we define the *statistical query* associated to $q$ with respect to the distribution $\mathcal{P}$ as the quantity $\mathbb{E}_{\mathbf{z}\sim\mathcal{P}}[q(\mathbf{z})]$. If $\mathcal{P}_n$ is the empirical distribution associated with $S_n$, the statistical query takes the form $\frac{1}{n}\sum_i q(z_i)$, which is known as *counting query*, see Dwork and Roth [2014]. Since each query $q$ is a function defined on $\mathcal{Z}$, it can be represented as $q = (q(z))_{z\in\mathcal{Z}} \in [-1, 1]^k$. A query is said to be *answered* when the vector $q = (q(z))_{z\in\mathcal{Z}}$ is reported. Similarly, any distribution $\mathcal{D} \in \Delta_k$ can be expressed as $\mathcal{D} = (\mathcal{D}(z))_{z\in\mathcal{Z}}$. Depending on the context, we refer to queries $q$ and distributions $\mathcal{D}$ as functions or vectors. Against this background, $\mathbb{E}_{\mathbf{z}\sim\mathcal{P}}[q(\mathbf{z})] = \langle q, \mathcal{P}\rangle$, where $\langle\cdot,\cdot\rangle$ is the standard scalar product in $\mathbb{R}^k$.

In many practical applications, queries take two values only. If, for instance, $S_n$ represents a set of individuals, a query $q$ could ask for the presence/absence of a specific feature in these individuals. In this particular case, a corresponding query may only take the values 0 and 1 ($q(z) = 1$ indicating the presence of the feature for the individual $z$, while $q(z) = 0$ indicates its absence). The statistical query $\mathbb{E}_{\mathbf{z}\sim\mathcal{P}}[q(\mathbf{z})]$ then corresponds to the expected proportion of individuals with a corresponding feature.

Within the field of differential privacy, an objective is to answer a prescribed set of queries $\mathcal{Q} \subset \{q : \mathcal{Z} \longrightarrow [-1, 1]\}$ as accurately as possible, while preserving the privacy of individuals. For this, we construct a privacy-preserving distribution $\mathcal{P}^{\text{priv}}$ over $\mathcal{Z}$, whose statistical properties are close to $\mathcal{P}$, i.e., more precisely, that its statistical queries closely approximate those of $\mathcal{P}$. Mathematically speaking, this is achieved by minimization of $\max_{q\in\mathcal{Q}} |\mathbb{E}_{\mathbf{z}\sim\mathcal{P}}[q(\mathbf{z})] - \mathbb{E}_{\mathbf{z}\sim\mathcal{P}^{\text{priv}}}[q(\mathbf{z})]|$ over all $\mathcal{P}^{\text{priv}}$ subject to privacy, in the sense of differential privacy defined below. Identifying $\mathcal{Q}$ with the set $\{(q(z))_{z\in\mathcal{Z}} : q \in \mathcal{Q}\} \subset [-1, 1]^k$ for convenience, we assume $\mathcal{Q}$ to be symmetric, i.e. $\mathcal{Q} = -\mathcal{Q}$, such that we could as well be aiming at minimizing $\max_{q\in\mathcal{Q}} (\mathbb{E}_{\mathbf{z}\sim\mathcal{P}}[q(\mathbf{z})] - \mathbb{E}_{\mathbf{z}\sim\mathcal{P}^{\text{priv}}}[q(\mathbf{z})])$. Since $\text{conv}(\mathcal{Q})$ is a polyhedron and the maximum of a linear function over a polyhedron is achieved at one of its vertices, we can, furthermore, replace $\mathcal{Q}$ by its convex hull. As a result, the considered optimization problem can be reduced to the *saddle-point formulation*

$$\min_{\mathcal{D}\in\Delta_k} \phi(\mathcal{D}), \text{ where } \phi(\mathcal{D}) := \max_{q\in\text{conv}(\mathcal{Q})} \langle q, \mathcal{P} - \mathcal{D}\rangle \,, \tag{$P$}$$

see Liu et al. [2021]. Our goal is to design a differentially private algorithm that approximates the optimal solution to ($P$) starting from a finite dataset $S_n = \{z_1, \dots, z_n\}$ consisting of $n$ data points independently drawn from $\mathcal{Z}$ according to $\mathcal{P}$. From this perspective, a linear relaxation of the saddle-point problem ($P$) has been addressed in a recent work by González et al. [2024] (see Section B of their appendix).

### Differential Privacy

Differential privacy applies to randomized algorithms, i.e. algorithms employing randomness as part of their logic or procedure for the protection of privacy. A randomized algorithm is considered differentially private if adding, removing or replacing a single data point in the dataset it applies to does not significantly change the distribution of its output. Two datasets $S_1$ and $S_2$ that differ in one data point only are called *neighboring*.

Mathematically, differential privacy can be formalized as follows:

**Definition 1** (Dwork and Roth [2014])**.** Let $\varepsilon, \delta \geq 0$. A randomized algorithm $\mathcal{A} : \mathcal{X} \longrightarrow \mathcal{O}$ is $(\varepsilon, \delta)$ - differentially private (DP) if for every pair of neighboring datasets $S_1$, $S_2$ and every subset $T \subset \mathcal{O}$

$$\mathbb{P}(\mathcal{A}(S_1) \in T) \leq e^{\varepsilon}\mathbb{P}(\mathcal{A}(S_2) \in T) + \delta \,. \tag{1}$$

Note that the parameters $\varepsilon$ and $\delta$ quantify privacy. If $\varepsilon = 0$ and $\delta = 0$, it follows that $\mathbb{P}(\mathcal{A}(S_1) \in T) = \mathbb{P}(\mathcal{A}(S_2) \in T)$. Thus, the distribution of the output of $\mathcal{A}$ is independent from the data and protects privacy perfectly. If $\delta = 0$, we say that $\mathcal{A}$ satisfies pure privacy and that it is $\varepsilon-$DP.

## 3 MAIN RESULTS

This article establishes a variation of the Frank-Wolfe algorithm for answering queries on a dataset under differential privacy constraints. Moreover, it proposes an algorithm for synthetic data generation which adheres to the principles of differential privacy through randomized smoothing. For their respective objectives, both algorithms aim at a solution for an entropy-regularized formulation of ($P$).

### 3.1 A regularized optimization problem

Without imposing any restrictions on the set of admissible distributions $\Delta_k$ in the saddle-point problem $(P)$, the solution to this optimization problem is given by $\mathcal{D}^* = \mathcal{P}$. Our goal, however, is to find an approximation to this solution that is $(\varepsilon, \delta)$-DP. Moreover, the optimization problem as stated in $(P)$ does not satisfy strong convexity. For this reason, we modify the optimization problem through incorporation of a regularization transforming it to a strongly convex-concave problem.

More precisely, for $\alpha \geq 0$, we define the *regularized saddle-point problem* $(P_\alpha)$ as

$$\min_{\mathcal{D}\in\Delta_k} \max_{q\in\mathrm{conv}(\mathcal{Q})} \mathcal{L}_\alpha(q,\mathcal{D}) \quad = \quad \min_{\mathcal{D}\in\Delta_k} \phi_\alpha(\mathcal{D}), \qquad (P_\alpha)$$

where $\mathcal{L}_\alpha(q,\mathcal{D}) := \langle q, \mathcal{P}-\mathcal{D}\rangle + \alpha H(\mathcal{D})$, $\phi_\alpha(\mathcal{D}) := \max_{q\in\mathrm{conv}(\mathcal{Q})} \mathcal{L}_\alpha(q,\mathcal{D})$, and $H(\mathcal{D}) = \sum_{z\in\mathcal{Z}} \mathcal{D}(z)\log(\mathcal{D}(z))$ is the negative entropy of $\mathcal{D}\in\Delta_k$.

The use of the entropy regularization term is motivated by two key observations. First, in practice, the set $\mathcal{Z}$ often contains a number of elements that is much larger than the sample size, i.e. $n < k$. As a result, for the empirical distribution $\mathcal{P}_n$ over $\mathcal{Z}$, $\mathcal{P}_n(z) = 0$ for some $z \in \mathcal{Z}$. Entropy regularization addresses this by pushing the optimum towards the uniform distribution, thereby ensuring that most points in $\mathcal{Z}$ have a non-zero probability of being sampled, even if they are not present in $S_n$. Second, the entropy function $H$ is strongly convex ensuring strong convexity of $\phi_\alpha$ and therefore a unique solution of $(P_\alpha)$.

The regularization parameter $\alpha$ controls the suboptimality of the solution with respect to the unregularized problem. A value of $\alpha$ that is close to zero ensures the regularized problem $(P_\alpha)$ to be similar to the original problem $(P)$.

For an explicit solution of the regularized optimization problem $(P_\alpha)$, we consider the corresponding dual problem:

$$\max_{q\in\mathrm{conv}(\mathcal{Q})} \psi_\alpha(q), \qquad (D_\alpha)$$

where $\psi_\alpha(q) := \langle q, \mathcal{P}\rangle - \alpha H^*\left(\frac{q}{\alpha}\right)$ and $H^*(y) = \log\left(\sum_j e^{y_j}\right)$ (the Fenchel conjugate of $H$). As a consequence of Zalinescu's theorem, see Zălinescu [1983], the Fenchel conjugate of $H$ is 1-smooth with respect to the $\|\cdot\|_\infty$, making $(D_\alpha)$ a smooth concave optimization problem over a polyhedron.

Accordingly, $(D_\alpha)$ can be expressed as maximization of an objective function of the form $\mathbb{E}_{\mathbf{z}\sim\mathcal{P}}[\ell(q,\mathbf{z})]$, where $\ell(q,\mathbf{z}) := q(\mathbf{z}) - \alpha H^*\left(\frac{q}{\alpha}\right)$ is both $L_0$-Lipschitz and $L_1$-smooth for suitable constants $L_0$ and $L_1$. Since the sets $\Delta_k$ and $\mathrm{conv}(\mathcal{Q})$ are convex, the design of an algorithm solving the dual falls within the framework of Differentially Private Stochastic Convex Optimization (DP-SCO). Lastly, according to Slater's theorem strong duality holds, i.e. $\phi^* = \psi^*$, where $\phi^*$ and $\psi^*$ denote minimum and maximum for $(P_\alpha)$ and $(D_\alpha)$, respectively.

In the following, we will focus on the regularized dual problem. We will later show how solving this problem leads to a differentially private solution of the unregularized primal, which is our main interest.

### 3.2 Solving the dual: The differentially private Frank-Wolfe algorithm

This section establishes the differentially private Frank-Wolfe algorithm (DPFW) as a solution to the dual optimization problem $(D_\alpha)$. An analysis of the corresponding solution to the unregularized optimization problem $(P)$ is done in three steps, each corresponding to one of the following three subsections. Section 3.2.1 quantifies the difference in the values of the dual $\psi_\alpha$ evaluated in the true maximizer and its approximation through the output of DPFW. Section 3.2.2 quantifies the difference in the values of the primal function $\phi_\alpha$ evaluated in the true minimizer and its approximation resulting from the output of DPFW. Section 3.2.3 considers an optimal choice of the regularization parameter $\alpha$ and quantifies the overall accuracy of the resulting value of the unregularized optimization problem $(P)$. Most notably, the quantification of accuracy in Section 3.2.3 is based on the true distribution $\mathcal{P}$, not on the empirical distribution $\mathcal{P}_n$ associated with $S_n$. For a high-level illustration of the individual steps leading to a differentially private solution of $(P)$ see Figure 1.

#### 3.2.1 Bounding the dual gap

The (non-privatized) Frank-Wolfe algorithm (FW) aims at minimization of a convex (or maximization of a concave) and differentiable objective function over a compact and convex feasible set. Since $\psi_\alpha$ is concave and 1-smooth and $\mathrm{conv}(\mathcal{Q})$ compact and convex, it is therefore particularly suited for solving the dual $(D_\alpha)$. While competing methods such as (stochastic) gradient descent require a projection step back to the feasible set in each iteration, FW only needs the solution of a convex problem over the same set in each iteration, and automatically stays in the feasible set. Under the constraints of the dual this is known to be computationally efficient, see Jaggi [2013] and Frank and Wolfe [1956].

The main idea of FW is to linearize the objective function, resulting in the consideration of the following,

simpler optimization problem over the feasible set:

$$\max_{q\in\mathcal{Q}}\langle\hat{\nabla}\psi_\alpha(q_0), q\rangle, \tag{2}$$

where

$$\hat{\nabla}\psi_\alpha(q) := \mathcal{P}_n - \nabla H^*\left(\frac{q}{\alpha}\right), \tag{3}$$

approximates the unknown gradient $\nabla\psi_\alpha(q) = \mathcal{P} - \nabla H^*\left(\frac{q}{\alpha}\right)$. Moreover, for privatization of the algorithm, solving the linearized optimization problem is randomized by replacing (2) with the optimization problem

$$s_0 = \operatorname*{argmax}_{q\in\mathcal{Q}}\{\langle\hat{\nabla}\psi_\alpha(q_0), q\rangle + u_{s,0}\}, \tag{4}$$

where $u_{s,0} \overset{i.i.d}{\sim} \text{Lap}(\lambda)$ and $\text{Lap}(\lambda)$ denotes a Laplace distribution with parameter $\lambda$; see Claim 3.9 in Dwork and Roth [2014] (Report Noisy Max). Starting from an initialization in some $q_0 \in \text{conv}(\mathcal{Q})$, an approximation of the maximizer is given by moving a step of size $\gamma$ in direction of the maximizer, i.e. we define

$$q_1 := q_0 + \gamma(s_0 - q_0).$$

Iteration of these steps results in a sequence of locations $q_0, q_1, \ldots$ approximating the maximizer of $\psi_\alpha$. FW iterates until a stopping criterion is met (typically when the gradient's magnitude becomes small (indicating convergence), or after a fixed number of iterations). In contrast to this, the differentially private Frank-Wolfe algorithm (DPFW) chooses its output $q_\alpha^{out}$ uniformly from $q_0, \ldots, q_{T-1}$, where $T$ corresponds to a predefined number of iterations.

The most significant difference between FW and DPFW, however, corresponds to the introduction of Laplace noise to the optimization step (4). The intuition behind this so-called *Report Noisy Max* is to calibrate the amount of noise, i.e. the choice of the parameter $\lambda$, to the algorithm's sensitivity (i.e., how much its output changes in response to a single data point) thereby introducing privacy; for a more detailed description of *Report Noisy Max*, see Section B.1 in the supplementary material.

An algorithmic description of DPFW specifying number of iterations, step size, and Laplace noise as needed for the subsequently derived theoretical guarantees is provided by Algorithm 1. It depends on the diameter of the set $\mathcal{Q}$ which, with respect to the $\|\cdot\|_r$-norm, is denoted by $D_r$.

The evaluation of the approximation of the maximizer of the dual ($D_\alpha$) through the output of DPFW is based on the following observation: since $\psi_\alpha$ is concave and differentiable, it holds that

$$\begin{aligned} 0 &\leq \psi_\alpha(q_\alpha^*) - \psi_\alpha(q_\alpha^{out}) \\ &\leq \max_{q\in\text{conv}(\mathcal{Q})}\langle\nabla\psi_\alpha(q), q_\alpha^{out} - q\rangle \ := g_{\psi_\alpha}(q_\alpha^{out}), \end{aligned}$$

**Algorithm 1** Differentially private Frank-Wolfe algorithm (DPFW)

**Input**: $(S_n,\ \mathcal{Q},\ (\varepsilon,\delta),\ \alpha)$
Number of iterations: $T = \frac{D_1^{3/2}\varepsilon n}{\sqrt{32\alpha\log(1/\delta)\log(2|\mathcal{Q}|)}}$
Step size: $\gamma = 2\sqrt{\frac{\alpha}{TD_\infty}}$
Noise: $\lambda = \frac{4D_1\sqrt{2T\log(1/\delta)}}{\varepsilon n}$
Initialization: $q_0 \in \mathcal{Q}$
**for** $t = 0, \ldots, T-1$ **do**
  Compute $\hat{\nabla}\psi_\alpha(q_t)$ according to (3)
  For all $s \in \mathcal{Q}$ sample $u_{s,t} \overset{i.i.d}{\sim} \text{Lap}(\lambda)$
  $s_t = \operatorname*{argmax}_{s\in\mathcal{Q}}\{\langle\hat{\nabla}\psi_\alpha(q_t), s\rangle + u_{s,t}\}$
  $q_{t+1} = q_t + \gamma(s_t - q_t)$
**Output**: $q_\alpha^{\text{out}} = q_U$ for $U \sim \text{Uni}(\{0, \ldots, T-1\})$

where $q_\alpha^*$ denotes the maximizer of $\psi_\alpha$. We refer to $g_{\psi_\alpha}(q_\alpha^{out})$ as *Frank-Wolfe Gap*. Naturally, an upper bound on $g_{\psi_\alpha}(q_\alpha^{out})$ is also an upper bound for the difference $\psi_\alpha(q_\alpha^*) - \psi_\alpha(q_\alpha^{out})$ which quantifies the error arising from utilization of DPFW to the end of maximizing $\psi_\alpha$. The following theorem verifies $(\varepsilon,\delta)$-differential privacy of DPFW and gives an explicit upper bound for the Frank-Wolfe gap.

**Theorem 1.** *The differentially private Frank -Wolfe algorithm DPFW is $(\varepsilon,\delta)$-DP. Moreover, the Frank-Wolfe Gap $g_{\psi_\alpha}(q_\alpha^{out}) := \max_q\langle\nabla\psi_\alpha(q_\alpha^{out}), q_\alpha^{out} - q\rangle$ satisfies*

$$\begin{aligned} &\mathbb{E}[g_{\psi_\alpha}(q_\alpha^{out})] \\ &= \mathcal{O}\left(\frac{\log^{1/4}(1/\delta)\log^{1/2}(|\mathcal{Q}|)}{\alpha^{1/4}(\varepsilon n)^{1/2}} + D_1\sqrt{\frac{\log(k)}{n}}\right). \end{aligned} \tag{5}$$

Theorem 1 highlights the typical trade-off offered by DP algorithms: stronger privacy protection (smaller values of $(\varepsilon,\delta)$) results in weaker accuracy, while higher accuracy necessitates the reduction of noise, which in turn degrades privacy guarantees. Additionally, smaller values of $\alpha$ indicate a larger Frank-Wolfe gap. At the same time, as stated in Section 3, a solution to the regularized problem ($P_\alpha$) is expected to be closer to a solution to the original optimization problem ($P$) for small values of $\alpha$. Moreover, if the number of queries is exponential, i.e $|\mathcal{Q}| = \exp(r)$ for some $r \geq 0$, the logarithmic factor $\log(|\mathcal{Q}|)$, enables answering them all in linear time.

Lastly, note that the second summand in (5) arises from the evaluation of the Frank-Wolfe gap against the population loss. Although not explicitly shown in this work, our method, when evaluated in terms of empirical loss, provides as upper bound, the first summand on the right hand side of (5) in Theorem 1.

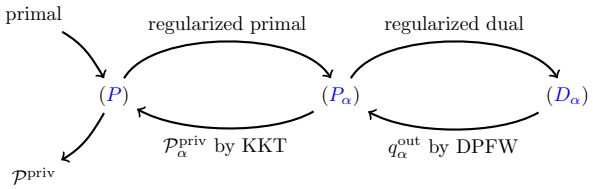


Figure 1: High-level illustration of the procedure to find $\mathcal{P}^{\text{priv}}$.

#### 3.2.2 Bounding the primal gap

In the previous section we derived an approximate $(\varepsilon, \delta)$-DP solution to $(D_\alpha)$ through the differentially private Frank-Wolfe algorithm DPFW. This section quantifies the deviation of the corresponding value of the objective function $\phi$ in the primal $(P)$ from the actual optimum $\phi^*$ of $\phi$. For this, we replace the maximizer in $(P_\alpha)$ by the output $q_\alpha^{\text{out}}$ of DPFW and, subsequently, analytically solve the resulting optimization problem through implementation of the Karush–Kuhn–Tucker conditions. The distribution solving the corresponding optimization problem is denoted as $\mathcal{P}_\alpha^{\text{priv}}$, i.e.

$$\begin{aligned}\mathcal{P}_\alpha^{\text{priv}} &= \operatorname*{argmin}_{\mathcal{D}\in\Delta_{\mathcal{Z}}} \mathcal{L}_\alpha(q_\alpha^{\text{out}}, \mathcal{D}) \\ &= \operatorname*{argmin}_{\mathcal{D}\in\Delta_{\mathcal{Z}}} \left(\langle q_\alpha^{\text{out}}, \mathcal{P}-\mathcal{D}\rangle + \alpha H(\mathcal{D})\right) =: \mathcal{A}_\alpha(S_n).\end{aligned} \tag{6}$$

It can be shown that the analytic solution to (6) is $\mathcal{P}_\alpha^{\text{priv}} = \nabla H^*\left(\frac{q_\alpha^{\text{out}}}{\alpha}\right)$ (see Lemma 3 in the supplementary material). To measure how much $\mathcal{P}_\alpha^{\text{priv}}$ deviates from the original $\mathcal{P}$, we estimate an upper bound on the primal gap, defined as follows:

$$\mathbf{Gap}_{(P)}(\mathcal{D}) = \phi(\mathcal{D}) - \phi^\star \,,$$

where $\phi^\star := \min_{\mathcal{D}\in\Delta_k} \phi(\mathcal{D})$.

The primal gap quantifies the deviation of a feasible point from its optimum. In the previous section, we established an upper bound on the Frank-Wolfe gap for the dual problem. This upper bound can be translated back to the primal setting, which is the content of Theorem 2. The entire process is summarized in Figure 1.

**Theorem 2.** *For any $\alpha > 0$, we define $q_\alpha^{out}$ as the output of Algorithm 1, Then, for the distribution $\mathcal{P}_\alpha^{priv} = \mathcal{A}_\alpha(S_n)$, obtained via (6), the following holds:*

$$\mathbb{E}[\boldsymbol{Gap}_{(P)}(\mathcal{P}^{priv})] \le \mathbb{E}[g_{\psi_\alpha}(q_\alpha^{out})] + \alpha \log k \,. \tag{7}$$

#### 3.2.3 Bounding the overall gap

Theorems 1 and 2 establish upper bounds on dual and primal gap of the proposed privacy-preserving algorithm solving the optimization problem $(P)$. By choosing $\alpha^*$ as the value that minimizes the right-hand side of (7), we obtain

$$\alpha^* = \frac{\log^{1/5}(1/\delta)\log^{2/5}|\mathcal{Q}|}{(\varepsilon n)^{2/5}\log^{4/5} k} \tag{8}$$

and, accordingly, we define $\mathcal{P}^{\text{priv}} = \mathcal{P}_{\alpha^*}^{\text{priv}}$. The following theorem summarizes these findings:

**Theorem 3.** *Let $q_{\alpha^*}^{out}$ be the output of Algorithm 1, and let $\alpha^*$ be the value defined in (8). Then, the distribution $\mathcal{P}^{priv} = \mathcal{A}_{\alpha^*}(S_n)$, obtained through (6), guarantees that the maximum error in answering all queries from $\mathcal{Q}$ using $\mathcal{P}^{priv}$ is bounded as follows:*

$$\begin{aligned}&\mathbb{E}\left[\max_{q\in\mathcal{Q}}\langle q, \mathcal{P}-\mathcal{P}^{priv}\rangle\right] \\ =&\mathcal{O}\left(\frac{\log^{1/5}(1/\delta)\log^{2/5}|\mathcal{Q}|\log^{1/5} k}{(\varepsilon n)^{2/5}} + D_1\sqrt{\frac{\log k}{n}}\right).\end{aligned}$$

We consider the specific setting $\mathcal{Z} = \{0,1\}^d$ (see also Section 2) to derive an upper bound for comparison with other algorithms (see Table 1 in Section 5). For this, we do not consider the population risk $\max_{q\in\mathcal{Q}}\langle q, \mathcal{P}-\mathcal{P}^{\text{priv}}\rangle$, but its empirical analogue $\max_{q\in\mathcal{Q}}\langle q, \mathcal{P}_n-\mathcal{P}^{\text{priv}}\rangle$. In this case, based on Theorem 3, the following bound can be derived:

$$\mathbb{E}[\mathbf{Gap}_{(P)}(\mathcal{P}^{\text{priv}})] = \mathcal{O}\left(\frac{\log^{2/5}(|\mathcal{Q}|)\log^{1/5}(1/\delta)d^{1/5}}{(\varepsilon n)^{2/5}}\right).$$

Note that this term does not include the summand $D_1\sqrt{\log k/n}$ arising from using the unbiased estimator $\hat{\nabla}\psi_\alpha$ for $\nabla\psi_\alpha$. In fact, replacing $\mathcal{P}$ with the empirical distribution $\mathcal{P}_n$ from $S_n$ allows exact computation of $\nabla\psi_\alpha(q) = \mathcal{P}_n - \nabla H^*(q/\alpha)$, eliminating the need for approximation. Lastly, we report the computational complexity for answering all queries by Algorithm 1. Note that for each iteration $t = 0,\dots,T-1$ the DPFW spends $\mathcal{O}(k)$ operations to compute the gradient approximation $\hat{\nabla}\psi_\alpha$, $\mathcal{O}(|\mathcal{Q}|)$ for sampling from a Laplace distribution and $\mathcal{O}(|\mathcal{Q}|)$ for solving the suboptimization problem (as it visits at least every vertex once). Hence, we obtain a total cost of $\mathcal{O}((k+|\mathcal{Q}|)T)$ . If the algorithm is implemented with optimal $\alpha^*$, the total cost is upper-bounded by

$$\mathcal{O}\left((k+|\mathcal{Q}|)\frac{(n\varepsilon)^{1.2}\log^{2/5}(k)}{\log^{0.6}(1/\delta)}\right).$$

### 3.3 Solving the primal: Randomized Smoothing and Stochastic Composite Minimization

This section provides yet another solution to the unregularized optimization problem ($P$). In contrast to Section 3.2, which focused on a solution by solving the regularized dual ($D_\alpha$), this section provides a solution by solving the regularized primal ($P_\alpha$). For this, note that ($P_\alpha$) can be written as

$$\min_{\mathcal{D}\in\Delta_k} \left[\phi(\mathcal{D}) + \alpha H(\mathcal{D})\right] , \qquad (P_\alpha)$$

where $\phi(\mathcal{D}) := \max_{q\in\text{conv}(\mathcal{Q})}\langle q, \mathcal{P}-\mathcal{D}\rangle$, and $H(\mathcal{D}) = \sum_{z\in\mathcal{Z}} \mathcal{D}(z)\log(\mathcal{D}(z))$ is the negative entropy of $\mathcal{D}\in\Delta_k$. If $\phi$ and $H$ in ($P_\alpha$) satisfy specific smoothness and convexity properties, the optimization problem would meet the requirements summarized as stochastic composite oracle model in d'Aspremont et al. [2022] and could be solved by the so-called (non-)accelerated complementary composite stochastic mirror-descent, an optimization technique proposed in d'Aspremont et al. [2022]. Non-smoothness of the inner maximization problem in ($P_\alpha$), i.e. non-smoothness of $\phi$, however, prevents from direct application of corresponding results. To resolve this issue, we pursue randomized smoothing of $\phi$ (Section 9) and a subsequent application of Algorithm 3.2 in d'Aspremont et al. [2022], guaranteeing fast optimization rates (Section 3.3.2).

#### 3.3.1 Randomized Smoothing

We resolve non-smoothness of $\phi$ by a convolution-based smoothing technique amenable to non-smooth stochastic optimization problems; see, for example, Duchi et al. [2012]. The intuition underlying this approach is that convolving two functions results in a new function that is at least as smooth as the smoothest of the two original functions. For our purposes, we consider convolution of $\phi$ with the density $\varphi_\sigma$ of a multivariate normal distribution with mean vector $\mathbf{0}$ and covariance matrix $\sigma^2 I_k$, $\sigma > 0$. Accordingly, we consider the smoothed objective function $\phi_\sigma$ defined as follows:

$$\phi_\sigma(\mathcal{D}) := \int_{\mathbb{R}^k} \phi(\mathcal{D}+x)\varphi_\sigma(x)\,dx = \mathbb{E}_\xi[\phi(\mathcal{D}+\xi)] , \quad (9)$$

where $\xi$ is a random vector in $\mathbb{R}^k$ with probability density $\varphi_\sigma$. In general, $\phi_\sigma$ is convex and differentiable whenever $\phi$ is convex.

Convexity of $\phi$ is established by the following lemma:

**Lemma 1.** *$\phi : \Delta_k \mapsto \mathbb{R}_+$ is convex and 1-Lipschitz with respect to $\|\cdot\|_1$.*

Properties of $\phi_\sigma$ inherited from $\phi$ are summarized by the following proposition.

**Proposition 1.** *Let $\mathcal{Q}\subseteq[-1,1]^k$ be a compact and convex set, and let $\phi(\mathcal{D}) := \max_{q\in\mathcal{Q}}\langle q, \mathcal{P}-\mathcal{D}\rangle$, the function $\phi_\sigma$ defined in (9). Then,*

*(i) $\|\phi_\sigma - \phi\|_\infty \le \sigma w(\mathcal{Q})$, where $w(\mathcal{Q}) := \mathbb{E}_{\xi\sim\varphi}\left[\max_{q\in\mathcal{Q}}\langle q,\xi\rangle\right]$ is the Gaussian width of $\mathcal{Q}$ and $\varphi$ denotes the density of the multivariate standard normal distribution.*

*(ii) $\phi_\sigma$ is convex and 1-Lipschitz with respect to $\|\cdot\|_1$.*

*(iii) $\phi_\sigma$ is $1/\sigma$-smooth with respect to $\|\cdot\|_1$.*

#### 3.3.2 Stochastic Composite Minimization

Randomized smoothing and regularization allow for reformulation of the primal problem ($P_\alpha$) into the following approximation within the complementary composite framework considered in d'Aspremont et al. [2022]:

$$\min_{\mathcal{D}\in\Delta_k} \Phi_{\alpha,\sigma}(\mathcal{D}) := \min_{\mathcal{D}\in\Delta_k} \left[\phi_\sigma(\mathcal{D}) + \alpha H(\mathcal{D})\right], \qquad (P_{\alpha,\sigma})$$

where $\phi_\sigma$ is $\frac{1}{\sigma}$-smooth and $H$ is 1-strongly convex.

Given a Gaussian vector $\xi\sim\mathcal{N}(0,\sigma^2 I_k)$, a stochastic first-order oracle for $\phi_\sigma$ can be constructed as

$$G_\sigma(\mathcal{D},\xi,S_n,\mathcal{Q}) \in \underset{q\in\text{conv}(\mathcal{Q})}{\text{argmax}} \langle q, \mathcal{P}_n - \mathcal{D} + \xi\rangle. \qquad (10)$$

As shown in Section C.2 of the supplementary material, this oracle is an unbiased estimator of $\nabla\phi_\sigma$ and has bounded variance. We notice that we cannot use the usual gradient descent on $\Phi_{\alpha,\sigma}$ for two reasons, the first is that our problem is constrained in the simplex $\Delta_k$ and the second is that we only have gradients related to $\nabla\phi_\sigma$ and not to the whole $\nabla\Phi_{\alpha,\sigma}$.

Solving equation ($P_{\alpha,\sigma}$) requires tools from the composite setting. For that, we claim that for any coefficients $A, B > 0$, and vectors $g\in\mathbb{R}^k, \mathcal{D}'\in\Delta_k$, the following problem has a closed form solution (see Section 6 of the Appendix)

$$\min_{\mathcal{D}\in\Delta_k}\left\{A[\langle g,\mathcal{D}\rangle + H(\mathcal{D})] + BD_H(\mathcal{D},\mathcal{D}')\right\}$$

with $H$ being the negative entropy function and $D_H$ being the Kullback-Leibler divergence. Now we have verified all the assumptions of Theorem 3.4 in d'Aspremont et al. [2022] and we provide an $(\varepsilon,\delta)$-DP variation of the Accelerated Composite Mirror Descent (Algorithm 2).

Algorithm 2 combines mirror descent with accelerated gradient descent. Mirror descent performs gradient updates in the dual space, which is particularly useful in non-Euclidean domains where standard updates

**Algorithm 2** Differentially Private Complementary Accelerated Mirror-Descent

**Input**: $(S_n,\ \mathcal{Q},\ (\varepsilon,\delta),\ \alpha)$
Number of iterations: $T = \frac{\log^{1/2} k}{\log^{1/2}(1/\delta)} \frac{\varepsilon n}{w(\mathcal{Q})}$
Smoothing parameter: $\sigma = \frac{4\sqrt{T\ln(1/\delta)}}{n\varepsilon}$
Step sizes: $\eta_t = t + \sqrt{\frac{4}{\alpha\sigma}} + 1$ , for $1 \le t \le T$
Initialisation $\mathcal{D}_1 = \mathcal{D}_1^{ag} = [1/k, \dots, 1/k]^\top$
**for** $1 \le t \le T$ **do**
  $\mathcal{D}_t^{md} = \frac{\sum_{\tau=1}^{t-1}\eta_\tau}{\sum_{\tau=1}^{t}\eta_\tau}\mathcal{D}_t^{ag} + \frac{\eta_t}{\sum_{\tau=1}^{t}\eta_\tau}\mathcal{D}_t$
  $g_t = G_\sigma(\mathcal{D}_t^{md}, \xi_t, S_n, \mathcal{Q}), \xi_t \sim \mathcal{N}(0, \sigma^2 I_k)$
  $\mathcal{D}_{t+1} = \operatorname{argmin}_{\mathcal{D}\in\Delta_k} \left\{ \eta_t[\langle g_t, \mathcal{D}\rangle + H(\mathcal{D})] + \left(\sum_{\tau=1}^{t-1}\eta_\tau\right) D_H(\mathcal{D}, \mathcal{D}_t) \right\}$
  $\mathcal{D}_{t+1}^{ag} = \frac{\sum_{\tau=1}^{t-1}\eta_\tau}{\sum_{\tau=1}^{t}\eta_\tau}\mathcal{D}_t^{ag} + \frac{\eta_t}{\sum_{\tau=1}^{t}\eta_\tau}\mathcal{D}_{t+1}$
**Output**: $\mathcal{P}_\alpha^{\text{priv}} = \mathcal{D}_{T+1}^{ag}$.

may lead to infeasible or geometrically meaningless solutions.

Our algorithm works for any $\alpha \ge 0$, but we offer a simplified interpretation for the case $\alpha = 0$. Starting from an initial $\mathcal{D}_1$, at the $t$-th iteration the current distribution $\mathcal{D}_t$ is mapped to the dual space via the mirror map $\nabla H$. There, the update is $\nabla H(\mathcal{D}_t) - \frac{\eta_t}{\sum_{\tau=1}^{t-1}\eta_\tau} g_t$, where $g_t = G_\sigma(\mathcal{D}_t, \xi_t, S_n, \mathcal{Q})$ approximates $\nabla\Phi_{\alpha,\sigma}$. The updated value is then returned back to the primal space using the inverse mirror map $\nabla H^{-1} = \nabla H^*$. Thus, the update in the primal space is:

$$\mathcal{D}_{t+1} = \nabla H^* \left( \nabla H(\mathcal{D}_t) - \frac{\eta_t}{\sum_{\tau=1}^{t-1}\eta_\tau} g_t \right) .$$

As demonstrated in Theorem 6.13 of Orabona [2020], this update is equivalent to

$$\mathcal{D}_{t+1} = \operatorname*{argmin}_{\mathcal{D}\in\Delta_k} \left\{ \eta_t \langle g_t, \mathcal{D}\rangle + \left( \sum_{\tau=1}^{t-1}\eta_\tau \right) D_H(\mathcal{D}, \mathcal{D}_t) \right\} .$$

In general, accelerated methods improve upon the convergence rate of standard first-order methods for convex optimization problems by exploiting the convexity and the smoothness of the objective function. For this, it combines the traditional mirror descent updates with an additional sequence of gradient evaluations at $(D_t^{md})$ to add a momentum to the mirror descent. While regular mirror descent takes steps based solely on the current gradient, accelerated mirror descent adds information from previous iterations (aggregation), allowing for faster convergence.

Differential privacy is introduced through the noisy oracle $G_\sigma$, which injects calibrated noise at each iteration, ensuring that the Algorithm is $(\varepsilon,\delta)-$DP. The next result provides the accuracy and the privacy guarantee for algorithm 2.

**Theorem 4.** *Algorithm 2 is $(\varepsilon,\delta)$-DP and the output $\mathcal{P}_\alpha^{priv}$ satisfies*

$$\mathbb{E}\left[ \boldsymbol{Gap}_{(P_{\alpha,\sigma})}(\mathcal{P}_\alpha^{priv}) \right] = \mathcal{O}\left( \frac{\log^{3/4}(1/\delta)}{\alpha \log^{5/4}(k)} \frac{w^{5/2}(\mathcal{Q})}{(n\varepsilon)^{3/2}} + \frac{\log^{1/2}(1/\delta)}{\alpha \log^{1/2}(k)} \frac{w(\mathcal{Q})}{n\varepsilon} \right).$$

Similarly to the approach taken with DPFW, optimizing over $\alpha$ yields the following result:

**Theorem 5.** *Let us consider the accuracy guarantee obtained in Theorem 4. For $n \ge \frac{w^3(\mathcal{Q})\log(1/\delta)}{\varepsilon \log^{3/2}(k)}$ and $\alpha^* = \frac{\log^{1/2}(1/\delta) w^{1/2}(\mathcal{Q})}{\log^{3/4}(k)\sqrt{n\varepsilon}}$, the distribution $\mathcal{P}^{priv} := \mathcal{P}_{\alpha^*}^{priv}$ satisfies*

$$\mathbb{E}\left[ \max_{q\in\mathcal{Q}} \langle q, \mathcal{P} - \mathcal{P}^{priv} \rangle \right] = \mathcal{O}\left( \frac{w^{1/2}(\mathcal{Q}) \log^{1/4}(k) \log^{1/4}(1/\delta)}{\varepsilon^{1/2} n^{1/2}} \right) .$$

If the query class is contained in the Euclidean unit ball $\mathcal{B}_2^k := \left\{ x \in \mathbb{R}^k : \|x\|_2 \le 1 \right\}$, the previous upper bound provides the strongest accuracy guarantee among the methods listed in Table 1. More specifically, if $\mathcal{Q}$ is a finite subset of $\mathcal{B}_2^k$, then $w(\mathcal{Q}) \le C \log|\mathcal{Q}|$ , where $C$ is a constant independent of the dimension $k$. In the following, computation complexity of Algorithm 2: for each iteration DPAM spends $\mathcal{O}(k)$ operations for sampling from a Gaussian distribution, $\mathcal{O}(|\mathcal{Q}|)$ operations to compute the gradient approximation $G$ (equation (10)), and $\mathcal{O}(k)$ operations to solve the sub-optimization problem. Thus, under the assumption that the Gaussian width $w(\mathcal{Q}) \ge 1$, the total cost of DPAM is

$$\mathcal{O}\left( (k + |\mathcal{Q}|) \frac{\log^{1/2}(k) n\varepsilon}{\log^{1/2}(1/\delta)} \right) .$$

## 4 EFFICIENT IMPLEMENTATION

The curse of dimensionality poses a significant challenge in query data release, as highlighted in the introduction of this work. Alternative approaches, such as DualQuery and PGM, mitigate this through different strategies. We propose two approaches for an efficient implementation of DPFW and DPAM. The first adopts a problem formulation similar to McKenna et al. [2019], exploiting the structure of the data distribution to reduce computational complexity. The second applies when the number of queries is substantially

smaller than the data domain size, without requiring distributional assumptions.

Given the data domain $\mathcal{Z} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$, we assume query sets over a finite subset of attributes $A \subset [d]$. Let $z_A$ be the sub-vector restricted to $A$. For all $z_A \in \Pi_{i\in A}\mathcal{X}_i$, we consider the marginal $\mu_A(z_A) = \frac{1}{k}\sum_{i=1}^{k} \mathbf{1}\left((z_i)_A = z_A\right)$, and $n_A := \Pi_{i\in A}|\mathcal{X}_i| < \infty$. We reformulate ($P$) as

$$\min_{D\in\Delta_{n_A}} \max_{q\in Q_A} \langle q, D - \mu_A\rangle \ , \qquad (P_A)$$

where $Q_A$ represents a marginal query set. Our algorithms compute an optimal $\mu_A^{priv}$ with computational costs discussed in Section 3. This serves as a substitute for $\hat{\mu}$ in McKenna et al. [2019], allowing the analysis to proceed using the PGM framework. This means that $\mu_A^{priv}$ can be integrated into their algorithm to compute the corresponding $\mathcal{P}_n$. Specifically, our algorithms replace equation (1) in their paper.

For a domain $\mathcal{Z} = \{z_1, \ldots, z_k\}$ and a query class $\mathcal{Q}$, we represent queries as a $k \times |\mathcal{Q}|$ matrix:

$$Q = \begin{bmatrix} q_1(z_1) & \cdots & q_{|\mathcal{Q}|}(z_1) \\ \vdots & \ddots & \vdots \\ q_1(z_k) & \cdots & q_{|\mathcal{Q}|}(z_k) \end{bmatrix}$$

with rank $p \leq |\mathcal{Q}|$. In this case a QR decomposition yields: $Q = AR = A\begin{bmatrix} R_1 \\ 0 \end{bmatrix}$, where $R_1$ is an $|\mathcal{Q}| \times |\mathcal{Q}|$ upper triangular matrix and $A$ is unitary matrix. ($P$) then writes as

$$\min_{D\in\Delta_k} \max_{q\in Q} \langle q, P - D\rangle = \min_{D\in\Delta_k} \max_{\tilde{q}\in \mathrm{Im}(R)} \langle \tilde{q}, A(P - D)\rangle.$$

The iterative updates $q_t$ in DPFW remain within the column span of $R$, meaning they take the form $q = (*, \ldots, *, 0, \ldots, 0)^\top$ with a maximum of $k - p$ nonzero entries. This structure significantly reduces computational complexity per iteration.

If $\mathcal{P}_n$ is computed and stored in the memory, it only needs to be computed once. Therefore, the cost for reduces to

$$\mathcal{O}\left(k|\mathcal{Q}|^2 + (p + |\mathcal{Q}|)\frac{\log^{1/2}(k) n\varepsilon}{\log^{1/2}(1/\delta)}\right) . \qquad (11)$$

Similar considerations hold for the DPAM. For DPAM the corresponding stochastic gradient (10) lies in a lower-dimensional subspace of dimension $p$, yielding the same cost as in (11).

## 5 CONCLUSION

This work presents the first significant population risk bounds for differentially private synthetic data. For this, we introduce two algorithms: DP-Frank-Wolfe (DPFW) and DP-Randomized Smoothing with mirror descent (DPAM). Both algorithms achieve tight worst-case error bounds that can compete with state-of-the-art methods, as shown in Table 1. Notably, DPAM maintains the worst-case error for empirical and population risk. For a specific class of queries, DPAM outperforms all previous methods in terms of sample size $n$ (with a rate of $1/\sqrt{n}$), offering stronger guarantees by incorporating smaller factors related to other parameters compared to the established algorithms.

| **Established** | **Empirical Loss** |
|---|---|
| MWEM | $\mathcal{O}\left(\frac{d^{1/4}\log^{1/2}\lvert\mathcal{Q}\rvert\log^{1/2}(1/\delta)}{n^{1/2}\varepsilon^{1/2}}\right)$ |
| DualQuery | $\mathcal{O}\left(\frac{d^{1/6}\log^{1/2}\lvert\mathcal{Q}\rvert\log^{1/6}(1/\delta)}{n^{1/3}\varepsilon^{1/3}}\right)$ |
| FEM | $\mathcal{O}\left(\frac{d^{3/4}\log^{1/2}\lvert\mathcal{Q}\rvert\log^{1/2}(1/\delta)}{n^{1/2}\varepsilon^{1/2}}\right)$ |
| sepFEM | $\mathcal{O}\left(\frac{d^{5/8}\log^{1/2}\lvert\mathcal{Q}\rvert\log^{1/2}(1/\delta)}{n^{1/2}\varepsilon^{1/2}}\right)$ |
| DQRS | $\mathcal{O}\left(\frac{d^{1/5}\log^{3/5}\lvert\mathcal{Q}\rvert\log^{1/5}(1/\delta)}{n^{2/5}\varepsilon^{2/5}}\right)$ |
| DPFW(E) | $\mathcal{O}\left(\frac{d^{1/5}\log^{2/5}\lvert\mathcal{Q}\rvert\log^{1/5}(1/\delta)}{n^{2/5}\varepsilon^{2/5}}\right)$ |
| **New** | **Population Loss** |
| DPFW(P) | $\mathcal{O}\left(\frac{d^{1/5}\log^{2/5}\lvert\mathcal{Q}\rvert\log^{1/5}(1/\delta)}{n^{2/5}\varepsilon^{2/5}} + D_1\frac{\log^{1/2}(d)}{n^{1/2}}\right)$ |
| DPAM | $\mathcal{O}\left(\frac{d^{1/4}w^{1/2}(\mathcal{Q})\log^{1/4}(1/\delta)}{n^{1/2}\varepsilon^{1/2}}\right)$ |

Table 1: This table provides upper bounds on the empirical loss $\max_{q\in\mathcal{Q}}\langle q, \mathcal{P}_n - \mathcal{P}^{\mathrm{priv}}\rangle$ and the population loss $\max_{q\in\mathcal{Q}}\langle q, \mathcal{P} - \mathcal{P}^{\mathrm{priv}}\rangle$ of established algorithms on the two newly proposed algorithms. For this, the data universe is assumed to be $\mathcal{Z} = \{0,1\}^d$.

We conclude by pointing towards future research on establishing optimality of upper bounds. In the supplementary material we discuss the results of Bun et al. [2018] on private lower bounds, and we conjecture that with proper control of the Gaussian width, DPAM is optimal up to a factor $\log(1/\delta)$.

## Acknowledgments

The authors gratefully acknowledge Cristóbal Guzmán's support and, in particular, his invaluable insights in the early stages of this work. Annika Betken gratefully acknowledges financial support from the Dutch Research Council (NWO) through VENI grant 212.164.

## References

Jacob Abernethy, Chansoo Lee, and Ambuj Tewari. Perturbation techniques in online learning and opti-

mization. *Perturbations, Optimization, and Statistics*, 233:233–264, 2016.

John M. Abowd. The U.S. Census Bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 2867. Association for Computing Machinery, 2018.

Alekh Agarwal, Peter L. Bartlett, Pradeep Ravikumar, and Martin J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Trans. Inform. Theory*, 58(5):3235–3249, 2012.

Hilal Asi, Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: optimal rates in $\ell_1$ geometry. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 393–403. PMLR, 2021.

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 (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 457–467. PMLR, 2021.

Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Thakurta. Private stochastic convex optimization with optimal rates. In *Neural Information Processing Systems*, 2019.

Raef Bassily, Albert Cheu, Shay Moran, Aleksandar Nikolov, Jonathan Ullman, and Steven Zhiwei Wu. Private query release assisted by public data. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 695–703. PMLR, 2020.

Raef Bassily, Cristóbal Guzmán, and Michael Menart. Differentially private stochastic optimization: new results in convex and non-convex settings. In *Advances in Neural Information Processing Systems*, volume 34, pages 9317–9329, 2021.

A. Beck. *First-order methods in optimization*. Society for Industrial and Applied Mathematics, 2017. ISBN 9781611974980.

Dimitri P. Bertsekas. Stochastic optimization problems with nondifferentiable cost functionals with an application in stochastic programming. *Journal of Optimization Theory and Applications*, 12(2):218–231, 1973.

Digvijay Boob and Cristóbal Guzmán. Optimal algorithms for differentially private stochastic monotone variational inequalities and saddle-point problems. *Mathematical Programming*, pages 1–43, 2021.

Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2014. ISBN 978-0-521-83378-3.

Mark Bun, Jonathan R. Ullman, and Salil P. Vadhan. Fingerprinting codes and the price of approximate differential privacy. *SIAM Journal on Computing*, 47(5):1888–1938, 2018.

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

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

Alexandre d'Aspremont, Cristóbal Guzmán, and Clément Lezane. Optimal algorithms for stochastic complementary composite minimization. *SIAM Journal on Optimization*, November 2022.

John C. Duchi, Peter L. Bartlett, and Martin J. Wainwright. Randomized Smoothing for Stochastic Optimization. *SIAM Journal on Optimization*, 22(2): 674–701, 2012.

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

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Theory of Cryptography Conference (TCC)*, volume 3876, pages 265–284. Springer, 2006.

Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60, 2010.

Lutz Dümbgen, Sara A. van de Geer, Mark C. Veraar, and Jon A Wellner. Nemirovski's inequalities revisited. *The American Mathematical Monthly*, 117(2): 138–160, 2010.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1–2):95–110, 1956.

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 *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 1170–1178, Beijing, China, 2014. PMLR.

Tomás González, Cristóbal Guzmán, and Courtney Paquette. Mirror descent algorithms with nearly

dimension-independent rates for differentially-private stochastic saddle-point problems, 2024.

Moritz Hardt and Guy N. Rothblum. A Multiplicative Weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 61–70. IEEE, 2010.

Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, STOC '10, page 705–714, New York, NY, USA, 2010. Association for Computing Machinery.

Moritz Hardt, Katrina Ligett, and Frank Mcsherry. A simple and practical algorithm for differentially private data release. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in neural information processing systems*, volume 25. Curran Associates, Inc., 2012.

Martin Jaggi. Revisiting Frank-Wolfe: projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 427–435, 2013.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005. ISSN 0022-0000. Learning Theory 2003.

Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, Harsha Nori, and Sergey Yekhanin. Differentially private synthetic data via foundation model APIs 1: Images, 2024.

Terrance Liu, Giuseppe Vietri, and Zhiwei Steven Wu. Iterative methods for private synthetic data: unifying framework and new methods. In *Neural Information Processing Systems*, pages 9765–9774, 2021.

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.

Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. Aim: An adaptive and iterative mechanism for differentially private synthetic data. *Proceedings of the VLDB Endowment*, 15(11):2599 – 2612, 2022.

Ilya Mironov. Rényi differential privacy. *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275, 2017.

Geoffrey Negiar, Gideon Dresdner, Alicia Y. Tsai, Laurent El Ghaoui, Francesco Locatello, and Fabian Pedregosa. Stochastic Frank-Wolfe for constrained finite-sum minimization. In *International Conference on Machine Learning*, 2020.

Francesco Orabona. *A modern introduction to online learning*. OpenBU, 2020.

K. S. Sesh Kumar and Marc Peter Deisenroth. Differentially private empirical risk minimization with sparsity-inducing norms. *ArXiv*, abs/1905.04873, 2019.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107 – 194, 2012.

Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: convergence results and optimal averaging schemes. In *Proceedings of the 30th International Conference on Machine Learning*, pages 71–79. PMLR, 2013.

Thomas Steinke and Jonathan R. Ullman. Between pure and approximate differential privacy. *Journal of Privacy and Confidentiality*, 7(2), 2016.

Kunal Talwar, Abhradeep Guha Thakurta, and Li Zhang. Nearly optimal private LASSO. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Jonathan R. Ullman and Salil P. Vadhan. PCPs and the Hardness of Generating Private Synthetic Data. In *Theory of Cryptography Conference (TCC)*, volume 6597, pages 400–416. Springer, 2011.

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

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: analyzing the connection to overfitting. *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282, 2017.

C. Zălinescu. On uniformly convex functions. *Journal of Mathematical Analysis and Applications*, 95:344–374, 1983.

## Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

# Supplementary material

This document provides auxiliary results needed for the proofs of the mathematical results in the main document. It, moreover, establishes the proofs of Theorems 1 and 2 stated in Section 3.2.1 of the main document (Section B), and the proof of Theorems 4 and 5 stated in Section 3.3 of the main document (Section C). The last section (Section D) provides a short discussion of lower bounds for the accuracy of algorithms for private synthetic data release.

**Notation:** For $\|\cdot\|$ a norm and $f : E \longrightarrow \mathbb{R}$ convex (over $E \subset \mathbb{R}^d$ convex), we denote with $f^*(y) = \sup_{x\in E}(\langle x, y\rangle - f(y))$ the Fenchel conjugate of $f$. The dual norm of $\|\cdot\|$ is denoted by $\|x\|_* = \sup_{\|y\|\leq 1}\langle x, y\rangle$ . For $f$, we denote with $\nabla f$ and $\nabla^2 f$ the gradient vector and Hessian matrix of $f$, respectively. For a vector $v = (v_1, \ldots, v_d)^\intercal$, the symbol $\mathrm{diag}(v)$ denotes a diagonal matrix whose elements on the main diagonal are given by the elements of $v$ and all the remaining entries are set to zero. For $\mathbf{z} \in \mathcal{Z} = \{z_1, \ldots, z_k\}$, we define the vector $e_{\mathbf{z}} := \left(\mathbf{1}_{\{\mathbf{z}=z_1\}}, \ldots, \mathbf{1}_{\{\mathbf{z}=z_k\}}\right)^\intercal$ . Any notation not explicitly introduced in this paragraph adheres to the conventions established in the main document.

## A Auxiliary results

Both algorithms established in this article are based on the consideration of an optimization problem of the form

$$\min_{\mathcal{D}\in\Delta_k} \Psi_{A,B,C}(\mathcal{D}), \quad \Psi_{A,B,C}(\mathcal{D}) := A\langle g, \mathcal{D}\rangle + BH(\mathcal{D}) + CD_H, (\mathcal{D}, \mathcal{D}'),$$

$A \in \mathbb{R}$, $B > 0$, $C \geq 0$, $g \in \mathbb{R}^k$, $\mathcal{D}' \in \Delta_k$, and where $H$ denotes the negative entropy function on $\Delta_k$ and $D_H$ the associated Bregman divergence. A sufficient condition guaranteeing a unique solution to an optimization problem of this type, corresponds to strong convexity of the objective function (here: $\Psi_{A,B,C}$). For this, note that the first summand in the representation of $\Psi_{A,B,C}$ is linear, while the second summand corresponds to the negative entropy function which is strongly convex according to Example 2.5 in Shalev-Shwartz [2012]. For establishing strong convexity of $\Psi_{A,B,C}$, it therefore suffices to show that $D_H(\cdot, D')$ is (strongly) convex. In fact, it can be shown that any strongly convex function $f$ induces strong convexity of the Bregman divergence in its first argument:

**Lemma 2.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable, $\mu$-strongly convex function, i.e. for all $x, y \in \mathbb{R}^n$,*

$$f(x) \geq f(y) + \langle \nabla f(y), x - y\rangle + \frac{\mu}{2}\|x - y\|^2.$$

*Then, the Bregman divergence $D_f(x, y)$ is strongly convex in $x$, with convexity parameter $\mu$.*

*Proof of Lemma 2.* Since $f$ is $\mu$-strongly convex, for any $x_1, x_2 \in \mathbb{R}^n$ and $\lambda \in [0, 1]$, we have

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2) - \frac{\mu}{2}\lambda(1-\lambda)\|x_1 - x_2\|^2.$$

Since the gradient is linear, for all $y \in \mathbb{R}^n$,

$$\langle \nabla f(y), \lambda x_1 + (1-\lambda)x_2 - y\rangle = \lambda\langle \nabla f(y), x_1 - y\rangle + (1-\lambda)\langle \nabla f(y), x_2 - y\rangle.$$

Using the above properties, we calculate the Bregman divergence for the convex combination $\lambda x_1 + (1-\lambda)x_2$ as follows:

$$\begin{aligned} D_f(\lambda x_1 + (1-\lambda)x_2, y) &= f(\lambda x_1 + (1-\lambda)x_2) - f(y) - \langle \nabla f(y), \lambda x_1 + (1-\lambda)x_2 - y\rangle \\ &\leq \left(\lambda f(x_1) + (1-\lambda)f(x_2) - \frac{\mu}{2}\lambda(1-\lambda)\|x_1 - x_2\|^2\right) \\ &\quad - f(y) - (\lambda\langle \nabla f(y), x_1 - y\rangle + (1-\lambda)\langle \nabla f(y), x_2 - y\rangle) \\ &= \lambda D_f(x_1, y) + (1-\lambda)D_f(x_2, y) - \frac{\mu}{2}\lambda(1-\lambda)\|x_1 - x_2\|^2. \end{aligned}$$

Thus, the Bregman divergence $D_f(x, y)$ is strongly convex in its first argument $x$ with parameter $\mu$. □

Section 3.2 aims at solving the optimization problem

$$\max_{q \in \text{conv}(\mathcal{Q})} \psi_\alpha(q), \tag{12}$$

where $\psi_\alpha(q) := \langle q, \mathcal{P} \rangle - \alpha H^* \left(\frac{q}{\alpha}\right)$. Paving the way for the proofs of the results reported in Section 3.2, we state the following auxiliary lemma establishing properties of $\psi_\alpha$ needed for the proof of Theorem 1.

**Lemma 3.** *The function $\psi_\alpha : [-1,1]^k \longrightarrow \mathbb{R}$, $\psi_\alpha$ is 2-Lipschitz and $\frac{1}{\alpha}$-smooth with respect to $\|\cdot\|_\infty$.*

*Proof of Lemma 3.* Note that $\psi_\alpha(q) := \mathbb{E}_{\mathbf{z}\sim\mathcal{P}}[\ell(q,\mathbf{z})]$ where $\ell(q,\mathbf{z}) := q(z) - \alpha H^* \left(\frac{q}{\alpha}\right)$. Let $\partial_j f$ denote the partial derivative with respect to the $j-$th entry of an $\mathbb{R}^k$-valued function $f$. Then, it holds that $\partial_j q(\mathbf{z}) = \delta_{\mathbf{z},j}$, where $\delta_{a,b} = 1$ if and only if $a = b$, and 0 else. Note that $\partial_j q(\mathbf{z})$ is integrable, such that, by the dominated convergence theorem, it follows that

$$\nabla \mathbb{E}_{\mathbf{z}\sim\mathcal{P}} \left[\ell(q,\mathbf{z})\right] = \mathbb{E}_{\mathbf{z}\sim\mathcal{P}} \left[\nabla \ell(q,\mathbf{z})\right]$$

Accordingly, it suffices to show that $\ell(\cdot,\mathbf{z})$ is $2-$Lipschitz and $\frac{1}{\alpha}-$smooth.

1. **Lipschitzness**: For $q_1, q_2 \in [-1,1]^k$, it holds that

$$\begin{aligned} |\ell(q_1,\mathbf{z}) - \ell(q_2,\mathbf{z}) &= \left| q_1(\mathbf{z}) - q_2(\mathbf{z}) + \alpha \left( H^* \left(\frac{q_2}{\alpha}\right) - H^* \left(\frac{q_1}{\alpha}\right)\right)\right| \\ &\leq \sup_{\mathbf{z}\in\mathcal{Z}} \left| q_1(\mathbf{z}) - q_2(\mathbf{z})\right| + \alpha \left\| \frac{q_2}{\alpha} - \frac{q_1}{\alpha} \right\|_\infty \leq 2\|q_1 - q_2\|_\infty \, . \end{aligned}$$

2. **Smoothness**: For $q_1, q_2 \in [-1,1]^k$, it holds that

$$\|\nabla \ell(q_1,\mathbf{z}) - \nabla \ell(q_2,\mathbf{z})\|_1 = \left\| e_{\mathbf{z}} - \nabla[H^*(q_1/\alpha)] - e_{\mathbf{z}} + \nabla[H^*(q_2/\alpha)] \right\|_1 \leq \frac{1}{\alpha} \|q_1 - q_2\|_\infty \, .$$

□

Algorithm 1 incorporates Laplace noise for guaranteeing differential privacy. For an accuracy guarantee of the algorithm, however, we need to control the effect of this mechanism. For this, we establish the following maximal inequality for i.i.d. Laplace random variables.

**Lemma 4.** *Consider $u_1, \ldots, u_k \overset{i.i.d}{\sim} Lap(\lambda)$ random variables. Then, it holds that*

$$\mathbb{E}\left[\max_{j=1,\ldots,k} u_j\right] \leq 2\lambda \log(2k) \, .$$

*Proof of Lemma 4.* For all $t > 0$,

$$\exp\left(t\mathbb{E}\left[\max_{j=1,\ldots,k} u_j\right]\right) \leq \mathbb{E}\left[\exp\left(t \max_{j=1,\ldots,k} u_j\right)\right] \leq \mathbb{E}\left[\sum_{i=1}^k \exp(tu_i)\right] = k\mathbb{E}[\exp(tu_1)].$$

The first inequality follows by Jensen's inequality applied to $\exp(\cdot)$, which is convex. Applying $\log(\cdot)$ to both sides (log is monotone and preserves the inequality)

$$\mathbb{E}\left[\max_{j=1,\ldots,k} u_j\right] \leq \frac{\log\left(k\mathbb{E}[\exp(tu_1)]\right)}{t} \quad \text{for all } t > 0.$$

Since the density function of the Laplacian distribution is known, $\mathbb{E}[\exp(tu)]$ can be computed exactly. In fact, it corresponds to the moment generating function. It is finite if $0 < t < \frac{1}{\lambda}$, since

$$\mathbb{E}[\exp(tu)] = \frac{1}{(1+t\lambda)(1-\lambda t)} \leq \frac{1}{1-\lambda t} \quad \text{for all } 0 < t < \frac{1}{\lambda}.$$

By choosing $t = \frac{1}{2\lambda}$, we finally get the upper bound $\mathbb{E}\left[\max_{j=1,\ldots,k} u_j\right] \leq 2\lambda \log(2k)$ . □

Section 3.3 aims at solving the unregularized optimization problem

$$\min_{\mathcal{D}\in\Delta_k} \phi(\mathcal{D}), \text{ where } \phi(\mathcal{D}) := \max_{q\in \mathrm{conv}(\mathcal{Q})} \langle q, \mathcal{P}-\mathcal{D}\rangle. \tag{$P$}$$

The following auxiliary Lemma characterizes this problem as a convex optimization problem.

**Lemma 5.** *The function $\phi : \Delta_k \mapsto \mathbb{R}_+$, defined by $\phi(\mathcal{D})$, is convex and 1-Lipschitz with respect to $\|\cdot\|_1$.*

*Proof of Lemma 5.* As $\phi$ is defined as the maximum of linear functions, it is convex. Lipschitzness can be proved by the triangle inequality: For any $\mathcal{D}, \mathcal{D}' \in \Delta_k$ it holds that

$$\phi(\mathcal{D}) - \phi(\mathcal{D}') \le \max_{q\in\mathrm{conv}(\mathcal{Q})} \langle q, \mathcal{D}-\mathcal{D}'\rangle \le \|\mathcal{D}-\mathcal{D}'\|_1,$$

where we used the Hölder inequality and that $q \in [-1,1]^k$ for all $q \in \mathcal{Q}$. □

In Section 3.3.2, the following approximation to the primal problem $(P_\alpha)$ is considered:

$$\min_{\mathcal{D}\in\Delta_k} \Phi_{\alpha,\sigma}(\mathcal{D}) := \min_{\mathcal{D}\in\Delta_k} \left[\phi_\sigma(\mathcal{D}) + \alpha H(\mathcal{D})\right], \tag{$P_{\alpha,\sigma}$}$$

where $\phi_\sigma(\mathcal{D}) := \int_{\mathbb{R}^k} \phi(\mathcal{D}+x)\varphi_\sigma(x)\,dx = \mathbb{E}_\xi[\phi(\mathcal{D}+\xi)]$ for a Gaussian random vector $\xi \sim \mathcal{N}(0, \sigma^2 I_k)$.

The following Proposition characterizes this problem as a convex optimization problem:

**Proposition 2.** *Let $\mathcal{Q} \subseteq [-1,1]^k$ be a compact and convex set. Then, the following holds:*

*(i)* $\|\phi_\sigma - \phi\|_\infty \le \sigma w(\mathcal{Q})$, *where* $w(\mathcal{Q}) := \mathbb{E}_{\xi\sim\varphi}\left[\max_{q\in\mathcal{Q}}\langle q,\xi\rangle\right]$ *is the Gaussian width of $\mathcal{Q}$ and $\varphi$ denotes the density of the multivariate standard normal distribution.*

*(ii)* $\phi_\sigma$ *is convex and 1-Lipschitz with respect to* $\|\cdot\|_1$.

*(iii)* $\phi_\sigma$ *is* $1/\sigma$*-smooth with respect to* $\|\cdot\|_1$.

*Proof of Proposition 2.* We prove each part separately:

(i) Let $\mathcal{D} \in \Delta_k$. Then, it holds that

$$\begin{aligned}\phi_\sigma(\mathcal{D}) - \phi(\mathcal{D}) &= \mathbb{E}_{\xi\sim\varphi_\sigma}\left[\max_{q\in Q}\langle q, \mathcal{P}-\mathcal{D}+\xi\rangle\right] - \max_{q\in Q}\langle q, \mathcal{P}-\mathcal{D}\rangle \\ &\le \mathbb{E}_{\xi\sim\varphi_\sigma}\left[\max_{q\in Q}\langle q,\xi\rangle\right] \\ &= \sigma w(Q).\end{aligned}$$

(ii) Convexity of $\phi_\sigma$ follows from Lemma 2.1 in Bertsekas [1973] due to convexity of $\phi$. Let $\mathcal{D}_1, \mathcal{D}_2 \in \mathbb{R}^k$. Then, Jensen's inequality and 1-Lipschitz continuity of $\phi$ yield

$$\begin{aligned}|\phi_\sigma(\mathcal{D}_1) - \phi_\sigma(\mathcal{D}_2)| &= |\mathbb{E}_{\xi\sim\varphi_\sigma}[\phi(\mathcal{D}_1+\xi)] - \mathbb{E}_{\xi\sim\varphi_\sigma}[\phi(\mathcal{D}_2+\xi)]| \\ &\le \mathbb{E}_{\xi\sim\varphi}[|\phi(\mathcal{D}_1+\xi) - \phi(\mathcal{D}_2+\xi)|] \\ &\le \|\mathcal{D}_1 - \mathcal{D}_2\|_1.\end{aligned}$$

Therefore, $\phi_\sigma$ is 1-Lipschitz with respect to the $\|\cdot\|_1$-norm.

(iii) Let $\xi = (\xi_1, \dots, \xi_k)^\intercal$ be a multivariate normal random vector with mean $\mathbf{0}$ and covariance matrix $\sigma^2 I_k$. According to Lemma 1.5 in Abernethy et al. [2016] $\nabla\phi_\sigma(\mathcal{D}) = \mathbb{E}_\xi[\phi(\mathcal{D}+\xi)\nabla\nu(\xi)]$, where $\nu(\xi) = \frac{\|\xi\|_2^2}{2\sigma^2}$. It then follows that

$$\begin{aligned}\|\nabla\phi_\sigma(\mathcal{D}_1) - \nabla\phi_\sigma(\mathcal{D}_2)\|_\infty &= \|\mathbb{E}_\xi\left[\big(\phi(\mathcal{D}_1+\xi) - \phi(\mathcal{D}_2\xi)\big)\nabla\nu(\xi)\right]\|_\infty \\ &= \frac{1}{\sigma^2}\max_{j\in[k]}\left|\mathbb{E}_\xi\left[\big(\phi(\mathcal{D}+\xi) - \phi(\mathcal{D}'+\xi)\big)\xi_i\right]\right| \\ &\le \sup_{x\in\mathbb{R}^k}|\phi(\mathcal{D}_1+x) - \phi(\mathcal{D}_2+x)| \cdot \max_{j\in[k]}\frac{\mathbb{E}_\xi|\xi_j|}{\sigma^2},\end{aligned}$$

where in the last step we used 1-Lipschitzness of $\phi$, and the Jensen inequality.

□

# B Proofs for the results in Section 3.2.1

This section establishes proofs for Theorems 1 and 2 in Section 3.2.1 of the main document. Section B.1 establishes the Laplace Mechanism as a tool for the incorporation of differential privacy in algorithms for synthetic data release and reviews some key properties of differentially private algorithms. Finally, in Sections B.2 and B.3, we provide the proofs for Theorem 1 and 2, respectively.

## B.1 The Laplacian mechanism and key properties of differential privacy

A common method for ensuring differential privacy involves adding noise to the output of a deterministic function $f$. In Algorithm 1, we apply such a noise-adding technique, known as the *Laplace Mechanism*, which guarantees pure differential privacy. To maintain generality, we present this mechanism in the context of a generic function $f$ and dataset $S$ (for further details, see Dwork et al. [2006]). Specifically, let $\varepsilon \geq 0$ and $S$ represent a dataset. For $k \geq 1$, let $f(S) \in \mathbb{R}^k$ be a deterministic function of $S$. The following (randomized) algorithm $\mathcal{A}$ is known as the *Laplace Mechanism*:

$$\mathcal{A}(S) = f(S) + (Y_1, \ldots, Y_k)^\intercal \ , \tag{13}$$

where $Y_1, \ldots, Y_k \overset{i.i.d}{\sim} \text{Lap}\left(\frac{\Delta(f)}{\varepsilon}\right)$ and where the quantity $\Delta(f) := \sup_{S_1 \sim S_2 S_1, S_2 \in \mathcal{Z}^n} \|f(S_1) - f(S_2)\|_{\ell^1}$ is called *sensitivity of* $f$. The supremum is taken over all neighboring datasets. The intuition behind the Laplace Mechanism is to add noise drawn from a Laplace distribution to the output of an algorithm, with the amount of noise calibrated to the algorithm's sensitivity (i.e., how much its output changes in response to a single data point). By introducing this scaled noise, the mechanism ensures that the output remains similar whether or not a particular individual is in the dataset, thus providing differential privacy. It can be shown that $\mathcal{A}$ in (13) is $(\varepsilon, 0)$ - ; see Dwork et al. [2006]. Suppose that $f(S) = (f_1(S), \ldots, f_k(S))^\intercal$, where each $f_j(S) \in \mathbb{R}$ is a function of the dataset $S$, and for all $j = 1, \ldots, k$ the sensitivity is of the form $\Delta(f_j) \leq L$ for some constant $L$. Then, the so-called Report Noisy Max (RNM) variable, defined by

$$\text{RNM}(S) = \underset{i=1,\ldots,k}{\text{argmax}} \{f_i(S) + Y_i\} \ , \tag{14}$$

where $Y_j \overset{i.i.d}{\sim} \text{Lap}\left(\frac{L}{\varepsilon}\right)$, is $(\varepsilon, 0)$-DP; see Claim 3.9 in Dwork and Roth [2014].

While the Laplace mechanism introduces differential privacy into the Differentially private Frank-Wolfe algorithm (Algorithm 1), the iterative nature of the algorithm raises the question how much privacy is lost due to composition of the individual iterations. For Algorithm 1, an answer to this question will be provided in Section B.2. It will be based on the well-known Advanced Composition Theorem:

**Theorem 6** (Theorem III.3, Dwork et al. [2010])**.** *For all $\varepsilon > 0$ and $1 > \delta > 0$ with $\log(1/\delta) \geq \varepsilon^2 T$, let $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_T)$ be a sequence of $(\varepsilon, 0) - DP$ algorithms where each $\mathcal{A}_j$ is sequentially and adaptively chosen. Then, the whole chain $\mathcal{A}$ is $(\hat{\varepsilon}, \delta)$-DP, where $\hat{\varepsilon} = 4\varepsilon\sqrt{2T\log(1/\delta)}$ .*

We conclude this section by presenting another key property of Differential Privacy we will rely on: the *post-processing*: If $\mathcal{A}$ is $(\varepsilon, \delta)$-DP and $f : \mathcal{O} \longrightarrow \mathcal{O}'$ is any deterministic function, then the composition $f \circ \mathcal{A} : \mathcal{Z}^n \longrightarrow \mathcal{O}'$ is also $(\varepsilon, \delta)$-DP. In other words, no matter what operations or transformations are applied to the outputs of a differentially private algorithm, the privacy guarantees provided by $\mathcal{A}$ will still hold; see Proposition 2.1 in Dwork and Roth [2014].

## B.2 Proof of Theorem 1: Differential privacy and accuracy of Algorithm 1

First, we establish differential privacy of the algorithm. For this, let $S_n = \{z_1, \ldots, z_n\} \overset{i.i.d}{\sim} \mathcal{P}$ and $\mathcal{Q} = \{q^{(1)}, \ldots, q^{|\mathcal{Q}|}\}$. Recall that $q_{t+1} = q_t + \gamma(s_t - q_t)$, where $s_t$ is defined by line 9 of Algorithm 1. Within each iteration of the algorithm, we use Report Noisy Max on the vector $\left(\langle\hat{\nabla}\psi_\alpha(q_t), q^{(1)}\rangle, \ldots, \langle\hat{\nabla}\psi_\alpha(q_t), q^{(|\mathcal{Q}|)}\rangle\right)^\intercal$, i.e.

$$\mathcal{A}(S_n) := \begin{pmatrix} \langle\hat{\nabla}\psi_\alpha(q_t), q^{(1)}\rangle \\ \vdots \\ \langle\hat{\nabla}\psi_\alpha(q_t), q^{(|\mathcal{Q}|)}\rangle \end{pmatrix} + \begin{pmatrix} u_{1,t} \\ \vdots \\ u_{|\mathcal{Q}|,t} \end{pmatrix} \quad \text{with} \quad u_{i,t} \overset{i.i.d}{\sim} \text{Lap}\,(\lambda) \ ,$$

where $\hat{\nabla}\psi_\alpha(q_t) = \mathcal{P}_n - \nabla H^*\left(\frac{q_t}{\alpha}\right) = \frac{1}{n}\sum_{i=1}^n e_{z_i} - \nabla H^*\left(\frac{q_t}{\alpha}\right)$ . For a neighboring dataset $S_n' = \{z_1', \dots, z_n'\}$ of $S_n$, the corresponding $\hat{\nabla}\psi_\alpha'(q_t) = \frac{1}{n}\sum_{i=1}^n e_{z_i'} - \nabla H^*\left(\frac{q_t}{\alpha}\right) = \mathcal{P}_n' - \nabla H^*\left(\frac{q_t}{\alpha}\right)$. Due to Claim 3.9 in Dwork and Roth [2014] discussed in Section B.1, for the algorithm to be $(\varepsilon, 0)$-DP, the parameter $\lambda$ must correspond to the maximum sensitivity of a single iteration within the algorithm. We consider the $i-$th iteration, i.e. the $i$-th entry of $\mathcal{A}(S_n)$, and define its deterministic part $f_i(S_n) := \langle q^{(i)}, \hat{\nabla}\psi_\alpha(q_t)\rangle$, while for a neighboring set $S_n'$, we define $f_i(S_n') := \langle q^{(i)}, \hat{\nabla}\psi_\alpha'(q_t)\rangle$. The corresponding sensitivity is

$$\begin{aligned}\Delta(f_i(S_n)) =& \sup_{S_n \sim S_n'} \left|\langle q^{(i)}, \hat{\nabla}\psi_\alpha'(q_t) - \hat{\nabla}\psi_\alpha(q_t)\rangle\right| \\ \leq& D_1 \sup_{S_n \sim S_n'} \left\| \frac{1}{n}\sum_{z_i \in S_n} e_{z_i} - \nabla H^*\left(\frac{q_t}{\alpha}\right) - \frac{1}{n}\sum_{z_i' \in S_n'} e_{z_i'} + \nabla H^*\left(\frac{q_t}{\alpha}\right)\right\|_\infty = \frac{D_1}{n} =: L\,.\end{aligned}$$

As a result, since the sensitivity of each entry $f_j$ is upper bounded by a universal constant $L$, independent of $j$, if we want the Algorithm to be $(\tilde{\varepsilon}, 0) -$ DP, we need $\lambda = \frac{D_1}{n\tilde{\varepsilon}}$. Since it performs $T$ iterations, the Advanced Composition Theorem, i.e. Theorem 6 in Section B.1, yields a privacy guarantee of

$$\left(4\tilde{\varepsilon}\sqrt{2T\log\left(\frac{1}{\delta}\right)}, \delta\right) \text{- DP}\,.$$

Hence, in order to make the output $(\varepsilon, \delta)$-DP, $\tilde{\varepsilon}$ needs to be chosen as $\tilde{\varepsilon} = \varepsilon\left(4\sqrt{2T\log\left(\frac{1}{\delta}\right)}\right)^{-1}$ . This concludes the privacy analysis.

In what follows, we establish the accuracy bound provided by Theorem 1. To simplify the analysis and follow the standard proof structure for the convergence of the Frank-Wolfe algorithm, we focus on $\mathcal{J}(q) := -\psi_\alpha(q)$, which is strongly convex. Note that this means that we replace line 9 of Algorithm 1 with

$$s_t = \operatorname*{argmin}_{s \in \text{conv}(\mathcal{Q})} \{\langle \hat{\nabla}\mathcal{J}(q_t), s\rangle + u_{s,t}\}.$$

(For this, note that $u_{s,t} \overset{\mathcal{D}}{=} -u_{s,t}$ by symmetry of the centered Laplace distribution.)

Our goal is to bound the Frank-Wolfe gap

$$g_{\mathcal{J}}(q) := \max_{s \in \text{conv}(\mathcal{Q})} \langle \nabla\mathcal{J}(q), q - s\rangle.$$

For this, we leverage the descent lemma (which holds due to the $L_1$-smoothness of $\psi_\alpha$)

$$\begin{aligned}\mathcal{J}(q_{t+1}) \leq& \mathcal{J}(q_t) + \langle \nabla\mathcal{J}(q_t), q_{t+1} - q_t\rangle + \frac{1}{2\alpha}\|q_{t+1} - q_t\|_\infty^2 \quad (\mathcal{J} \text{ is } \frac{1}{\alpha}-\text{smooth for Lemma 3, and descent lemma}) \\ \leq& \mathcal{J}(q_t) + \gamma\langle \nabla\mathcal{J}(q_t), s_t - q_t\rangle + \frac{\gamma^2 D_\infty^2}{2\alpha} \quad (\text{Definition of } q_{t+1}\,,\ \text{line 10 of Algorithm 1}) \\ =& \mathcal{J}(q_t) + \gamma\langle \nabla\mathcal{J}(q_t), s_t - p_t\rangle + \gamma\langle \nabla\mathcal{J}(q_t), p_t - q_t\rangle + \frac{\gamma^2 D_\infty^2}{2\alpha} \\ =& \mathcal{J}(q_t) + \gamma\langle \nabla\mathcal{J}(q_t), s_t - p_t\rangle - \gamma g_{\mathcal{J}}(q_t) + \frac{\gamma^2 D_\infty^2}{2\alpha}\,,\end{aligned}$$

where

$$p_t = \operatorname*{argmin}_{s \in \text{conv}(\mathcal{Q})} \langle \nabla\mathcal{J}(q_t), s - q_t\rangle\,.$$

As a next step, we introduce the auxiliary variable $V_t$ defined as follows:

$$V_t := \max_{s \in \mathcal{Q}} u_{s,t} - \min_{s \in \mathcal{Q}} u_{s,t}.$$

Since, by definition of $s_t$, for all $q \in \text{conv}(\mathcal{Q})$, for all $t = 1, \dots, T$, and for all $s \in \mathcal{Q}$

$$\langle \hat{\nabla}\mathcal{J}(q), s_t \rangle + \min_{s\in\mathcal{Q}} u_{s,t} \leq \langle \hat{\nabla}\mathcal{J}(q), s_t \rangle + u_{s,t} \leq \langle \hat{\nabla}\mathcal{J}(q), s \rangle + \max_{s\in\mathcal{Q}} u_{s,t} \ ,$$

it holds that

$$\langle \hat{\nabla}\mathcal{J}(q), s_t - s \rangle \leq V_t \ \text{ a.s. } \ \forall s \in \text{conv}(\mathcal{Q}). \tag{15}$$

It therefore follows that

$$\begin{aligned}
\mathcal{J}(q_{t+1}) \leq& \mathcal{J}(q_t) - \gamma g_{\mathcal{J}}(q_t) + \gamma \langle \nabla\mathcal{J}(q_t), s_t - p_t \rangle + \frac{\gamma^2 D_\infty^2}{2\alpha} \\
=& \mathcal{J}(q_t) - \gamma g_{\mathcal{J}}(q_t) + \gamma \langle \nabla\mathcal{J}(q_t) - \hat{\nabla}\mathcal{J}(q_t), s_t - p_t \rangle + \gamma \langle \hat{\nabla}\mathcal{J}(q_t), s_t - p_t \rangle + \frac{\gamma^2 D_\infty^2}{2\alpha} \\
\leq& \mathcal{J}(q_t) - \gamma g_{\mathcal{J}}(q_t) + \gamma \langle \nabla\mathcal{J}(q_t) - \hat{\nabla}\mathcal{J}(q_t), s_t - p_t \rangle + \gamma V_t + \frac{\gamma^2 D_\infty^2}{2\alpha}.
\end{aligned}$$

Rearranging the inequality yields

$$g_{\mathcal{J}}(q_t) \leq \frac{1}{\gamma}(\mathcal{J}(q_t) - \mathcal{J}(q_{t+1})) + \left( \|\nabla\mathcal{J}(q_t) - \hat{\nabla}\mathcal{J}(q_t)\|_\infty \|s_t - p_t\|_1 \right) + V_t + \frac{\gamma D_\infty^2}{2\alpha},$$

which implies

$$\begin{aligned}
\sum_{t=0}^{T-1} g_{\mathcal{J}}(q_t) \leq& \frac{1}{\gamma} \sum_{t=0}^{T-1} (\mathcal{J}(q_t) - \mathcal{J}(q_{t+1})) + D_1 \sum_{t=0}^{T-1} \|\nabla\mathcal{J}(q_t) - \hat{\nabla}\mathcal{J}(q_t)\|_\infty + \sum_{t=0}^{T-1} V_t + \frac{\gamma D_\infty^2 T}{2\alpha} \\
=& \frac{1}{\gamma}\mathcal{J}(q_0) - \mathcal{J}(q_T) + D_1 \sum_{t=0}^{T-1} \|\nabla\mathcal{J}(q_t) - \hat{\nabla}\mathcal{J}(q_t)\|_\infty + \sum_{t=0}^{T-1} V_t + \frac{\gamma D_\infty^2 T}{2\alpha} \ .
\end{aligned}$$

Moreover, note that

$$\|\hat{\nabla}\mathcal{J}(q_t) - \nabla\mathcal{J}(q_t)\|_\infty = \left\| \frac{1}{n}\sum_{i=1}^n e_{z_i} - \nabla H^*\left(\frac{q_t}{\alpha}\right) - \mathbb{E}_{\mathbf{z}\sim\mathcal{P}}[e_{\mathbf{z}}] + \nabla H^*\left(\frac{q_t}{\alpha}\right) \right\|_\infty = \left\| \frac{1}{n}\sum_{i=1}^n e_{z_i} - \mathbb{E}_{\mathbf{z}\sim\mathcal{P}}[e_{\mathbf{z}}] \right\|_\infty \ .$$

Further, since $\mathcal{J}$ is 2-Lipschitz due to Lemma 3, $\mathcal{J}(q_0) - \mathcal{J}(q_t) \leq 2\|q_0 - q_t\| \leq 2D_\infty$, and we can simplify the above expression to

$$\frac{1}{T}\sum_{t=0}^{T-1} g_{\mathcal{J}}(q_t) \leq \frac{2D_\infty}{\gamma T} + D_1 \left\| \frac{1}{n}\sum_{i=1}^n e_{z_i} - \mathbb{E}_{\mathbf{z}\sim\mathcal{P}}[e_{\mathbf{z}}] \right\|_\infty + \frac{1}{T}\sum_{t=0}^{T-1} V_t + \frac{D_\infty^2 \gamma}{2\alpha} \ .$$

Considering the expectation of the previous expression, we obtain

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} g_{\mathcal{J}}(q_t)\right] \leq \frac{2D_\infty}{\gamma T} + D_1 \mathbb{E}\left\| \frac{1}{n}\sum_{i=1}^n e_{z_i} - \mathbb{E}_{\mathbf{z}\sim\mathcal{P}}[e_{\mathbf{z}}] \right\|_\infty + \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[V_t] + \frac{1}{2\alpha} D_\infty^2 \gamma \ .$$

It follows from Corollary 2.3 in Dümbgen et al. [2010] that $\mathbb{E}\left[\|\frac{1}{n}\sum_{i=1}^n e_{z_i} - \mathbb{E}_{\mathbf{z}\sim\mathcal{P}}[e_{\mathbf{z}}]\|_\infty\right] \leq \mathcal{O}\left(\sqrt{\log k/n}\right)$. Additionally, according to Lemma 4, for all $t = 0, \dots, T-1$, it holds that $\mathbb{E}[V_t] \leq 2\lambda \log(2|\mathcal{Q}|)$. We conclude that

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} g_{\mathcal{J}}(q_t)\right] \leq \frac{2D_\infty}{\gamma T} + D_1 \mathcal{O}\left(\sqrt{\frac{\log k}{n}}\right) + 2\lambda \log 2|\mathcal{Q}| + \frac{1}{2\alpha} D_\infty^2 \gamma \ .$$

Recall that $\lambda = \frac{D_1}{n\tilde{\varepsilon}} = \frac{4\sqrt{2}\sqrt{T\log(1/\delta)}}{\varepsilon n}$. Minimizing the right hand side of the above inequality with respect to $\gamma$, we find that the minimizer $\gamma^* = 2\sqrt{\frac{\alpha}{TD_\infty}}$ yields

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} g_{\mathcal{J}}(q_t)\right] \leq 2D_\infty^{3/2}\sqrt{\frac{1}{\alpha T}} + D_1 \mathcal{O}\left(\sqrt{\frac{\log k}{n}}\right) + \frac{8\sqrt{2}\log(2|\mathcal{Q}|)\sqrt{T\log(1/\delta)}}{\varepsilon n} \ .$$

Finally, minimizing the right-hand side of the above inequality with respect to $T$ yields $T^* = \frac{D_\infty^{3/2}\varepsilon n}{\sqrt{32\alpha\log(1/\delta)\log(2|\mathcal{Q}|)}}$ as minimizer and

$$\mathbb{E}[g_{\mathcal{L}}(q_\alpha^{\text{out}}))] = \mathcal{O}\left(\frac{D_\infty^{3/4}\log^{1/4}(1/\delta)\log^{1/2}(|\mathcal{Q}|)}{\alpha^{1/4}(\varepsilon n)^{1/2}} + \frac{D_1\log^{1/2}(k)}{\sqrt{n}}\right),$$

where $q_\alpha^{\text{out}}$ denotes the output of the Frank-Wolfe Algorithm.

### B.3 Proof of Theorem 2: Upper bound on primal gap

This section is structured as follows. We begin by proving a key technical result (Lemma 3), which forms the foundation for the proof of Theorem 2. Theorem 2 then establishes an upper bound for the accuracy of

$$\mathcal{P}_\alpha^{\text{priv}} = \underset{\mathcal{D}\in\Delta_k}{\operatorname{argmin}}\left(\langle q_\alpha^{\text{out}}, \mathcal{P}-\mathcal{D}\rangle + \alpha H(\mathcal{D})\right).$$

Lastly, we conclude by presenting the proof of Theorem 3.

**Lemma 6.** *Consider the negative entropy function $H$ on $\Delta_k$, $H(\mathcal{D}) = \sum_{1\le i\le k}\mathcal{D}(i)\log(\mathcal{D}(i))$, and the Bregman divergence $D_H(\mathcal{D},\mathcal{D}')$. Moreover, suppose that $\mathcal{D}'\in\Delta_k$ has only strictly positive coefficients. For $A\in\mathbb{R}$, $B>0$, $C\ge 0$ and vectors $g\in\mathbb{R}^k$, the following problem*

$$\min_{\mathcal{D}\in\Delta_k}\Psi_{A,B,C}(\mathcal{D}), \text{ where } \Psi_{A,B,C}(\mathcal{D}) := A\langle g,\mathcal{D}\rangle + BH(\mathcal{D}) + CD_H(\mathcal{D},\mathcal{D}')$$

*has an unique solution $\mathcal{D}_*\in\Delta_k$ and*

$$\mathcal{D}_* \propto \exp\Big(\frac{-Ag_i - B + C\log\mathcal{D}'(i)}{B+C}\Big).$$

*Proof of Lemma 6.* Note that the first summand in the representation of $\Psi_{A,B,C}$ is linear, while the second summand corresponds to the negative entropy function which is strongly convex according to Example 2.5 in Shalev-Shwartz [2012]. Moreover, recall that Lemma 2 establishes strong convexity $D_H(\cdot,D')$. As a result, $\Psi_{A,B,C}$ is strongly convex, as well. Therefore, there is only an unique minimum $\mathcal{D}$ in $(\mathbb{R}_+)^k$ which is also the stationary point i.e.

$$\nabla\Psi_{A,B,C}(\mathcal{D}) = Ag + B\nabla H(\mathcal{D}) + C[\nabla H(\mathcal{D}) - \nabla H(\mathcal{D}')] = 0. \tag{16}$$

We evaluate equation (16) in its vector form. For all $1\le i\le k$,

$$Ag_i + (B+C)\Big[1+\log\mathcal{D}(i)\Big] - C\Big(1+\log\mathcal{D}'(i)\Big) = 0,$$

which yields

$$\begin{aligned}(B+C)\log\mathcal{D}(i) &= -Ag_i - B + C\log\mathcal{D}'(i)\\ \mathcal{D}(i) &= \exp\left(\frac{-Ag_i - B + C\log\mathcal{D}'(i)}{B+C}\right).\end{aligned}$$

For the optimality in $\Delta_k$, we will study $\Psi(\lambda\mathcal{P})$ for any $\lambda\in\mathbb{R},\mathcal{P}\in(\mathbb{R}_+)^k$. We notice that $H(\lambda\mathcal{P}) = \lambda H(\mathcal{P}) + \lambda\log(\lambda)\sum_i\mathcal{P}(i)$, therefore

$$\begin{aligned}\Psi_{A,B,C}(\lambda\mathcal{P}) &= A\langle g,\lambda\mathcal{P}\rangle + BH(\lambda\mathcal{P}) + C[H(\lambda\mathcal{P}) - H(\mathcal{D}') - \langle\nabla H(\mathcal{D}'),\lambda\mathcal{P}-\mathcal{D}'\rangle]\\ &= \lambda\Big[A\langle g,\mathcal{P}\rangle + BH(\mathcal{P}) + C[H(\mathcal{P}) - H(\mathcal{D}') - \langle\nabla H(\mathcal{D}'),\mathcal{P}-\mathcal{D}'\rangle]\Big]\\ &+ B\lambda\log(\lambda)\sum_i\mathcal{P}(i) + C\lambda\log(\lambda)\sum_i\mathcal{P}(i) + C(\lambda-1)H(\mathcal{D}') - C(\lambda-1)\langle\nabla H(\mathcal{D}'),\mathcal{D}'\rangle\\ &= \lambda\Psi_{A,B,C}(\mathcal{P}) + (B+C)\lambda\log(\lambda)\Big(\sum_i\mathcal{P}(i)\Big) + C(\lambda-1)\Big(H(\mathcal{D}') - \langle\nabla H(\mathcal{D}'),\mathcal{D}'\rangle\Big)\end{aligned}$$

We notice that the second term scales with $\sum_i \mathcal{P}(i)$ and the last one is independent of $\mathcal{P}$. Now we consider $\mathcal{D}_* \in \Delta_k$

$$\mathcal{D}_*(i) := \frac{\mathcal{D}(i)}{\sum_{j=1}^k \mathcal{D}(j)}, \quad \lambda_* := \frac{1}{\sum_{j=1}^k \mathcal{D}(j)}$$

We know that for all $\mathcal{P} \in \frac{1}{\lambda_*}\Delta_k$ (i.e. $\lambda_*\mathcal{P} \in \Delta_k$)

$$\begin{aligned}\Psi_{A,B,C}(\lambda_*\mathcal{P}) - \Psi_{A,B,C}(\mathcal{D}_*) &= \Psi_{A,B,C}(\lambda_*\mathcal{P}) - \Psi_{A,B,C}(\lambda_*\mathcal{D}) \\ &= \lambda_*\Big(\Psi_{A,B,C}(\mathcal{P}) - \Psi_{A,B,C}(\mathcal{D})\Big) \geq 0\end{aligned}$$

with equality only if $\mathcal{P} = \mathcal{D}$. □

**Corollary 3.** Given the assumptions of Theorem 6, consider

$$p = \operatorname*{argmin}_{\mathcal{D}\in\Delta_k} \left[\langle q, -\mathcal{D}\rangle + \alpha H(\mathcal{D})\right] . \tag{17}$$

Then, it holds that

1. $\langle q, p\rangle = \alpha H(p) + \alpha H^*\left(\frac{q}{\alpha}\right)$ ,
2. $p = \nabla H^*\left(\frac{q}{\alpha}\right)$ .

*Proof of Corollary 3.* 1. Note that

$$\min_{\mathcal{D}\in\Delta_k} \left[\langle q, -\mathcal{D}\rangle + \alpha H(\mathcal{D})\right] = -\alpha \max_{\mathcal{D}\in\Delta_k} \left[\langle \frac{q}{\alpha}, \mathcal{D}\rangle - H(\mathcal{D})\right] = -\alpha H^*\left(\frac{q}{\alpha}\right) .$$

By definition, the argmin $p = (p_1, \ldots, p_k)$ is the value that attains the minimum, and due to strong convexity, this value is unique.

2. Using the properties of the Fenchel conjugate, for all $\mathcal{D} \in \Delta_k$ it holds $\mathcal{D} = \nabla H^*\left(\nabla H(\mathcal{D})\right)$ . Therefore, we only have to compute $\nabla H$, evaluated at the optimum $p$. Since $p$ is by definition the optimum, it must satisfy the Karush–Kuhn–Tucker (KKT) conditions for (17). Recall that $\mathcal{D} = (\mathcal{D}(1), \mathcal{D}(2), \ldots, \mathcal{D}(k))$. The Lagrangian associated to (17) is

$$L(\mathcal{D}, \lambda_1, \ldots, \lambda_k, \mu) = -\sum_{j=1}^k q_j\mathcal{D}(j) + \alpha H(\mathcal{D}) + \mu\left(1 - \sum_{j=1}^k \mathcal{D}(j)\right) - \sum_{j=1}^k \lambda_j\mathcal{D}(j) .$$

From Lemma 6, for all $j = 1, \ldots, k$, $p_j$ are strictly positive and $\lambda_j = 0$). Therefore, $\nabla_{\mathcal{D}} L(\mathcal{D}, \mu) := \nabla_{\mathcal{D}} L(\mathcal{D}, 0, \ldots, 0, \mu)$ evaluated at the optimum $p$,

$$\nabla_{\mathcal{D}} L(p, \mu) = -q + \alpha\nabla H(p) - \mu\mathbf{1} = 0 ,$$

where $\mathbf{1} = [1, \ldots, 1]$. Therefore, $\nabla H(p) = \frac{q}{\alpha} + \frac{\mu\mathbf{1}}{\alpha}$ , and $p = \nabla H^*\left(\nabla H(p)\right) = \nabla H^*\left(\frac{q}{\alpha} + \frac{\mu\mathbf{1}}{\alpha}\right)$ . The last step is to show that $\nabla H^*\left(\frac{q}{\alpha} + \frac{\mu\mathbf{1}}{\alpha}\right) = \nabla H^*\left(\frac{q}{\alpha}\right)$ . The Fenchel conjugate of the negative entropy is $H^*(y_1, \ldots, y_k) = \log\left(\sum_{j=1}^k e^{y_j}\right)$ , and its gradient $\nabla H^*(y_1, \ldots, y_k) = \left(\sum_{j=1}^k e^{y_j}\right)^{-1} (e^{y_1}, \ldots, e^{y_k})^\intercal$ . Evaluating at $\frac{q}{\alpha} + \frac{\mu\mathbf{1}}{\alpha}$ we get $\nabla H^*(\frac{q_1}{\alpha} + \frac{\mu}{\alpha}, \ldots, \frac{q_k}{\alpha} + \frac{\mu}{\alpha}) = \left(\sum_{j=1}^k e^{q_j/\alpha}\right)^{-1} e^{-\mu/\alpha} \cdot e^{+\mu/\alpha}\left(e^{q_1/\alpha}, \ldots, e^{q_k/\alpha}\right)^\intercal = \nabla H^*\left(\frac{q}{\alpha}\right)$ .

□

*Proof of Theorem 2.* By definition $\mathbf{Gap}_{(P)}\left(\mathcal{P}_\alpha^{\text{priv}}\right) := \phi(\mathcal{P}_\alpha^{\text{priv}}) - \phi^*$, where $\phi^* = \min_{\mathcal{D}\in\Delta_k} \phi(\mathcal{D})$ . Note that for all $\mathcal{D} \in \Delta_k$, it holds that $\phi_\alpha(\mathcal{D}) = \phi(\mathcal{D}) + \alpha H(\mathcal{D})$, and since $H(\mathcal{D}) \leq 0$, $\phi_\alpha^* = \min_{\mathcal{D}\in\Delta_k} \phi_\alpha(\mathcal{D}) \leq \min_{\mathcal{D}\in\Delta_k} \phi(\mathcal{D}) = \phi^*$ . Therefore,

$$\mathbf{Gap}_{(P)}(\mathcal{P}_\alpha^{\text{priv}}) = \phi(\mathcal{P}_\alpha^{\text{priv}}) - \phi^* \leq \phi_\alpha(\mathcal{P}_\alpha^{\text{priv}}) - \alpha H(\mathcal{P}_\alpha^{\text{priv}}) - \phi_\alpha^* .$$

Next, the positive entropy $-H(\mathcal{D})$ attains its maximal value $\log(k)$ for $\mathcal{D}$ with $\mathcal{D}(z) = \frac{1}{k}$ for all $z \in \mathcal{Z}$ .

It follows that

$$\begin{aligned}
\mathbf{Gap}_{(P)}(\mathcal{P}_\alpha^{\text{priv}}) \leq & \phi_\alpha(\mathcal{P}_\alpha^{\text{priv}}) - \phi_\alpha^* - \alpha H(\mathcal{P}_\alpha^{\text{priv}}) \\
\leq & \phi_\alpha(\mathcal{P}_\alpha^{\text{priv}}) - \phi_\alpha^* + \alpha \log(k) \\
= & \phi_\alpha(\mathcal{P}_\alpha^{\text{priv}}) - \psi_\alpha^* + \alpha \log(k) \quad \text{(strong duality)} \\
= & \phi_\alpha(\mathcal{P}_\alpha^{\text{priv}}) - \psi_\alpha(q_\alpha^{\text{out}}) + \psi_\alpha(q_\alpha^{\text{out}}) - \psi_\alpha^* + \alpha \log(k) \\
\leq & \phi_\alpha(\mathcal{P}_\alpha^{\text{priv}}) - \psi_\alpha(q_\alpha^{\text{out}}) + \alpha \log(k) \ .
\end{aligned}$$

It remains to deal with the term $\phi_\alpha(\mathcal{P}_\alpha^{\text{priv}}) - \psi_\alpha(q_\alpha^{\text{out}})$. Note that we have analytical expressions for both $\phi_\alpha(\cdot)$ and $\psi_\alpha(\cdot)$ . For all $\mathcal{D} \in \Delta_k$ and $q \in \text{conv}(\mathcal{Q})$,

$$\phi_\alpha(\mathcal{D}) = \langle \tilde{q}_\mathcal{D}, \mathcal{P} - \mathcal{D} \rangle + \alpha H(\mathcal{D}) \quad \text{and} \quad \psi_\alpha(q) = \langle q, \mathcal{P} \rangle - \alpha H^* \left( \frac{q}{\alpha} \right) \ ,$$

where $\tilde{q}_\mathcal{D}$ is the vector $\tilde{q}_\mathcal{D} = \underset{\mathcal{D} \in \Delta_k}{\text{argmax}} \left( \langle q, \mathcal{P} - \mathcal{D} \rangle + \alpha H(\mathcal{D}) \right)$. Then,

$$\begin{aligned}
\phi_\alpha(\mathcal{P}_\alpha^{\text{priv}}) - \psi_\alpha(q_\alpha^{\text{out}}) = & \langle \tilde{q}_{\mathcal{P}_\alpha^{\text{priv}}}, \mathcal{P} - \mathcal{P}_\alpha^{\text{priv}} \rangle + \alpha H(\mathcal{P}_\alpha^{\text{priv}}) - \langle q_\alpha^{\text{out}}, \mathcal{P} \rangle + \alpha H^* \left( \frac{q_\alpha^{\text{out}}}{\alpha} \right) \\
= & \langle \tilde{q}_{\mathcal{P}_\alpha^{\text{priv}}}, \mathcal{P} - \mathcal{P}_\alpha^{\text{priv}} \rangle - \langle q_\alpha^{\text{out}}, \mathcal{P} \rangle + \langle \mathcal{P}_\alpha^{\text{priv}}, q_\alpha^{\text{out}} \rangle \quad \text{(Corollary 3)} \\
= & \langle \tilde{q}_{\mathcal{P}_\alpha^{\text{priv}}} - q_\alpha^{\text{out}}, \mathcal{P} - \mathcal{P}_\alpha^{\text{priv}} \rangle \\
= & \left\langle \tilde{q}_{\mathcal{P}_\alpha^{\text{priv}}} - q_\alpha^{\text{out}}, \mathcal{P} - \nabla H^* \left( \frac{q_\alpha^{\text{out}}}{\alpha} \right) \right\rangle \quad \text{(Corollary 3)} \\
= & \left\langle \tilde{q}_{\mathcal{P}_\alpha^{\text{priv}}} - q_\alpha^{\text{out}}, \nabla \psi_\alpha(q_\alpha^{\text{out}}) \right\rangle \\
\leq & \max_{q \in \text{conv}(\mathcal{Q})} \left\langle q - q_\alpha^{\text{out}}, \nabla \psi_\alpha(q_\alpha^{\text{out}}) \right\rangle =: g_{\psi_\alpha} \left( q_\alpha^{\text{out}} \right) \ .
\end{aligned}$$

□

*Proof of Theorem 3.* A combination of the utility guarantee from Theorem 1 and the upper bound of the primal gap from Theorem 2 yields

$$\mathbb{E}\left[ \mathbf{Gap}_{(P)}(\mathcal{P}_\alpha^{\text{priv}}) \right] = \mathcal{O}\left( \frac{\log^{1/4}(1/\delta) \log^{1/2}(|\mathcal{Q}|)}{\alpha^{1/4} (\varepsilon n)^{1/2}} + D_1 \sqrt{\frac{\log(k)}{n}} \right) + \alpha \log k \ .$$

Optimizing the right hand side of the previous expression with respect to $\alpha$ yields the desired result. Finally, note that the $\mathbf{Gap}_{(P)}(\mathcal{P}_{\alpha^*}^{\text{priv}}) = \max_{q \in \mathcal{Q}} \langle q, \mathcal{P} - \mathcal{P}^{\text{priv}} \rangle$ since $\phi^* = 0$ . □

## C Proofs for the results in Section 3.3

This section introduces the concept of Rényi differential privacy, establishes related key technical results from optimization theory and provides proofs for Theorems 4 and 5

### C.1 The Gaussian Mechanism and key properties of Rényi Differential Privacy

The proof of Theorem 5 builds upon the definition of Rényi Differential Privacy, which we introduce here along with some of its key properties. We also describe the widely-used Gaussian mechanism, central to the privacy analysis of our algorithm. For a more comprehensive discussion of these topics, we refer the reader to Mironov [2017] and Dwork and Roth [2014]

**Definition 2.** For two probability distributions $f$ and $g$ supported over $\mathcal{R}$, the Rényi divergence of order $\beta > 1$ is defined as

$$D_\beta(f\|g) := \frac{1}{\beta-1}\log \mathrm{E}_{x\sim g}\left[\left(\frac{f(x)}{g(x)}\right)^\beta\right].$$

**Definition 3.** ($(\beta,\epsilon)$-RDP). A randomized mechanism $f : \mathcal{Z}^n \mapsto \mathcal{R}$ is said to be $\epsilon$-Rényi differentially private of order $\beta$, (denoted as $(\beta,\epsilon)$-RDP), if

$$\sup_{S_1\sim S_2} D_\beta\left(f(S_1)\|f\left(S_2\right)\right) \leq \epsilon\,. \tag{18}$$

Analogous to the Laplacian Mechanism, the so-called Gaussian Mechanism can be defined on the basis of a function $f : \mathcal{Z}^n \longrightarrow \mathbb{R}^k$ and a dataset $S$ by

$$\mathcal{A}(S) = f(S) + (Y_1,\ldots,Y_k)^\intercal\,, \tag{19}$$

where $Y_j \overset{i.i.d}{\sim} \mathcal{N}\left(0, 2\log(1.25/\delta)\Delta_2^2(f)/\varepsilon^2\right)$ , where the quantity $\Delta_2(f) = \max_{S_1\sim S_2}\|f(S_1) - f(S_2)\|_2$ is called $\ell^2-$*sensitivity of* $f$, and where the supremum is taken over all neighboring datasets.

For any choice of $\varepsilon$ and $\delta$, the Gaussian Mechanism satisfies $(\varepsilon,\delta)$-differential privacy (see Appendix A of Dwork and Roth [2014]). Moreover, it can be shown that the Gaussian Mechanism achieves privacy in the sense of Rényi Differential Privacy (RDP). More specifically, for any $\beta > 1$ and $\sigma > 0$,

$$\mathcal{A}(S) = f(S) + (Y_1,\ldots,Y_k)^\intercal\,, \quad Y_j \overset{i.i.d}{\sim} \mathcal{N}(0,\sigma^2)\,, \tag{20}$$

is $\left(\beta, \frac{1}{\sigma^2}\frac{\beta}{2}\Delta_2^2(f)\right)$-RDP. In order to prove (20), we will utilize the following lemma, which outlines a well-known property of Gaussian distributions.

**Lemma 7.** *For $\mu_1,\mu_2 \in \mathbb{R}^k$ and $\Sigma$ a positive definite matrix of size $k$, let $f \sim \mathcal{N}(\mu_1,\Sigma)$ and $g \sim \mathcal{N}(\mu_2,\Sigma)$. Then, for any $\beta > 0$ the following holds*

$$D_\beta(f\,||\,g) = \frac{\beta}{2}(\mu_1-\mu_2)^\intercal\Sigma^{-1}(\mu_1-\mu_2)\,.$$

*Proof.* For $f \sim \mathcal{N}(\mu_1,\Sigma)$ and $g \sim \mathcal{N}(\mu_2,\Sigma)$ it holds that

$$\begin{aligned}
\frac{f(x)}{g(x)} &= \exp\left(-\frac{1}{2}\left((x-\mu_1)^\intercal\Sigma^{-1}(x-\mu_1) - (x-\mu_2)^\intercal\Sigma^{-1}(x-\mu_2)\right)\right)\\
&= \exp\left(-\frac{1}{2}\left(x^\intercal\Sigma^{-1}x - 2\mu_1^\intercal\Sigma^{-1}x + \mu_1^\intercal\Sigma^{-1}\mu_1 - x^\intercal\Sigma^{-1}x + 2\mu_2^\intercal\Sigma^{-1}x - \mu_2^\intercal\Sigma^{-1}\mu_2\right)\right)\\
&= \exp\left((\mu_1-\mu_2)^\intercal\Sigma^{-1}x + \frac{1}{2}\left(\mu_2^\intercal\Sigma^{-1}\mu_2 - \mu_1^\intercal\Sigma^{-1}\mu_1\right)\right).
\end{aligned}$$

From this, it follows that

$$\begin{aligned}
\mathbb{E}_{x\sim g}\left(\frac{f(x)}{g(x)}\right)^\beta &= \mathbb{E}_{x\sim\mathcal{N}(\mu_2,\Sigma)}\exp\left(\beta\left((\mu_1-\mu_2)^\intercal\Sigma^{-1}x + \frac{1}{2}(\mu_2^\intercal\Sigma^{-1}\mu_2 - \mu_1^\intercal\Sigma^{-1}\mu_1)\right)\right)\\
&= \mathbb{E}_{x\sim\mathcal{N}(\mu_2,\Sigma)}\exp\left(\beta(\mu_1-\mu_2)^\intercal\Sigma^{-1}x\right)\exp\left(\frac{\beta}{2}(\mu_2^\intercal\Sigma^{-1}\mu_2 - \mu_1^\intercal\Sigma^{-1}\mu_1)\right).
\end{aligned}$$

The second factor is a constant and can be taken out of the expectation, we focus on computing the expectation:

$$\mathbb{E}_{x\sim\mathcal{N}(\mu_2,\Sigma)}\exp\left(\beta(\mu_1-\mu_2)^\intercal\Sigma^{-1}x\right).$$

Since $x \sim \mathcal{N}(\mu_2,\Sigma)$, we know that $x = \mu_2 + z$, where $z \sim \mathcal{N}(0,\Sigma)$. Substituting this into the expectation:

$$\mathbb{E}_{z\sim\mathcal{N}(0,\Sigma)}\exp\left(\beta(\mu_1-\mu_2)^\intercal\Sigma^{-1}(\mu_2+z)\right) = \exp\left(\beta(\mu_1-\mu_2)^\intercal\Sigma^{-1}\mu_2\right)\mathbb{E}_{z\sim\mathcal{N}(0,\Sigma)}\exp\left(\beta(\mu_1-\mu_2)^\intercal\Sigma^{-1}z\right).$$

The expectation $\mathbb{E}_{z\sim\mathcal{N}(0,\Sigma)}\exp(a^\intercal z)$ for a normal random variable $z\sim\mathcal{N}(0,\Sigma)$ is given by $\exp\left(\frac{1}{2}a^\intercal\Sigma a\right)$, where $a$ is a vector. Thus, we have

$$\mathbb{E}_{z\sim\mathcal{N}(0,\Sigma)}\exp\left(\beta(\mu_1-\mu_2)^\intercal\Sigma^{-1}z\right)=\exp\left(\frac{\beta^2}{2}(\mu_2-\mu_1)^\intercal\Sigma^{-1}(\mu_2-\mu_1)\right).$$

Combining everything, we get

$$\mathbb{E}_{x\sim\mathcal{N}(\mu_2,\Sigma)}\left(\frac{f(x)}{g(x)}\right)^\beta=\exp\left(\beta(\mu_1-\mu_2)^\intercal\Sigma^{-1}\mu_2+\frac{\beta^2}{2}(\mu_2-\mu_1)^\intercal\Sigma^{-1}(\mu_2-\mu_1)+\frac{\beta}{2}(\mu_2^\intercal\Sigma^{-1}\mu_2-\mu_1^\intercal\Sigma^{-1}\mu_1)\right).$$

The first term and the last term simplify to

$$\begin{aligned}
&\exp\left(\beta(\mu_1-\mu_2)^\intercal\Sigma^{-1}\mu_2+\frac{\beta}{2}(\mu_2^\intercal\Sigma^{-1}\mu_2-\mu_1^\intercal\Sigma^{-1}\mu_1)\right)\\
=&\exp\left(\frac{\beta}{2}\Big[(\mu_1-\mu_2)^\intercal\Sigma^{-1}\mu_2+\mu_1^\intercal\Sigma^{-1}\mu_2-\mu_2^\intercal\Sigma^{-1}\mu_2+\mu_2^\intercal\Sigma^{-1}\mu_2-\mu_1^\intercal\Sigma^{-1}\mu_1\Big]\right)\\
=&\exp\left(\frac{\beta}{2}\Big[(\mu_1-\mu_2)^\intercal\Sigma^{-1}\mu_2+\mu_1^\intercal\Sigma^{-1}(\mu_2-\mu_1)\Big]\right)\\
=&\exp\left(-\frac{\beta}{2}\Big[(\mu_2-\mu_1)^\intercal\Sigma^{-1}(\mu_2-\mu_1)\Big]\right).
\end{aligned}$$

In the end, we obtain

$$\mathbb{E}_{x\sim\mathcal{N}(\mu_2,\Sigma)}\left(\frac{f(x)}{g(x)}\right)^\beta=\frac{\beta(\beta-1)}{2}(\mu_2-\mu_1)^\intercal\Sigma^{-1}(\mu_2-\mu_1).$$

Thus, the Rényi divergence is

$$D_\beta(f\|g)=\frac{\beta}{2}(\mu_2-\mu_1)^\intercal\Sigma^{-1}(\mu_2-\mu_1).$$

□

Lemma 7 shows a way of ensuring RDP for (20). In fact, notice that

$$\mathcal{A}(S)\sim\mathcal{N}\left(f(S),\sigma^2I_k\right).$$

Then, by Lemma 7, it follows that

$$\sup_{S_1\sim S_2}D_\beta\left(f(S_1)\|f(S_2)\right)=\sup_{S_1\sim S_2}\frac{1}{\sigma^2}\frac{\beta}{2}\sup_{S_1\sim S_2}\|f(S_1)-f(S_2)\|_2^2=\frac{1}{\sigma^2}\frac{\beta}{2}\Delta_2^2(f). \tag{21}$$

(22)

The previous expression proves that the Gaussian mechanism defined by (20) is $\left(\beta,\frac{1}{\sigma^2}\frac{\beta}{2}\Delta_2^2(f)\right)$ −RDP.

We conclude this part by reporting three major properties of RDP (Mironov [2017]):

1. **Post-processing**: If $\mathcal{A}$ is $(\beta,\epsilon)$-RDP and $g$ is a randomized mapping, then $g\circ A$ is $(\beta,\epsilon)$-RDP.

2. **Adaptive Composition**: If $\mathcal{A}_1:\mathcal{Z}^n\to\mathcal{X}_1$ is $(\beta,\epsilon_1)$-RDP, and $\mathcal{A}_2:\mathcal{X}_1\times\mathcal{Z}^n\to\mathcal{X}_2$ is $(\beta,\epsilon_2)$-RDP, then

   $$(\mathcal{A}_1(\cdot),\mathcal{A}_2(\mathcal{A}_1(\cdot),\cdot))\text{ is }(\beta,\epsilon_1+\epsilon_2)\text{-RDP.}$$

3. **From RDP to DP**: If $\mathcal{A}$ is $(\beta,\epsilon)$-RDP, then it is $\left(\epsilon+\frac{\log(1/\delta)}{\beta-1},\delta\right)$-DP for any $0<\delta<1$.

### C.2 Proof of Theorems 4 and 5

In this section we present the proof of privacy and convergence for Algorithm 2. The privacy is injected into the algorithm through the randomized smoothing of $\phi$. The overall analysis of Algorithm 2 then follows from an application of the Advanced Composition Theorem 6. The only difference with the privacy analysis of the Frank-Wolfe Algorithm (see Theorem 1) is that we use the closely related definition of Rényi-differential privacy (RDP) (see Mironov [2017]). To motivate this choice, the randomized smoothing we present is based on the introduction of Gaussian noise, which turns out to be easier to analyse in terms of RDP. Finally, the privacy guarantee will be translated in terms of the classical $(\varepsilon, \delta)$ definition of privacy. The accuracy guarantee will be a direct consequence of Theorem 3.4 in d'Aspremont et al. [2022].

More precisely, in order to use the setting presented in d'Aspremont et al. [2022], we need to verify the following properties of the smoothed gradient:

- $\phi_\sigma : \Delta_k \mapsto \mathbb{R}$ is a convex, 1-Lipschitz and $1/\sigma$-smooth function with respect to $\|\cdot\|_1$. Given a Gaussian sampler $\xi \sim \mathcal{N}(0, \sigma^2 I)$, we have access to a stochastic first-order oracle for $\phi_\sigma$, namely

$$G(\mathcal{D}, \xi, S_n, \mathcal{Q}) \in \underset{q \in \mathrm{conv}(\mathcal{Q})}{\mathrm{argmax}} \langle q, \mathcal{P} - \mathcal{D} + \xi \rangle.$$

  Notice that almost surely this is a well-defined and unique assignment. In fact, the maximum of the linear function $q \mapsto \langle q, \mathcal{P} - \mathcal{D} \rangle$ over the polyhedron $\mathrm{conv}(\mathcal{Q})$ takes its maximum in at least one of its finitely many vertices. The maximizer is not unique only if one of the vertices is orthogonal to $\mathcal{P} - \mathcal{D} + \xi$. As $\xi$ has a continuous distribution on $\mathbb{R}^d$, the maximizer is almost surely unique. Further,

$$\mathbb{E}_\xi[G(\mathcal{D}, \xi, S_n, \mathcal{Q})] = \nabla \phi_\sigma(\mathcal{D}) \tag{23}$$

$$\mathbb{E}_\xi \|G(\mathcal{D}, \xi, S_n, \mathcal{Q}) - \nabla \phi_\sigma(\mathcal{D})\|_\infty^2 \leq 2, \tag{24}$$

  where the first equality holds by the Dominated Convergence Theorem, and the inequality holds since the $\ell_\infty$-diameter of $Q$ is at most 2.

- $H : \Delta_k \mapsto \mathbb{R}_+$ is 1-strongly convex with respect to $\|\cdot\|_1$ (see Example 2.5 in Shalev-Shwartz [2012]).

*Proof of Theorem 5.* Proposition 2 guarantees $\phi_\sigma$ to be $1/\sigma$-smooth, and from (24) it follows that the variance of the estimator $\Phi(\mathcal{D}, \xi)$ is bounded. Then, according to Theorem 3.4 in d'Aspremont et al. [2022], the accuracy guarantee for $(P_{\alpha,\sigma})$ achieves the rate

$$\mathbb{E}\left[\mathbf{Gap}_{(P_{\alpha,\sigma})}(\mathcal{P}_\alpha^{\mathrm{priv}})\right] \leq \mathcal{O}\left(\frac{1}{\alpha\sigma T^2} + \frac{1}{\alpha T}\right). \tag{25}$$

**Choice of $\sigma$:** The parameter $\sigma$ corresponds to the variance of the Gaussian noise in the randomized smoothing, and it regulates the privacy budget of each iteration of Algorithm 2. Therefore, the choice of $\sigma$ follows from the overall privacy of the Algorithm.

The privacy enters the objective through $G_\sigma(\mathcal{D}_t^{md}, \xi_t, S_n, \mathcal{Q}) = \mathrm{argmax}_{q \in \mathrm{conv}(\mathcal{Q})} \langle q, \mathcal{P}_n - \mathcal{D} + \xi \rangle$ , which is the composition of a Gaussian mechanism and a deterministic function. More precisely, let $f$ be the $\mathbb{R}^k$-valued function of the dataset $S_n$ defined as $f(S_n) := \mathcal{P}_n - \mathcal{D}$. We define the Gaussian mechanism as $\mathcal{M}(S_n) = f(S_n) + \xi$. The sensitivity of $f$ is then given by

$$\Delta_2(f) = \sup_{S_n \sim S_n'} \|f(S_n) - f(S_n')\|_2 = \sup_{S_n \sim S_n'} \|\mathcal{P}_n - \mathcal{D} - \mathcal{P}_n' + \mathcal{D}\|_2 = \sup_{S_n \sim S_n'} \left\| \frac{1}{n}\sum_{i=1}^n e_{z_i} - \frac{1}{n}\sum_{i=1}^n e_{z_i'} \right\|_2 = \frac{\sqrt{2}}{n}.$$

The last step is justified by the fact that if $S_n$ and $S_n'$ are neighboring, then there exists only one index $i$ for which $z_i \neq z_i'$, and the corresponding $e_{z_i}$ and $e_{z_i'}$ have distance $\sqrt{2}$ with respect to the $\|\cdot\|_2$-norm. Hence, an application of the Gaussian mechanism yields $\mathcal{M}$ to be $\left(\beta, \frac{1}{\sigma^2}\frac{\beta}{n^2}\right)$-RDP (see (21)) . Notice the conclusion remains the same for $G_\sigma$ by the post-processing property of RDP.

For $T$ iterations, the Advanced Composition Theorem for RDP ensures that our algorithm is $(\beta, \frac{\beta T}{n^2\sigma^2})$-RDP. Lastly, by transfering privacy from RDP to DP, the whole Algorithm is $\big(\frac{\beta T}{n^2\sigma^2} + \frac{\log 1/\delta}{\beta - 1}, \delta\big)$-DP. Optimal tuning with respect to $\beta > 1$ (i.e. minimizing the privacy budget $\frac{\beta T}{n^2\sigma^2} + \frac{\log 1/\delta}{\beta-1}$ over $\beta$), leads to

$$\beta^* = 1 + \sqrt{\frac{\log 1/\delta}{T}} n\sigma \, .$$

Finally, for a chosen $\varepsilon > 0$, we obtain $\frac{\beta^* T}{(n\sigma)^2} + \frac{\log 1/\delta}{\beta^* - 1} \leq \varepsilon$ by setting $\sigma = \frac{4\sqrt{T \log 1/\delta}}{n\varepsilon}$ . In fact,

$$\varepsilon \geq \frac{\beta^* T}{(n\sigma)^2} + \frac{\log(1/\delta)}{\beta^* - 1} = 2\sqrt{\frac{T \log(1/\delta)}{n^2\sigma^2}} + \frac{T}{n^2\sigma^2} \, .$$

By multiplying on both sides of the previous expression with $\sigma^2 > 0$, we obtain

$$\varepsilon \geq \frac{\beta^* T}{(n\sigma)^2} + \frac{\log(1/\delta)}{\beta^* - 1} \iff \varepsilon\sigma^2 - 2\sigma\sqrt{\frac{T \log(1/\delta)}{n^2}} - \frac{T}{n^2} \geq 0 \, .$$

The two roots of the corresponding polynomial of second degree in $\sigma$ are

$$\sigma_1 := \frac{1}{\varepsilon n}\left(\sqrt{T\log(1/\delta)} - \sqrt{T\log(1/\delta) + \varepsilon T}\right) \, , \quad \sigma_2 := \frac{1}{\varepsilon n}\left(\sqrt{T\log(1/\delta)} + \sqrt{T\log(1/\delta) + \varepsilon T}\right) \, ,$$

and any $\sigma \in (-\infty, \sigma_1] \cup [\sigma_2, +\infty)$ satisfies the inequality. Finally, note that for $\delta$ sufficiently small, $\log(1/\delta) > \varepsilon$, thus

$$\sigma_2 \leq \frac{(1+\sqrt{2})\sqrt{T\log(1/\delta)}}{\varepsilon n} \leq \frac{4\sqrt{T\log(1/\delta)}}{\varepsilon n} \, .$$

**Choice of $\alpha$:** The choice of $\alpha$ is based on the accuracy gap between $(P_{\alpha,\sigma})$ and $(P)$. In the proof of Theorem 2 we derived the upper bound

$$\mathbf{Gap}_{(P)}(\mathcal{P}_\alpha^{\text{priv}}) = \phi(\mathcal{P}_\alpha^{\text{priv}}) - \phi^* \leq \phi_\alpha(\mathcal{P}_\alpha^{\text{priv}}) - \phi_\alpha^* + \alpha \log k = \mathbf{Gap}_{(P_\alpha)}(\mathcal{P}_\alpha^{\text{priv}}) + \alpha \log k \, .$$

Taking the expectation on both sides yields

$$\mathbb{E}[\mathbf{Gap}_{(P)}(\mathcal{P}_\alpha^{\text{priv}})] \leq \mathbb{E}[\mathbf{Gap}_{(P_\alpha)}(\mathcal{P}_\alpha^{\text{priv}})] + \alpha \log k \, .$$

Further, for any $\mathcal{D} \in \Delta_k$

$$\begin{aligned}
\sigma w(\mathcal{Q}) &:= \mathbb{E}_{\xi\sim\varphi_\sigma}\Big[\max_{q\in Q}\langle q, \xi\rangle\Big] \\
&\geq \mathbb{E}_{\xi\sim\varphi_\sigma}\Big[\max_{q\in Q}\langle q, \mathcal{P} - \mathcal{D} + \xi\rangle\Big] - \max_{q\in\mathcal{Q}}\langle q, \mathcal{P} - \mathcal{D}\rangle \\
&= \phi_\sigma(\mathcal{D}) - \phi(\mathcal{D})
\end{aligned}$$

Moreover, due to symmetry of $\xi$, it holds that

$$\begin{aligned}
\sigma w(\mathcal{Q}) &\geq \max_{q\in\mathcal{Q}}\langle q, \mathcal{P} - \mathcal{D}\rangle - \mathbb{E}_{\xi\sim\varphi_\sigma}\Big[\max_{q\in Q}\langle q, \mathcal{P} - \mathcal{D} - \xi\rangle\Big] \\
&= \phi(\mathcal{D}) - \phi_\sigma(\mathcal{D}).
\end{aligned}$$

Then, for $\mathcal{D}_\alpha^* = \underset{\mathcal{D}\in\Delta_k}{\operatorname{argmin}}\left(\langle q_\alpha^*, \mathcal{P} - \mathcal{D}\rangle + \alpha H(\mathcal{D})\right)$ ,

$$\begin{aligned}
\mathbb{E}[\mathbf{Gap}_{(P_\alpha)}(\mathcal{P}_\alpha^{\text{priv}})] =& \Big[\phi(\mathcal{P}_\alpha^{\text{priv}}) + \alpha H(\mathcal{P}_\alpha^{\text{priv}})\Big] - \Big[\phi(\mathcal{D}_\alpha^*) + \alpha H(\mathcal{D}_\alpha^*)\Big] \\
=& \Big[\phi_\sigma(\mathcal{P}_\alpha^{\text{priv}}) + \alpha H(\mathcal{P}^{\text{priv}})\Big] - \Big[\phi_\sigma(\mathcal{D}_\alpha^*) + \alpha H(\mathcal{D}_\alpha^*)\Big] \\
&+ \Big(\phi(\mathcal{P}_\alpha^{\text{priv}}) - \phi_\sigma(\mathcal{P}_\alpha^{\text{priv}})\Big) + \Big(\phi_\sigma(\mathcal{D}_\alpha^*) - \phi(\mathcal{D}_\alpha^*)\Big) \\
\leq& \mathbb{E}\left[\mathbf{Gap}_{(P_{\alpha,\sigma})}(\mathcal{P}_\alpha^{\text{priv}})\right] + 2\sigma w(\mathcal{Q}) \, .
\end{aligned}$$

Together with (25) this yields

$$\mathbb{E}[\mathbf{Gap}_{(P_\alpha)}(\mathcal{P}_\alpha^{\text{priv}})] \leq \mathbb{E}\left[\mathbf{Gap}_{(P_{\alpha,\sigma})}(\mathcal{P}_\alpha^{\text{priv}})\right] + 2\sigma w(\mathcal{Q}) \ = \mathcal{O}\left(\frac{1}{\alpha\sigma T^2} + \frac{1}{\alpha T}\right) + 2\sigma w(\mathcal{Q}) \ .$$

Note that in the upper bound from (25), the term $O(\frac{1}{\alpha\sigma T^2})$ can be reduced in practise with other techniques (such as restarting algorithm in d'Aspremont et al. [2022]) such that $\mathcal{O}(\frac{1}{\alpha T})$ becomes dominant while to the best of our knowledge, there is no technique to reduce $\mathcal{O}(\frac{1}{\alpha T})$. For simplicity, in our analysis, we will focus on the case where $\mathcal{O}(\frac{1}{\alpha T})$ is dominant compared to the other term i.e. $\frac{1}{\sigma} = \mathcal{O}(T)$. Later, we will show that this regime is justified if $n$ is sufficiently large.

We will show that the optimal number of iterations, denoted as $T^*$ satisfies this inequality and is consistent with the value stated in the theorem, i.e we will have to verify that $\frac{\sqrt{T\log(1/\delta)}}{\varepsilon n} \geq \frac{1}{T}$ (by rearranging the inequality and plugging $\sigma = \frac{4\sqrt{T\log(1/\delta)}}{\varepsilon n}$).

Therefore, we obtain

$$\mathbb{E}[\mathbf{Gap}_{(P_\alpha)}(\mathcal{P}_\alpha^{\text{priv}})] \leq \mathcal{O}\left(\frac{1}{\alpha T}\right) + 2\sigma w(\mathcal{Q}) \ .$$

By combining all the previous inequalities and substituting the value of $\sigma$ obtained from the privacy analysis, we arrive at

$$\mathbb{E}[\mathbf{Gap}_{(P)}(\mathcal{P}_\alpha^{\text{priv}})] \leq \mathbb{E}[\mathbf{Gap}_{(P_\alpha)}(\mathcal{P}_\alpha^{\text{priv}})] + \alpha\log k \leq \mathcal{O}\left(\frac{1}{\alpha T} + \alpha\log k + \frac{\sqrt{T\log(1/\delta)}}{\varepsilon n}\ w(\mathcal{Q})\right) \ .$$

Next, we optimize the right hand side of the upper bound above, obtaining over $\alpha$, yielding to $\alpha^* = \frac{1}{\sqrt{T\log k}}$ and

$$\mathbb{E}[\mathbf{Gap}_{(P)}(\mathcal{P}_\alpha^{\text{priv}})] \leq \mathcal{O}\left(\frac{\sqrt{\log k}}{\sqrt{T}} + \frac{\sqrt{T\log(1/\delta)}}{\varepsilon n}\ w(\mathcal{Q})\right) \ .$$

Finally, a last minimization over $T$ yields

$$T^* = \sqrt{\frac{\log k}{\log(1/\delta)}}\frac{n\varepsilon}{w(\mathcal{Q})} \ ,$$

and

$$\mathcal{O}\left(\frac{w(\mathcal{Q})^2\sqrt{\log(1/\delta)}}{n\epsilon\sqrt{\log k}} + \frac{\sqrt{w(\mathcal{Q})}\log^{1/4}(k)\log^{1/4}(1/\delta)}{n^{1/2}\varepsilon^{1/2}}\right) \ .$$

The second term of the previous expression is dominant when $n \to \infty$ .

Finally, note that the condition $T^* \geq \frac{1}{\sigma}$ is satisfied whenever $n \geq \frac{w^3(\mathcal{Q})\log(1/\delta)}{\epsilon\log^{3/2}(k)}$, which is one of the assumptions of Theorem 5. □

## D Private lower bounds

In this section, we provide evidence for the efficiency of our rates by studying lower bounds for private synthetic data release. We note that previous work has established such lower bounds (see, e.g., Ullman and Vadhan [2011], Bun et al. [2018], Steinke and Ullman [2016]), and, in fact, we will make use of some of these results. The population private lower bound is based on a lower bound previously established for the empirical DP synthetic data problem. We summarize this result below:

**Theorem 7.** *[Lemma 2.11, Theorem 5.16 in Bun et al. [2018]] For all $\gamma > 0$, $\log(k) \geq 6\log(1/\gamma)$, and $k \geq \log(k)/\gamma^2$, there exists a family of queries $\mathcal{Q}$ and a support $\mathcal{Z}$ of size $k$ such that any $(\varepsilon, \delta)$-DP algorithm that takes as input a dataset $S_n$, and outputs $\mathcal{A}(S_n)$ such that $\mathbb{E}_{\mathcal{A}}\Big[\max_{q \in conv(\mathcal{Q})} \langle q, \mathcal{P}_n - \mathcal{A}(S_n)\rangle\Big] \leq \gamma$ , requires at least $n = \tilde{\Omega}\left(\frac{\sqrt{\log(k)}\log|\mathcal{Q}|}{\varepsilon\gamma^2}\right)$ .*

In particular, Theorem 7 implies that for any $(\varepsilon, \delta)$-DP algorithm $\mathcal{A}$ that satisfies the assumptions of Theorem 7 and outputs a synthetic data distribution $\mathcal{A}(S_n)$ with $\mathbb{E}_{\mathcal{A}}\Big[\max_{q \in \mathrm{conv}(\mathcal{Q})} \langle q, \mathcal{P}_n - \mathcal{A}(S_n)\rangle\Big] \leq \gamma$ , it must hold that

$$\gamma \geq \tilde{\Omega}\left(\log^{1/4}(k)\sqrt{\frac{\log|\mathcal{Q}|}{\varepsilon n}}\right).$$

This result establishes a fundamental lower bound on the accuracy $\gamma$ for such algorithms, indicating that $\gamma$ cannot be arbitrarily small without a corresponding increase in the dataset size $n$.

Next we provide a reduction from empirical to population guarantees for DP algorithms. This reduction is based on a known resampling argument used in DP stochastic convex optimization Bassily et al. [2019] and stochastic variational inequalities Boob and Guzmán [2021].

**Lemma 8** (Appendix C in Bassily et al. [2019])**.** *Let $\mathcal{P}$ be a distribution supported on $\mathcal{Z}$, and let $\mathcal{P}_n$ denote the empirical distribution corresponding to a dataset $S_n \in \mathcal{Z}^n$. Suppose there exists an $(\varepsilon/[4\log(1/\delta)], e^{-\varepsilon}\delta/[8\log(1/\delta)])$-differentially private (DP) algorithm $\mathcal{A}$ such that*

$$\mathbb{E}_{\mathcal{A}}\left[\max_{q \in conv(\mathcal{Q})} \langle q, \mathcal{P} - \mathcal{A}(S_n)\rangle\right] \leq \gamma,$$

*where $\gamma > 0$ is a fixed accuracy parameter.*

*Then, there exists an $(\varepsilon, \delta)$-DP algorithm $\mathcal{B}$ such that*

$$\mathbb{E}_{\mathcal{B}}\left[\max_{q \in conv(\mathcal{Q})} \langle q, \mathcal{P}_n - \mathcal{B}(S_n)\rangle\right] \leq \gamma.$$

*Proof.* Given algorithm $\mathcal{A}$ as in the statement and dataset $S_n \in \mathcal{Z}^n$, consider the following algorithm $\mathcal{B}$, which takes $S_n$ as input, and does the following: first, it samples $n$ independent copies from the empirical distribution associated to $S_n$. Calling this new dataset $T_n$, it then runs $\mathcal{A}$ on $T_n$ and outputs $\mathcal{B}(S_n) := \mathcal{A}(T_n)$.

First, $\mathcal{B}$ is $(\varepsilon, \delta)$-DP with respect to $S_n$: this follows from a simple analysis of the probability of repeating examples from $S_n$ in dataset $T_n$, together with the group privacy property (see Appendix C in Bassily et al. [2019]). Next,

$$\begin{aligned}&\mathbb{E}_{\mathcal{B}}\big[\max_{q \in \mathrm{conv}(\mathcal{Q})} \langle q, \mathcal{P}_n - \mathcal{B}(S_n)\rangle\big] \\ =&\mathbb{E}_{\mathcal{A}, T_n \sim (\mathcal{P}_n)^n}\big[\max_{q \in \mathrm{conv}(\mathcal{Q})} \langle q, \mathcal{P}_n - \mathcal{A}(T_n)\rangle\big] \leq \gamma,\end{aligned}$$

where the last inequality holds by the population accuracy of $\mathcal{A}$. □

Combining Theorem 7 and Lemma 8, we obtain the following lower bound for population loss.

**Theorem 8.** *For all $\gamma > 0$, $\log(k) \geq 6\log(1/\gamma)$, and $k \geq \log(k)/\gamma^2$, there exists a family of queries $\mathcal{Q}$ and a support $\mathcal{Z}$ of size $k$ such that any $(\varepsilon, \delta)$-DP algorithm that takes as input a dataset $S_n \sim \mathcal{P}^n$, and outputs $\mathcal{A}(S_n)$ such that*

$$\mathbb{E}_{\mathcal{A}, S_n}\Big[\max_{q \in conv(\mathcal{Q})} \langle q, \mathcal{P} - \mathcal{A}(S_n)\rangle\Big] \leq \gamma\,,$$

*satisfies*

$$\gamma = \tilde{\Omega}\left(\frac{\log^{1/4}(k)\log^{1/2}|\mathcal{Q}|}{\varepsilon^{1/2} n^{1/2}}\right).$$

Let $\mathcal{Q}$ denote the query class derived from Theorem 8, and let $\mathcal{A}(\cdot)$ represent Algorithm 2, which satisfies $(\varepsilon, \delta)$-differential privacy. For a fixed $\delta > 0$ and $\mathcal{P}^{\text{priv}} = \mathcal{A}(S_n)$, the theorem gives

$$\mathbb{E}\left[\max_{q\in\mathcal{Q}} \langle q, \mathcal{P} - \mathcal{P}^{\text{priv}}\rangle\right] = \tilde{\Omega}\left(\frac{\log^{1/4}(k)\log^{1/2}|\mathcal{Q}|}{\varepsilon^{1/2}n^{1/2}}\right).$$

We observe that this lower bound differs by a factor of approximately $w^{1/2}(\mathcal{Q})\log^{1/4}(1/\delta)$ compared to the upper bound established by Algorithm 2 in Theorem 5, which is

$$\mathbb{E}\left[\max_{q\in\mathcal{Q}} \langle q, \mathcal{P} - \mathcal{P}^{\text{priv}}\rangle\right] = \mathcal{O}\left(\frac{w^{1/2}(\mathcal{Q})\log^{1/4}(k)\log^{1/4}(1/\delta)}{\varepsilon^{1/2}n^{1/2}}\right).$$

In particular, since the lower bound does not explicitly depend on $\delta$, our upper bound on the utility guarantee is optimal when $\mathcal{Q} \subset \mathcal{B}_2^k$. As mentioned towards the end of the paper, in this case, $w(\mathcal{Q}) = \mathcal{O}(\log|\mathcal{Q}|)$, which aligns with the upper bound established in Theorem 8 for a fix $\delta > 0$ .