Certified private data release for sparse Lipschitz functions

Konstantin Donhauser[*]
ETH Zurich

Johan Lokna[*]
ETH Zurich

Amartya Sanyal
MPI, Tübingen

March Boedihardjo
ETH Zurich

Robert Hönig
ETH Zurich

Fanny Yang
ETH Zurich

Abstract

As machine learning has become more relevant for everyday applications, a natural requirement is the protection of the privacy of the training data. When the relevant learning questions are unknown in advance, or hyper-parameter tuning plays a central role, one solution is to release a differentially private synthetic data set that leads to similar conclusions as the original training data. In this work, we introduce an algorithm that enjoys fast rates for the utility loss for sparse Lipschitz queries. Furthermore, we show how to obtain a certificate for the utility loss for a large class of algorithms.

1 Introduction

Since sensitive personal information is extensively used in modern data analysis, ensuring the privacy of individual data points has become increasingly critical. Differential privacy (DP) [Dwork et al., 2006] attempts to address this issue and is used by both governmental agencies [Abowd, 2018] and commercial actors [Dwork et al., 2019]. Intuitively, a differential private procedure ensures that its output is not affected significantly by individual data points such that it is not possible to determine whether a particular data point is part of the data set or not. Formally, a probabilistic algorithm $\mathcal{A}$ is said to be ϵ-DP if it satisfies the conditions in Definition 1.

Definition 1. *An algorithm $\mathcal{A}$ is ϵ-DP with $\epsilon > 0$ if for any data sets D, D' differing in a single entry and any measurable subset $S \subset \mathrm{im}(\mathcal{A})$ of the image of $\mathcal{A}$, we have*

$$\mathbb{P}(\mathcal{A}(D) \in S) \leq \exp(\epsilon)\, \mathbb{P}(\mathcal{A}(D') \in S)$$

A long line of research focuses on preserving differential privacy while extracting specific information from data, such as executing specific machine learning algorithms [Bassily et al., 2014, Chaudhuri et al., 2011, Feldman and Xiao, 2014, Raskhodnikova et al., 2008] or answering a number of predetermined queries [Blum et al., 2005, Dagan and Kur, 2022, Dwork and Nissim, 2004, Ghazi et al., 2021, Hardt and Talwar, 2010, Steinke and Ullman, 2016, Abadi et al., 2016]. Despite their good performance, these approaches face the fundamental limitation that no further queries can be answered after releasing the model without affecting privacy guarantees. Moreover, the information leakage introduced by hyper-parameter tuning and model selection must be accounted for in the process to avoid a loss in privacy guarantees (see e.g., Papernot and Steinke [2021]).

An approach that mitigates the above shortcomings is to release a synthetic data set in a differentially private manner that is ideally representative of the original data. This process is also known as "data sanitization" [Dwork et al., 2009] and has the advantage that any operation performed on the released data set does not introduce further privacy leakage - a particularly useful property when it is difficult to predict potential future use cases. Moreover, once a differentially private synthetic data set is generated, any model selection algorithm can be performed on the synthetic data in a non-private way.

On a high level, most data sanitization algorithms rely on some sort of discrepancy measure between data sets, which is then approximately minimized by the returned synthetic data set. A common choice used in SOTA algorithms (see McKenna et al. [2021, 2022], Zhang et al. [2017] and references therein) is to take the Eu-

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

clidean norm between the histograms of the discretized marginals of the data. This choice for the discrepancy measure, however, does not take the geometry of the underlying space into account. For example, if there is a natural ordering in the domain (e.g. naturally occurring continuous covariates such as age, income, etc.), DP data generating algorithms could potentially achieve better performance by taking this inherent structure into account.

One natural discrepancy measure that incorporates such a structure is the Wasserstein distance, studied in recent works [Boedihardjo et al., 2022, He et al., 2023]. However, the authors show that the required number of samples in the original dataset scales exponentially in the dimension of the data (that is, the minimax optimal rate is of order $n^{-1/d}$). An alternative measure is the maximum Wasserstein distance of marginal measures over subsets of variables. Formally, for an algorithm $\mathcal{A}$ that takes an empirical distribution μ_D of a data set D as input and returns a probability measure $\mathcal{A}(D)$[1], we define the following discrepancy measure, also referred to as the *utility loss* in Boedihardjo et al. [2022]

$$\mathcal{U}(\mu_D, \mathcal{A}(D)) := \sup_{S\subset[d];|S|=s} W_1\left(P^S_{\#}\mu_D, P^S_{\#}\mathcal{A}(D)\right), \tag{1}$$

where W_1 is the 1-Wasserstein distance, and $P^S_{\#}\mu_D$ is the marginal measure of μ_D on the s-dimensional canonical subspace defined by the coordinates in S. The utility loss measures the transportation cost between the DP measure $\mathcal{A}(D)$ and the empirical measure of the private data set μ_D. Moreover, by the Kantorovich-Rubinstein Duality Theorem [Villani et al., 2009], the utility loss is equivalent to the *maximum mean discrepancy*

$$\mathcal{U}(\mu_D, \mathcal{A}(D)) = \sup_{f\in\mathcal{F}} \left| \int f(x)d\mu_D - \int f(z)d\mathcal{A}(D) \right|, \tag{2}$$

over the function class $\mathcal{F}$ consisting of all 1-Lispchitz functions (w.r.t. some metric ρ) which are additionally s-sparse; that is, f can be expressed as a function that only depends on s dimensions and that is constant over all other dimensions. The expression in (2) corresponds to what previous works refer to as the accuracy or usefulness when the output is a discrete measure.

In this paper, we are the first to answer the following question

Is it possible to generate a private synthetic dataset with a small structured utility loss in Equation (1) *from a reasonably sized original datasets*?

A positive answer to this question would motivate concrete applied future work on approximate implementations. In Section 4 we show that there exists indeed an algorithm (Sections 2 and 3) that can achieve a rate of order $n^{-1/s}$ for the utility loss, and thus overcomes the curse of dimensionality. This rate is minimax optimal as a function of n, neglecting logarithmic factors. In order to further enhance the practical utility of our framework, we simultaneously address the question

Can we privately release practically meaningful guarantees for the utility loss?

A tight instance-dependent upper bound (which we call *certificate*) will allow the practitioner to account for the maximum utility loss when deriving insights from the synthetic data. While Theorem 1 in Section 4 proves an upper bound for the expected utility loss, it only holds for the optimal, practically infeasible, algorithm presented in Section 3. Further, evaluating the error bound for a given configuration, i.e. for given d, n, s, will likely result in a loose, practically meaningless bound. Instead, we propose an instance-dependent and computable high-probability upper bound that can be privately released alongside the DP measure $\mathcal{A}(D)$. We further show experimentally in Section 5 that this upper bound is tight.

1.1 Notation

We refer to data sets of size n with $D \in T^n$. In the main text of this paper we usually assume that $T = [0,1]^d$ is the hyper cube equipped with the ℓ_∞-metric and let $T_{k,s} := \{1/2k, \cdots, (2k-1)/2k\}^s$ be the centers of a minimal $1/2k$-covering of $[0,1]^s$ of size $N = k^s$. We denote with $\mathcal{M}_{\mathbb{P}}(T)$ the set of probability measures on T and we let $\mathcal{M}(T)$ denote the set of signed measures on T. Moreover, μ_D is the empirical measure of a data set D and $\mathrm{Lap}(\lambda)$ is the Laplace distribution with zero mean and variance $2\lambda^2$. We denote the matrix vector-1-norm with $\|.\|_1$, and use the standard big-Oh notation and the symbols $\lesssim_d, \gtrsim_d, \asymp_d$ to hide universal constants only depending on d. Finally, we denote with $\triangle^{d-1} \subset \mathbb{R}^d$ the probability simplex and for any function $g : T \to T'$ mapping T to T', we denote with $g_{\#} : \mathcal{M}(T) \to \mathcal{M}(T')$ the push-forward operator that outputs a signed measure satisfying $g_{\#}\mu(A) = \mu(g^{-1}(A))$ for any $A \subset T'$.

[1] We note that in the literature, the algorithm's output $\mathcal{A}(D)$ is usually a data set rather than a probability measure. Nevertheless, a data set can always be constructed from a probability measure using standard techniques such as subsampling or discretization (see e.g., [Boedihardjo et al., 2022, He et al., 2023]). We therefore simplify our notation by disregarding this distinction in our paper.

2 Certified DP data generation

In this section, we present a general framework for private data release sketched in Algorithm 1. However, unlike existing approaches, in addition to a DP measure $\mathcal{A}(D)$, Algorithm 1 also returns a certificate $\mathcal{B}_\mathcal{G}$ for the utility loss - a computable upper bound for the utility loss that holds with probability greater equal $1-\delta$ for some $\delta > 0$ and depends on the specific algorithmic choices as well as the particular dataset.

Algorithm 1 Privacy-Preserving Data Generation Framework

Require: Given a query operator $\mathbb{T}$, a noise generating processes $\mathbb{P}_\eta$ and a proxy utility loss $\mathcal{G}$

1: project $v \leftarrow \mathbb{T}\mu_D$
2: construct the ϵ-DP vector $v_{\text{DP}} := v + \eta$ with $\eta \sim \mathbb{P}_\eta$
3: $\mu_{\text{DP}} \leftarrow$ `minimize` $\mathcal{B}_\mathcal{G}(\mu, v_{\text{DP}})$ with respect to $\mu \in \mathcal{M}_\mathbb{P}(T)$
4: **return** DP measure $\mathcal{A}(D) \leftarrow \mu_{\text{DP}}$ and certificate $\mathcal{B}_\mathcal{G}(\mu_{\text{DP}}, v_{\text{DP}})$ for $\mathcal{U}(\mu_D, \mu_{\text{DP}})$

Following the standard abstract pattern of common data release frameworks, Algorithm 1 consists of three steps. First, a linear query operator $\mathbb{T} : \mathcal{M}_\mathbb{P}(T) \to \mathbb{R}^m$ projects the empirical data distribution of the data set D in the domain T, onto a high-dimensional Euclidean space $\mathbb{R}^m$. Moreover, let $\text{im}(\mathbb{T})$ be the image of $\mathbb{T}$ and $\mathbb{T}^{-1} : \text{im}(\mathbb{T}) \to \mathcal{M}_\mathbb{P}(T)$ be any right-inverse, defined as satisfying $\mathbb{T}\mathbb{T}^{-1}\mathbb{T} = \mathbb{T}$. The second step is the standard privatization procedure of adding noise from some distribution $\mathbb{P}_\eta$ to the queries. In the third step, we project back from the query space $\mathbb{R}^m$ to the space of probability measures. In Algorithm 1, this third step is done by minimizing the upper bound from a DP certificate $\mathcal{B}_\mathcal{G} : \mathcal{M}_\mathbb{P}(T) \times \mathbb{R}^m \to \mathbb{R}$ for the utility loss $\mathcal{U}$.

In contrast to existing algorithms, instead of solely releasing the final DP-measure μ_{DP}, we also output a certificate for the chosen sanitization/generation procedure that we detail below.

DP certificate Before defining the DP certificate, we introduce the concept of proxy utility loss

Definition 2. *For a right-inverse $\mathbb{T}^{-1}$, we say that $\mathcal{G} : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is a proxy utility loss on $\mathbb{R}^m$ (*dominating $\mathcal{U}$*) if for all $v, v' \in \text{im}(\mathbb{T}) \subseteq \mathbb{R}^m$,*

$$\mathcal{U}(\mathbb{T}^{-1}v, \mathbb{T}^{-1}v') \leq \mathcal{G}(v, v'). \tag{3}$$

Moreover, $\mathcal{G}$ should be jointly translation invariant, that is $\mathcal{G}(v, v') = \mathcal{U}(v + u, v' + u)$ for any $v, v', u \in \mathbb{R}^m$, and satisfy the triangle inequality.

Next, we propose $\mathcal{B}_\mathcal{G}$ as our DP certificate, a quantity that can be computed for any probability measure $\mu \in \mathcal{M}_\mathbb{P}(T)$, choice of query operator $\mathbb{T}$, and proxy utility loss $\mathcal{G}$. For any $\delta > 0$ and any DP-vector $v_{\text{DP}} := \mathbb{T}\mu_D + \eta$, define

$$\mathcal{B}_\mathcal{G}(\mu, v_{\text{DP}}) := \underbrace{\sup_{\tilde{\mu} \in \mathcal{M}_\mathbb{P}(T)} \mathcal{U}(\tilde{\mu}, \mathbb{T}^{-1}\mathbb{T}\tilde{\mu})}_{\text{discretization error}} + \underbrace{q_{1-\delta}\left(\mathcal{G}(0, \eta)\right)}_{\text{privatization error}} + \underbrace{\mathcal{G}(v_{\text{DP}}, \mathbb{T}\mu) + \mathcal{U}(\mathbb{T}^{-1}\mathbb{T}\mu, \mu)}_{\text{projection error}} \tag{4}$$

where $q_{1-\delta}(Z)$ denotes the $1-\delta$-quantile of the random variable Z and $\mathcal{G}$ is a "proxy" utility loss, that is, any function $\mathcal{G}$ that satisfies Definition 3.

In the following lemma, we show that the certificate $\mathcal{B}_\mathcal{G}(\mu, v_{\text{DP}})$ is a high probability upper bound of the utility loss for any measure $\mu \in \mathcal{M}_\mathbb{P}(T)$.

Lemma 1. *For any proxy utility loss $\mathcal{G}$ from Definition 2 and for $v_{DP} = \mathbb{T}\mu_D + \eta$ with $\eta \sim \mathbb{P}_\eta$, we have that with probability $1-\delta$ over η (for any $\delta > 0$), it holds that uniformly over all probability measures $\mu \in \mathcal{M}_\mathbb{P}(T)$, we have*

$$\mathcal{U}(\mu_D, \mu) \leq \mathcal{B}_\mathcal{G}(\mu, v_{DP}). \tag{5}$$

Proof Using the triangle inequality and Equation (3), we can upper bound the utility loss (2) for any $\mu \in \mathcal{M}_\mathbb{P}(T)$ and μ_D: $\mathcal{U}(\mu_D, \mu) \leq$

$$\begin{aligned}
&\mathcal{U}(\mu_D, \mathbb{T}^{-1}\mathbb{T}\mu_D) + \mathcal{U}(\mathbb{T}^{-1}\mathbb{T}\mu_D, \mathbb{T}^{-1}\mathbb{T}\mu) + \mathcal{U}(\mathbb{T}^{-1}\mathbb{T}\mu, \mu) \\
&\leq \mathcal{U}(\mu_D, \mathbb{T}^{-1}\mathbb{T}\mu_D) + \mathcal{G}(\mathbb{T}\mu_D, \mathbb{T}\mu) + \mathcal{U}(\mathbb{T}^{-1}\mathbb{T}\mu, \mu) \\
&\leq \mathcal{U}(\mu_D, \mathbb{T}^{-1}\mathbb{T}\mu_D) + \mathcal{G}(\mathbb{T}\mu_D, v_{\text{DP}}) \\
&+ \mathcal{G}(v_{\text{DP}}, \mathbb{T}\mu) + \mathcal{U}(\mathbb{T}^{-1}\mathbb{T}\mu, \mu) \\
&= \mathcal{U}(\mu_D, \mathbb{T}^{-1}\mathbb{T}\mu_D) + \mathcal{G}(0, \eta) + \mathcal{G}(v_{\text{DP}}, \mathbb{T}\mu) \\
&+ \mathcal{U}(\mathbb{T}^{-1}\mathbb{T}\mu, \mu)
\end{aligned} \tag{6}$$

where in the last equality we used that $v_{\text{DP}} = \mathbb{T}\mu_D + \eta$ and the joint translation invariance of $\mathcal{G}$.

The first term can be described as the discretization error associated with $\mathbb{T}^{-1}\mathbb{T}$ and can be bounded by the supremum over all probability measures. Since we know the noise distribution $\mathbb{P}_\eta$, the second term can be upper bounded with high probability by its $(1-\delta)$-quantile. Finally, the third term only depends on μ and v_{DP}, which is DP by construction, and the fourth term only depends on μ. As a result, we obtain the desired DP upper bound in Equation (4). □

3 Instantiation of the algorithm for the sparse Wasserstein loss

In this section we present an instantiation of Algorithm 1 for the utility loss in Equation (1). The presented algorithm has an exponential run-time complexity in n (see discussion in Section 4) and we leave

practically useful approximate algorithms as a future work. For simplicity of exposition, throughout this section, we only consider the case where the underlying space $T = [0,1]^d$ is the hypercube equipped with the ℓ_∞-metric and refer to reader to Appendix C for results on general metric spaces. We first define the specific choice of the query operator $\mathbb{T}$, the noise generating processes $\mathbb{P}_\eta$, and the proxy utility loss $\mathcal{G}$. Finally, we summarize Algorithm 1 with these choices.

Query operator $\mathbb{T}$ Similar to previous works (see e.g., [McKenna et al., 2022, 2021, Zhang et al., 2017]), we choose $\mathbb{T}\mu_D$ to be the vector representing all discretized s-marginals. More precisely, the output of the query operator $\mathbb{T} : \mathcal{M}_\mathbb{P}(T) \to \mathbb{R}^{\binom{d}{s}k^s}$ consists of all $\binom{d}{s}$ blocks $\mathbb{T}\mu_D =: v = [v^{S_1}, \cdots, v^{S_K}]^T$ with $K = \binom{d}{s}$ of size $v^{S_j} \in \mathbb{R}^{k^s}$ and $k \in \mathbb{N}_+$ is some discretization parameter. Moreover, $S_j \subset [d]$ and $|S_j| = s$ are all subsets of size s.

Every block is constructed by first projecting the measure μ on its marginals $\mu^{S_j} = P^{S_j}_\# \mu$, which we then discretize by a finite measure on $T_{k,s}$ forming a $1/2k$-covering of $P^{S_j}T$ of size k^s. Finally, we choose any (arbitrary) right-inverse $\mathbb{T}^{-1} : \mathrm{im}(\mathbb{T}) \to \mathcal{M}_\mathbb{P}(T_{k,d}) \subset \mathcal{M}_\mathbb{P}(T)$ such that it returns a finite measure on $T_{k,d}$.

Noise vector η As in [Boedihardjo et al., 2022, He et al., 2023], we use the (matrix transformed) Laplace mechanism [Dwork et al., 2006, Xiao et al., 2010b], which generates a DP "copy" v_{DP} (Step 2 in Algorithm 1) of the vector $v := \mathbb{T}\mu_D$ (Step 1 in Algorithm 1) by adding matrix transformed i.i.d Laplace noise[2]

$$v_{\mathrm{DP}} := v + \eta := v + [\Phi\tilde{\eta}]_{1:m}\,, \tag{7}$$

where $\tilde{\eta}$ is an i.i.d. Laplace random vector with variance as in Lemma 2 and $\Phi \in \mathbb{R}^{m_\Phi \times m_\Phi}$ is some invertible matrix of dimension $m_\Phi \geq m$. A standard quantity when comparing two datasets is the sensitivity of $\mathbb{T}$

$$\Delta_\mathbb{T} := \sup_{D,D'} \|\mathbb{T}\mu_D - \mathbb{T}\mu_{D'}\|_1, \tag{8}$$

where we take the supremum over all datasets $D, D' \subset T^n$ of size n which differ in at most one point. Using the previous definitions, the following privacy guarantee holds:

Lemma 2. *(Corollary of Theorem 3.6 in [Dwork and Roth, 2014]) The vector v_{DP} is ϵ-DP for $\eta = [\Phi\tilde{\eta}]_{1:m}$ and*

$$\tilde{\eta} \sim \left(\mathrm{Lap}\left(\frac{\|\Phi^{-1}\|_1 \Delta_\mathbb{T}}{\epsilon}\right)\right)^{m_\Phi}. \tag{9}$$

[2]Since all entries of $\mathbb{T}\mu_D$ are multiples of $\frac{1}{n}$, an alternative choice would be to use the discrete Laplace mechanism [He et al., 2023, Inusah and Kozubowski, 2006], which yields the same theoretic guarantees in Section 4 when straight forwardly modifying the proofs in Appendix C.

By multiplying the noise in Equation (7) with the matrix Φ, we obtain a "correlated" noise. While such a noise has been previously used in the literature (see e.g., [Xiao et al., 2010a]), a key insight in Boedihardjo et al. [2022] is to use a Haar-matrix transformed Laplacian noise to obtain tight guarantees for the utility loss of 1-Lipschitz continuous functions (i.e., when $s = d$). Based on this idea, we now describe our choice of Φ used in Step 2 of Algorithm 1. Recall that we denote with $v^{S_j} \in \mathbb{R}^{k^s}$ the j-th block of the vector v. For every $j \in \left[\binom{d}{s}\right]$ we define $v^{S_j}_{\mathrm{DP}} := v^{S_j} + \left[\Phi^{S_j}\tilde{\eta}^{S_j}\right]_{1:k^s}$ where Φ^{S_j} are the scaled versions of the transposed Haar-matrix from Lemma 3 in Appendix B such that $\|(\Phi^{S_j})^{-1}\|_1 = 1$. Furthermore, $\tilde{\eta}^{S_j}$ are i.i.d. Laplacian random vectors with variance as in Lemma 2 and $\Delta_\mathbb{T} = \binom{d}{s}\frac{2}{n}$.

Proxy utility loss $\mathcal{G}$ We now describe a choice for a proxy utility loss $\mathcal{G} = \mathcal{L}_\mathbb{T}$ such that the certificate $\mathcal{B}_{\mathcal{L}_\mathbb{T}}$ can be effectively minimized using linear programming (see Section 5.3 for a computationally efficient approximation). Inspired by [Boedihardjo et al., 2022] which studies the case where $s = d$, we choose

$$\mathcal{L}_\mathbb{T}(v,u) := \max_{S\subset[d],|S|=s} \frac{1}{k}\sum_{l=1}^{k^s}\left|\sum_{i=1}^{l} v_i^S - u_i^S\right|. \tag{10}$$

where v_i^S is the i-th element of the block $v^S \in \mathbb{R}^{k^s}$. We assume that the elements of v_{DP} (and thus also $\mathbb{T}\mu_D$) are ordered as follows: note that any block vector $v^{S_j}_{\mathrm{DP}} \in \mathbb{R}^{k^s}$ of v_{DP} has a one-to-one mapping to a discrete signed measure $\omega \in \mathcal{M}_\mathbb{P}(T_{k,s})$ on the s-dimensional discrete hyper cube $T_{k,s}$, defined by $\omega^{S_j}_{v_{\mathrm{DP}}} := \sum_{z_i \in T_{k,s}} (v^{S_j}_{\mathrm{DP}})_i \delta[z_i]$, where δ is the Dirac-delta point measure. Using this definition, we order the indices of v_i^S such that the corresponding centers z_i form a Hamiltonian path, or more formally, such that for all $i \leq k^s - 1$, $\|z_i - z_{i+1}\|_\infty = 1/2k$.

We refer to Appendix C.1 and C.2 for a proof that $\mathcal{L}_\mathbb{T}$ indeed satisfies Definition 2. On a high level, the Hamiltonian path allows us to reduce the problem of constructing a proxy utility loss function over vectors v^S representing discrete measures in a s-dimensional space to one of construction a proxy utility loss function over discrete measures on an interval of $\mathbb{R}$.

Minimization of $\mathcal{B}_\mathcal{G}(\mu_{\mathbf{DP}}, v_{\mathbf{DP}})$ in Step 3 We finally describe how to minimize $\mathcal{B}_\mathcal{G}(\mu_{\mathrm{DP}}, v_{\mathrm{DP}})$ in Step 3 of Algorithm 1. Note that as the query operator $\mathbb{T}$ discretizes every marginal measure using a $1/2k$-covering, the maximum discretization error in Equation (4) equals

$$\sup_{\tilde{\mu}\in\mathcal{M}_\mathbb{P}(T)} \mathcal{U}\left(\tilde{\mu}, \mathbb{T}^{-1}\mathbb{T}\tilde{\mu}\right) = 1/2k. \tag{11}$$

Moreover, the term $\mathcal{U}\left(\mathbb{T}^{-1}\mathbb{T}\mu, \mu\right)$ is zero whenever $\mu \in \mathcal{M}_{\mathbb{P}}(T_{k,d}) = \mathbb{T}^{-1}\mathbb{T}\mathcal{M}(T)$. Thus, minimizing the upper bound from the certificate $\mathcal{B}_{\mathcal{L}_{\mathbb{T}}}$ in Step 3 in Algorithm 1 simplifies to

$$\arg \min_{\mu \in \mathcal{M}_{\mathbb{P}}(T_{k,d})} \mathcal{L}_{\mathbb{T}}(v_{\mathrm{DP}}, \mathbb{T}\mu). \tag{12}$$

Finally, we can output the certificate in Step 4 in Algorithm 1 after computing the $1-\delta$ quantile of $\mathcal{L}_{\mathbb{T}}(0, \eta)$, which can be efficiently approximated using Monte Carlo samples.

4 Statistical rates for the utility loss

In this section, we present rates for the expected utility loss of the instantiation of Algorithm 1 as described in Section 3. To the best of our knowledge, we are the first to study the utility loss, presented in Equation (1). Theorem 1 shows that the rate of the utility loss only depends on s in the exponent but not on d. Thus, we see that by only considering s-sparse marginals in Equation (1) we can effectively overcome the curse of dimensionality.

Theorem 1. *Let $\mathcal{F}$ be the set of s-sparse 1-Lipschitz functions on $T = [0,1]^d$ with respect to the ℓ_∞-metric. Then, for any $n\epsilon \geq d$ and any $s \leq d$, Algorithm 1 is ϵ-DP and for $k \asymp \left(\binom{d}{s}\frac{\log(\epsilon n)^2}{n\epsilon}\right)^{-1/s}$ has an expected utility loss (2) at most*

$$\mathbb{E}\,\mathcal{U}(\mu_D, \mathcal{A}(D)) \lesssim_s \left(\binom{d}{s}\frac{\log(\epsilon n)^2}{n\epsilon}\right)^{1/s}. \tag{13}$$

We refer to Appendix C.3 for the proof of Theorem 1 which is a consequence of the general statement, presented in Theorem 2 that applies to general metric spaces. We note that when $s = d$, we obtain exactly the rate in [Boedihardjo et al., 2022] up to a logarithmic factor.

Optimality in n The rate in Theorem 1 is optimal in n up to logarithmic factors. Indeed, as a corollary of the results in Section 8 in [Boedihardjo et al., 2022] we obtain the following information theoretic lower bound on the expected utility loss (for constant ϵ)

$$\inf_{\mathcal{A}(D):\epsilon\text{-DP}} \sup_{D \in T^n} \mathbb{E}\,\mathcal{U}(\mu_D, \mathcal{A}(D)) \gtrsim \left(\frac{\lfloor d/s \rfloor}{n\epsilon}\right)^{1/s} \tag{14}$$

which has the same exponent $1/s$ as the term in the upper bound in Theorem 1. We present a proof sketch for the lower bound in Appendix E.

Open problem: tightness in d While this lower bound matches the exponential decay rate in n of the upper bound in Theorem 1, the dependency on d is not the same. Nonetheless, tightening this gap poses a challenging problem and we believe it will require novel creative ideas. The main difficulty arises from the interdependence of the $\binom{d}{s}$ marginal measures, which makes it challenging to enhance either of the two bounds without carefully considering this dependency. We motivate future work to solve the open problem of finding the right dependency on d in the lower bound in Equation 14 supported by a matching upper bound.

Proof sketch The proof of Theorem 1 builds on the ideas developed in [Boedihardjo et al., 2022] for the case where $s = d$. While the result is a relatively straight forward extension, the main technical contributions are two-folds: we simplify the proofs in the mentioned paper and present them in the context of Section 2 (see Appendix C.1), which then allows us to extend the results in Boedihardjo et al. [2022] to the case where $s < d$ (see Section C.2 and C.3).

The first part of the proof is devoted to showing that $\mathcal{L}_{\mathbb{T}}$ indeed satisfies Definition 2. Using Equation (6) (with $\mathcal{G} = \mathcal{L}_{\mathbb{T}}$), we then upper bound the utility loss in expectation by

$$\begin{aligned}\mathbb{E}\left[\mathcal{U}(\mu_D, \mathcal{A}(D))\right] &\leq \mathcal{U}(\mu_D, \mathbb{T}^{-1}\mathbb{T}\mu_D) + \mathbb{E}\left[\mathcal{L}_{\mathbb{T}}(\mathbb{T}\mu_D, v_{\mathrm{DP}})\right] \\ &+ \mathbb{E}\left[\mathcal{L}_{\mathbb{T}}(v_{\mathrm{DP}}, \mathbb{T}\mathcal{A}(D))\right] + \mathcal{U}(\mathbb{T}^{-1}\mathbb{T}\mathcal{A}(D), \mathcal{A}(D)) \\ &\leq 1/2k + 2\,\mathbb{E}\left[\mathcal{L}_{\mathbb{T}}(\mathbb{T}\mu_D, v_{\mathrm{DP}})\right] \end{aligned} \tag{15}$$

where we used the fact that $\mathcal{A}(D)$ is a solution of Equation (12) and thus $\mathcal{U}(\mathbb{T}^{-1}\mathbb{T}\mathcal{A}(D), \mathcal{A}(D)) = 0$ and $\mathcal{L}_{\mathbb{T}}(v_{\mathrm{DP}}, \mathbb{T}\mathcal{A}(D)) \leq \mathcal{L}_{\mathbb{T}}(\mathbb{T}\mu_D, v_{\mathrm{DP}})$, and that the discretization error is upper bounded by $1/2k$. Thus, we obtain an upper bound for the expected utility loss by bounding the term $\mathbb{E}\left[\mathcal{L}_{\mathbb{T}}(\mathbb{T}\mu_D, v_{\mathrm{DP}})\right] = \mathbb{E}\left[\mathcal{L}_{\mathbb{T}}(0, [\Phi\tilde{\eta}]_{1:m})\right]$, which only depends on the random vector $\tilde{\eta}$ and Φ, but not on the measure μ_D. In this step, we crucially rely on the choice of Φ in Section 3. Finally, we obtain the bound in Theorem 1 by optimizing over the discretization parameter k.

4.1 Further discussion

Run-time complexity We now discuss the run-time complexity of Algorithm 1. First note that both Step 1 and 2 in Algorithm 1 have a run-time complexity of $O(d^s n + m) = O(d^s n + d^s k^s)$ (and thus polynomial in d). However, solving the minimization problem in Step 3 in Algorithm 1 requires running a linear program over the $|T_k| = k^d$ free variables, which has a run-time complexity of $O(\mathrm{poly}(m + k^d)) = O(\mathrm{poly}(d^s k^s + k^d))$.

In particular, when pluging-in the optimal choice for the discretization parameter k from Theorem 1, we obtain a run-time complexity of order $O(d^s n + \mathrm{poly}(n^{d/s}))$. For small constant choices of s we therefore obtain an

exponential run-time complexity in d, and computational hardness results [Dwork et al., 2009, Ullman and Vadhan, 2011] for the special case of estimating 2-way marginals in fact suggest that the exponential dependency in d cannot be avoided. Nevertheless, we can still hope for practically meaningful approximate algorithms with fast (polynomial) run-time complexity (see Section 5) and motivate future work on this topic.

Other types of sparsity In Theorem 2 we obtain fast rates without a dependency on d in the exponent by restricting $\mathcal{F}$ to s-sparse Lipschitz functions. As we show in Appendix D, we can also obtain similar results when $\mathcal{F}$ is the set of all 1-Lipschitz functions but the data itself lives on a (unknown) s-dimensional space. Importantly, the algorithm can adapt to the degree of the sparsity and does not need to have access to the effective dimension s of the data. This opens up the pathway for adaptive DP data generating algorithms, which we leave as future work.

Comparison with [Boedihardjo et al., 2022, He et al., 2023] Previous works considered the special case where the utility loss is the Wasserstein distance, i.e. the loss from Equation (1) for $s = d$. In this case, the authors show that the optimal rate for the utility loss is of order $n^{-1/d}$ (see Equation (18) in Section 6). Theorem 1 shows that by restricting to s-way marginals, we can address the curse of dimensionality in the rates for the utility loss, resulting in only a linear dependency in d instead of an exponential.

Comparison with approaches minimizing the Euclidean distance A natural question to ask is how the rates in Theorem 1 compare with the ones of algorithms minmizing the squared Euclidean distance in Step 3 in Algorithm 1, as commonly done in previous works (see McKenna et al. [2021, 2022], Zhang et al. [2017] and references therein). More formally, we draw η in Step 2 from an i.i.d. Laplacian (resp. Gaussian) distribution and in Step 3 construct a dataset by minimizing the average Euclidean distance of the vectorized representations of its marginals to v_{DP}. From a straight forward computation, such an algorithm only yields an expected utility loss of order $\tilde{O}_{d,s}((\epsilon n)^{-1/(s+1)})$, hiding logarithmic dependencies. When comparing with the rates in Theorem 1, we can see that there is a gap in the exponent of $1/(s+1)$ vs. $1/s$. We believe that this rate is tight. We present a more detailed discussion in Appendix F.

5 A tighter certificate and numerical evaluation using public data

In this section, we present numerical simulations illustrating the utility loss $\mathcal{U}$ and the certificate $\mathcal{B}_{\mathcal{L}_{\mathbb{T}}}$ from Section 3 for Algorithm 1. To avoid the exponential run-time complexity, we first introduce a computationally efficient approximation of Algorithm 1 using public data in Section 5.1. We then present the numerical simulations on real-world data sets in Section 5.3. In Section 5.4, we discuss a tighter choice for the proxy utility loss $\mathcal{U}_{\mathbb{T}}$ which, in turn, yields a tighter certificate. The numerical analysis presented in this paper serves as a proof of concept and motivates future research on efficient approximate algorithms.

5.1 Computationally efficient approximation of Step 3 in Algorithm 1

To avoid the exponential run-time complexity of Step 3 in Algorithm 1, we restrict the search space for μ_{DP} to a set $\widetilde{\mathcal{M}}_{\mathbb{P}}(T_k)$ of discrete measures that are supported on a given public data set $D_{\mathrm{pub}} = \{z_i\}_{i=1}^{n_{\mathrm{a}}} \subset T_k$. Then, we approximate Step 3 in Algorithm 1 (using Equation (12)) as $\mu_{\mathrm{approx}} \in \arg\min_{\mu \in \widetilde{\mathcal{M}}_{\mathbb{P}}(T_k)} \mathcal{L}_{\mathbb{T}}(v_{\mathrm{DP}}, \mathbb{T}\mu)$

$$\text{with} \quad \widetilde{\mathcal{M}}_{\mathbb{P}}(T_k) = \left\{ \sum_{i=1}^{n_{\mathrm{a}}} \alpha_i \delta[z_i] \mid \alpha \in \Delta^{n_{\mathrm{a}}-1} \right\}. \quad (16)$$

The approach of using a public data set had been previously proposed in the literature [Boedihardjo et al., 2021, Liu et al., 2021a] to improve the computational efficiency. In fact, the optimization problem in Equation (16) is still a linear program and can be solved with run-time complexity $O(\mathrm{poly}(d^s n_{\mathrm{a}} + m)) = O(\mathrm{poly}(d^s)\ \mathrm{poly}(n_{\mathrm{a}} + k^s))$. Thus, we obtain a polynomial dependency on d given that n_{a} grows at most polynomially in d.

5.2 Experimental setting

For all experiments we use $\epsilon = 1$. We use the real-world data sets *ACSIncome* and *ACSTravelTime* [Ding et al., 2021] collected from the "American Community Survey" and rescale them such that the data lives in the hypercube. For the private data, we use $n = 195665$ samples from California from 2018 (ACSIncome) and $n = 91200$ samples from New York from 2018 (ACSTravelTime). As public data sets, we randomly choose $n_{\mathrm{a}} = 4000$ samples from Alabama from 2018, California from 2014 (ACSIncome), Massachusetts from 2018, and New York from 2014 (ACSTravelTime). Furthermore, from the ACSIncome data set we only consider $d = 5$ features ("AGEP, SCHL, OCCP, POBP, WKHP") and from

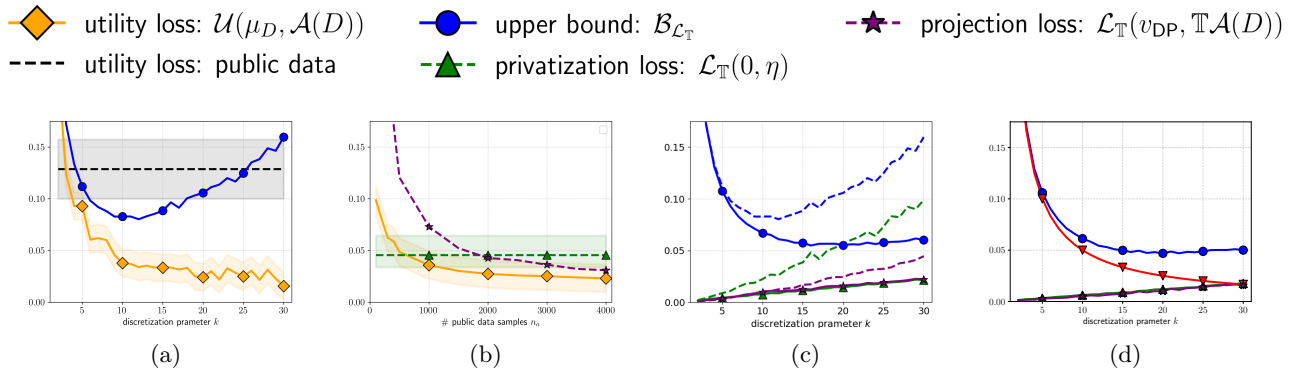


Figure 1: a) upper and lower bound of the utility loss (see Appendix A) and the utility loss of the empirical measure of the "raw" public data as a function of function of the discretization parameter k (dashed horizontal line). Moreover, the upper bound from Equation (4) with $\delta = 0.1$ for Algorithm 1 (blue line). b) we compare the term $\mathcal{L}_\mathbb{T}(v_{\text{DP}}, \mathbb{T}\mathcal{A}(D))$ (dashed purple line) with the term $\mathcal{L}_\mathbb{T}(v_{\text{DP}}, \mathbb{T}\mu_D) = \mathcal{L}_\mathbb{T}(0, \eta)$, (dashed green line) as a function of the size of the public data set. We plot the mean, 0.05 and 0.95 quantiles of $\mathcal{L}_\mathbb{T}(\eta, 0)$ (horizontal lines). Moreover, we plot the upper and lower bounds for the utility loss (yellow line) (see Appendix A). We choose the discretization parameter $k = 25$ for the query operator $\mathbb{T}$. (c) the upper bound (solid lines) in Equation (4) when using $\mathcal{G} = \mathcal{U}_\mathbb{T}$ instead of $\mathcal{L}_\mathbb{T}$ (dashed line) for the upper bound in Step 4 in Algorithm 1. (d) illustration of the upper bound from Equation (4) and its individual terms for $\mathcal{G} = \mathcal{U}_\mathbb{T}$ with $\delta = 0.1$ for Algorithm 1 as a function of the discretization parameter k.

the ACSTravelTime data set, we consider $d = 4$ features ("AGEP, SCHL, PUMA, POVPIP") (see [Ding et al., 2021] for the documentation). If not further specified, we choose the ACSIncome data set with the samples from Alabama from 2018 as public data and the samples form California from 2018 as private data set.

Moreover, while exactly computing the utility loss (2) turns out to be computationally infeasible for our choices of n and n_a, we can compute sharp upper and lower bounds as described in Appendix A. For all plots, we take the average over 10-independent runs and use 200 samples to approximating the quantiles in Equation (5).

5.3 Numerical evaluation using real world data

In this section we present numerical experiments for Algorithm 1. We illustrate in Figure 2 the certificate from Equation (4) and the utility loss for the measure generated by Algorithm 1 as a function of the discretization parameter k. As the experiments show, we achieve significantly smaller utility loss than when simply using the "un-optimized" public data. Moreover, the certificate from Algorithm 1 captures the trend of the utility loss for small k and yields a non-trivial guarantee. We refer to Section 5.4 for a tighter choice for the certificate.

Moreover, we argue based on Figure 1b that we can measure the "sub-optimality" of an approximate solution μ_{approx} for Step 3 in Algorithm 1 by comparing the terms $\mathcal{L}_\mathbb{T}(v_{\text{DP}}, \mathbb{T}\mu_{\text{approx}})$ and $\mathcal{L}_\mathbb{T}(v_{\text{DP}}, \mathbb{T}\mu_D) = \mathcal{L}_\mathbb{T}(0, \eta)$ (see Figure 1b). Intuitively, these two terms capture the "distances" between the DP measure μ_{approx} and the private measure μ_D to the "reference point" v_{DP} respectively. Thus, once $\mathcal{L}_\mathbb{T}(v_{\text{DP}}, \mathbb{T}\mu_{\text{approx}}) \approx \mathcal{L}_\mathbb{T}(v_{\text{DP}}, \mathbb{T}\mu_D)$, μ_D and μ_{approx} have the same "distance" to v_{DP}, we can no longer expect an improvement in the utility loss when further minimizing $\mathcal{L}_\mathbb{T}(v_{\text{DP}}, \mathbb{T}\mu_{\text{approx}})$. We illustrate this in Figure 1b, where we plot the utility loss and the two terms as a function of the amount of public data samples. By increasing the amount of public data samples, we can improve our approximation of Step 3 in Algorithm 1. As the results show, once $\mathcal{L}_\mathbb{T}(v_{\text{DP}}, \mathbb{T}\mu_{\text{approx}}) \approx \mathcal{L}_\mathbb{T}(v_{\text{DP}}, \mathbb{T}\mu_D)$ the utility stagnates, meaning that we do not benefit from further optimizing $\mathcal{L}_\mathbb{T}(v_{\text{DP}}, \mathbb{T}\mu_{\text{approx}})$ by increasing the amount of public samples.

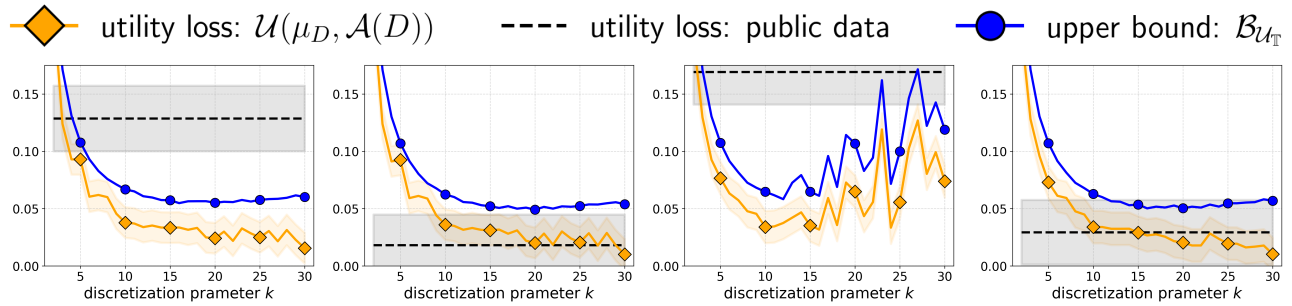
5.4 A tighter certificate

In this section, we present a tighter choice for the certificate in Step 4 in Algorithm 1.

A tighter choice for the proxy utility loss $\mathcal{G} = \mathcal{U}_\mathbb{T}$ We first present an alternative choice for the proxy utility loss $\mathcal{U}_\mathbb{T}$ that yields a sharper certificate in Step 4 of Algorithm 1. A natural idea to construct a tighter proxy utility loss is to simply "extend" the definition of the utility loss $\mathcal{U}$ to signed measures on the marginals. We do this as follows: for any two vectors $u, v \in \mathbb{R}^m$, we define $\mathcal{U}_\mathbb{T}(v, u) :=$

$$\max_{\substack{S\subset[d]\\|S|=s}} \sup_{\substack{f\in\mathcal{F}(T_{k,s})\\f(0)=0}} \left| \sum_{z_i \in T_{k,s}} f(z_i) \left(\omega_v^S(\{z_i\}) - \omega_u^S(\{z_i\})\right) \right|, \tag{17}$$

where $\mathcal{F}(T_{k,s})$ is the set of all 1-Lipschitz continuous functions over $T_{k,s}$ w.r.t. the ℓ_∞-metric and ω_v^S are as



(a) Income AL 18' (b) Income CA 14' (c) TravelTime MA 18' (d) TravelTime NY 14'

Figure 2: same curves as Figure 1a but for the certificate $\mathcal{G} = \mathcal{U}_\mathbb{T}$ for different combinations of public and private data sets.

in Section 3.

Clearly, $\mathcal{U}_\mathbb{T}$ coincides with $\mathcal{U}$ on $\text{im}(\mathbb{T})$ (i.e., satisfies Equation (3)), is jointly translation invariant, and satisfies the triangular inequality. Hence, $\mathcal{U}_\mathbb{T}$ from Equation (17) is a valid choice for the upper bound in Equation (4). Concerning the computational complexity, we note that the RHS in Equation (17) can be computed by taking the maximum over $\binom{d}{s}$-solutions of linear programs, which has a total run-time complexity $O\left(\binom{d}{s}\text{poly}(k^s)\right)$. While this allows for a tractable computation of the certificate $\mathcal{B}_{\mathcal{U}_\mathbb{T}}$ by taking the maximum over linear programs, *optimizing* over the set of measures $\mathcal{M}_\mathbb{P}(T)$ is not tractable [3]. Thus, we keep $\mathcal{G} = \mathcal{L}_\mathbb{T}$ in Step 3 in Algorithm 1 but release the certificate in Step 4 of Algorithm 1 using $\mathcal{G} = \mathcal{U}_\mathbb{T}$.

Numerical evaluation In Figure 1c, we compare the certificates and the individual terms in Equation (5) for the choices $\mathcal{G} = \mathcal{U}_\mathbb{T}$ and $\mathcal{G} = \mathcal{L}_\mathbb{T}$. Our results clearly show that the choice $\mathcal{B}_{\mathcal{U}_\mathbb{T}}$ yields a significantly tighter certificate than $\mathcal{B}_{\mathcal{L}_\mathbb{T}}$, especially for large values of k. Figure 1d plots the individual terms in the certificate from Equation (4) for the choice $\mathcal{G} = \mathcal{U}_\mathbb{T}$. The results show that increasing k leads to a higher privacy and projection error, but lowers the dicretization error (see Equation (4)). This highlights the trade-off between more fine-grained discretizations and the need to add sufficient noise in order to preserve privacy.

Finally, Figure 2 illustrates the certificate from Equation (4) and the utility loss for the measure generated by Algorithm 1 as a function of the discretization parameter k for different choices of the public and private data set. Our experiments show that when the public data has a large distribution shift, we achieve a significantly smaller utility loss than when simply using the "un-optimized" public data. Moreover, in all plots, the certificate closely matches the utility loss and correctly captures the trend.

[3]The only exception here is the case where $s = d$, in which case we would obtain the algorithm in He et al. [2023].

6 Related work

Releasing datasets privately while minimizing the utility loss for a specific function class is a major challenge in differential privacy. Most studies have focused on preserving utility for counting queries (referred to as statistical queries in Kearns [1998]). The field started with Blum et al. [2008], who applied the Exponential Mechanism algorithm of McSherry and Talwar [2007] to release a private data set while ensuring that the loss of utility for any a priori known set of counting queries grows at most logarithmically with the set size. Subsequent works by Dwork et al. [2009], Hardt et al. [2012], Hardt and Rothblum [2010], Roth and Roughgarden [2010] improved both the statistical and computational complexity and studied fundamental statistical-computational tradeoffs and gaps [Dwork et al., 2009, Ullman and Vadhan, 2011]

In order to reduce sample complexity, several works have considered sparsity assumptions in various ways. For example, Blum and Roth [2013] propose an efficient algorithm for queries that take on a non-zero value only on a small subset of an unstructured discrete domain. More related to us, a special class of linear statistical or counting queries are k-way marginals [Dwork et al., 2015, Liu et al., 2021b, Thaler et al., 2012]. A k-way marginal query involves fixing the values of k indices and determining the proportion of data that matches those values. Further, a range of works [Barak et al., 2007, Cheraghchi et al., 2012, Gupta et al., 2011] also study the query class of k-way conjunctions and provide fast algorithms when k is small. Further common problems in the privacy literature related to this paper include histogram release [Abowd et al., 2019, Acs et al., 2012, Hay et al., 2009, Meng et al., 2017, Nelson and Reuben, 2019, Qardaji et al., 2013, Xiao et al., 2010b, Xu et al., 2013, Zhang et al., 2016] and private clustering [Balcan et al., 2017, Ghazi et al., 2020, Stemmer, 2020, Su et al., 2016].

Finally, recent works [Boedihardjo et al., 2022, He et al., 2023, Wang et al., 2016] studied the case where $\mathcal{F}$ is the class of all 1-Lipschitz continuous functions, resulting in

the utility loss (2) equaling the 1-Wasserstein distance. A small Wasserstein distance is desirable in many practical applications as it for instance guarantees that clusters present in the original data remain preserved (see the discussion in [Boedihardjo et al., 2022]). However, prior results [Boedihardjo et al., 2022, He et al., 2023] also suggest that ensuring a small Wasserstein distance requires exponentially many samples in the dimension. For example, for the d-dimensional hypercube $[0,1]^d$ (with $d \geq 2$) equipped with the ℓ_∞-metric, the papers [Boedihardjo et al., 2022, He et al., 2023] together show that the optimal expected utility loss is of order

$$\mathbb{E}\, \mathcal{U}(\mu_D, \mathcal{A}(D)) \asymp \left(\frac{1}{n\epsilon}\right)^{1/d}, \tag{18}$$

where n is the size of the data set D and μ_D its corresponding empirical measure.

7 Conclusion and future work

Especially in sensitive domains, we desire a certificate for the maximum utility loss to ensure that the data is provably minimally affected by the DP mechanism. We take a step in this direction by introducing Algorithm 1 in Section 2, which simultaneously releases a DP discrete probability measure and provides a certificate for the maximum utility loss when $\mathcal{F}$ is the class of all s-sparse Lipschitz continuous functions. As shown in Section 4, our algorithm achieves an optimal non-asymptotic exponential decay rate for the expected utility loss and effectively overcomes the curse of dimensionality for moderate choices of s.

Future work The certificate in Algorithm 1 can be computed for any "approximate" solution and we leave practically meaningful, efficient approximations of Step 3 in Algorithm 1 as future work. Moreover, we motivate theoretical research on the right dependency on d in Theorem 1. Improving the upper bound would likely result in a novel algorithm with potentially practical applications, while an improved lower bound would require the development of new mathematical ideas and provide evidence for the optimality of Algorithm 1.

Acknowledgements

KD was supported by the ETH AI Center and the ETH Foundations of Data Science. AS was supported by the ETH AI Center.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

John Abowd, Robert Ashmead, Garfinkel Simson, Daniel Kifer, Philip Leclerc, Ashwin Machanavajjhala, and William Sexton. Census topdown: Differentially private data, incremental schemas, and consistency with public knowledge. *US Census Bureau*, 2019.

John M Abowd. The us census bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2867–2867, 2018.

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

Maria-Florina Balcan, Travis Dick, Yingyu Liang, Wenlong Mou, and Hongyang Zhang. Differentially private clustering in high-dimensional euclidean spaces. In *International Conference on Machine Learning*, pages 322–331, 2017.

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

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *IEEE symposium on foundations of computer science*, pages 464–473, 2014.

Avrim Blum and Aaron Roth. Fast private data release algorithms for sparse queries. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques: International Workshop*, pages 395–410, 2013.

Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138, 2005.

Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database privacy. In *Proceedings of the ACM Symposium on Theory of Computing*, 2008.

March Boedihardjo, Thomas Strohmer, and Roman Vershynin. Privacy of synthetic data: A statistical framework. *arXiv preprint arXiv:2109.01748*, 2021.

March Boedihardjo, Thomas Strohmer, and Roman Vershynin. Private measures, random walks, and

synthetic data. *arXiv preprint arXiv:2204.09167*, 2022.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12 (3), 2011.

Mahdi Cheraghchi, Adam Klivans, Pravesh Kothari, and Homin K Lee. Submodular functions are noise stable. In *Proceedings of the ACM-SIAM symposium on Discrete Algorithms*, pages 1586–1592, 2012.

Yuval Dagan and Gil Kur. A bounded-noise mechanism for differential privacy. In *Conference on Learning Theory*, pages 625–661, 2022.

Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Cynthia Dwork and Kobbi Nissim. Privacy-preserving datamining on vertically partitioned databases. In *Advances in Cryptology–CRYPTO 2004: International Cryptology Conference*, pages 528–544, 2004.

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 Theory of Cryptography*, 2006.

Cynthia Dwork, Moni Naor, Omer Reingold, Guy N Rothblum, and Salil Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the ACM symposium on Theory of computing*, pages 381–390, 2009.

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

Cynthia Dwork, Nitin Kohli, and Deirdre Mulligan. Differential privacy in practice: Expose your epsilons! *Journal of Privacy and Confidentiality*, 9(2), Oct. 2019.

Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. In *Proceedings of the Conference on Learning Theory*, pages 1000–1019, 2014.

Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. Differentially private clustering: Tight approximation ratios. *Advances in Neural Information Processing Systems*, 33:4040–4054, 2020.

Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. On avoiding the union bound when answering multiple differentially private queries. In *Proceedings of Conference on Learning Theory*, volume 134, pages 2133–2146, 15–19 Aug 2021.

Jin Sheng Guf and Wei Sun Jiang. The Haar wavelets operational matrix of integration. *International Journal of Systems Science*, 27(7):623–628, 1996.

Anupam Gupta, Moritz Hardt, Aaron Roth, and Jonathan Ullman. Privately releasing conjunctions and the statistical query barrier. In *Proceedings of the ACM symposium on Theory of computing*, pages 803–812, 2011.

Moritz Hardt and Guy N Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *IEEE symposium on foundations of computer science*, pages 61–70, 2010.

Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 705–714, 2010.

Moritz Hardt, Katrina Ligett, and Frank Mcsherry. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially-private histograms through consistency. *arXiv preprint arXiv:0904.0942*, 2009.

Yiyun He, Roman Vershynin, and Yizhe Zhu. Algorithmically effective differentially private synthetic data, 2023.

Seidu Inusah and Tomasz Kozubowski. A discrete analogue of the laplace distribution. *Journal of Statistical Planning and Inference*, 136:1090–1102, 03 2006. doi: 10.1016/j.jspi.2004.08.014.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.

Terrance Liu, Giuseppe Vietri, Thomas Steinke, Jonathan Ullman, and Steven Wu. Leveraging public data for practical private query release. In *Proceedings of the International Conference on Machine Learning*, pages 6968–6977, 2021a.

Terrance Liu, Giuseppe Vietri, and Steven Z Wu. Iterative methods for private synthetic data: Unifying framework and new methods. *Advances in Neural Information Processing Systems*, 34:690–702, 2021b.

Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the NIST contest: A scalable and general approach to differentially private synthetic

data. *CoRR*, abs/2108.04978, 2021. URL `https://arxiv.org/abs/2108.04978`.

Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. AIM: an adaptive and iterative mechanism for differentially private synthetic data. *CoRR*, abs/2201.12677, 2022. URL `https://arxiv.org/abs/2201.12677`.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *IEEE Symposium on Foundations of Computer Science*, pages 94–103, 2007.

Xue Meng, Hui Li, and Jiangtao Cui. Different strategies for differentially private histogram publication. *Journal of Communications and Information Networks*, 2(3):68–77, 2017.

Boel Nelson and Jenni Reuben. Sok: Chasing accuracy and privacy, and catching both in differentially private histogram publication. *arXiv preprint arXiv:1910.14028*, 2019.

Nicolas Papernot and Thomas Steinke. Hyperparameter tuning with renyi differential privacy. *arXiv preprint arXiv:2110.03620*, 2021.

Wahbeh Qardaji, Weining Yang, and Ninghui Li. Understanding hierarchical methods for differentially private histograms. *Proceedings of the VLDB Endowment*, 6(14):1954–1965, 2013.

Sofya Raskhodnikova, Adam Smith, Homin K Lee, Kobbi Nissim, and Shiva Prasad Kasiviswanathan. What can we learn privately. In *Proceedings of the Symposium on Foundations of Computer Science*, pages 531–540, 2008.

Aaron Roth and Tim Roughgarden. The median mechanism: Interactive and efficient privacy with multiple queries. In *Proc. STOC*, 2010.

Thomas Steinke and Jonathan Ullman. Between pure and approximate differential privacy. *Journal of Privacy and Confidentiality*, 7(2):3–22, 2016.

Uri Stemmer. Locally private k-means clustering. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, page 548–559, 2020.

Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, and Hongxia Jin. Differentially private k-means clustering. In *Proceedings of the sixth ACM conference on data and application security and privacy*, pages 26–37, 2016.

Justin Thaler, Jonathan Ullman, and Salil Vadhan. Faster algorithms for privately releasing marginals. In *Automata, Languages, and Programming: 39th International Colloquium*, pages 810–821, 2012.

Jonathan Ullman and Salil Vadhan. Pcps and the hardness of generating synthetic data. In *Proceedings of Theory of Cryptography*, volume 5978, pages 572–587, 2011.

S. S. Vallender. Calculation of the wasserstein distance between probability distributions on the line. *Theory of Probability & Its Applications*, 18(4):784–786, 1974. doi: 10.1137/1118101.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

Ziteng Wang, Chi Jin, Kai Fan, Jiaqi Zhang, Junliang Huang, Yiqiao Zhong, and Liwei Wang. Differentially private data releasing for smooth queries. *The Journal of Machine Learning Research*, 17(1):1779–1820, 2016.

Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. Differential privacy via wavelet transforms, 2010a.

Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. Differential privacy via wavelet transforms. *IEEE Transactions on knowledge and data engineering*, 23 (8):1200–1214, 2010b.

Jia Xu, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, Ge Yu, and Marianne Winslett. Differentially private histogram publication. *The VLDB journal*, 22:797–822, 2013.

Jun Zhang, Xiaokui Xiao, and Xing Xie. Privtree: A differentially private algorithm for hierarchical decompositions. In *Proceedings of the international conference on management of data*, pages 155–170, 2016.

Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4): 1–41, 2017.

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). No, our experiments are only very limited.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Yes
   (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). No, our experiments are only very limited.

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

A Experimental Setting

For completeness, we describe the upper and lower bounds for the utility loss (1) used in Figures 1 and 2. Although we can compute the Wasserstein distance for discrete measures using linear programming, for the choices of n and n_a considered in this paper, a direct computation turns out to be computationally infeasible. Instead, for any measures μ, μ', we can bound the wasserstein distance using the triangle inequality

$$\mathcal{U}(\mu,\mu') \leq \mathcal{U}(P_{\#}^{T_{k,d}}\mu, P_{\#}^{T_{k,d}}\mu') + \mathcal{U}(\mu, P_{\#}^{T_{k,d}}\mu) + \mathcal{U}(P_{\#}^{T_{k,d}}\mu', \mu') \quad \text{and} \tag{19}$$

$$\mathcal{U}(\mu,\mu') \geq \mathcal{U}(P_{\#}^{T_{k,d}}\mu, P_{\#}^{T_{k,d}}\mu') - \mathcal{U}(\mu, P_{\#}^{T_{k,d}}\mu) - \mathcal{U}(P_{\#}^{T_{k,d}}\mu', \mu'), \tag{20}$$

where $P^{T_{k,d}} : \mathcal{M}_{\mathbb{P}}(T) \to \mathcal{M}_{\mathbb{P}}(T_{k,d})$ projects the space T to $T_{k,d}$. We remark that $\mathcal{U}(P_{\#}^{T_{k,d}}\mu, P_{\#}^{T_{k,d}}\mu') = \mathcal{U}(\mathbb{T}^{-1}\mathbb{T}\mu, \mathbb{T}^{-1}\mathbb{T}\mu') = \mathcal{U}_{\mathbb{T}}(T\mu, \mathbb{T}\mu')$, which can be computed efficiently (for moderate choices of k) as described in Section 3. Moreover, by construction $\mathcal{U}(P_{\#}^{T_{k,d}}\mathcal{A}(D), \mathcal{A}(D)) = 0$, and for any data set D, we can bound

$$\mathcal{U}(P_{\#}^{T_{k,d}}\mu_D, \mu_D) \leq \max_{S\subset[d];|S|=s} \frac{1}{n}\sum_{i=1}^{|D|} \|P^S z_i^S - P_{\#}^{T_{k,d}} P^S z_i\|_\infty, \tag{21}$$

Finally, we plot in Figures 1 and 2 the upper and lower bounds from Equation (19) and 20, as well as the term $\mathcal{U}(P_{\#}^{T_{k,d}}\mu_D, P_{\#}^{T_{k,d}}\mathcal{A}(D))$ and $\mathcal{U}(P_{\#}^{T_{k,d}}\mu_D, P_{\#}^{T_{k,d}}\mu_{D_{\text{pub}}})$, respectively, for $k = 30$.

B Haar Basis

In this section, we give a quick introduction to the Haar matrices, which play a crucial role in the proofs of Theorem 2. For a more in depth discussion please refer to [Guf and Jiang, 1996].

The k-th transposed Haar matrix M_k is a $2^k \times 2^k$ matrix. We can separate the columns into $k+1$ levels $L_0, \ldots, L_k$ where level L_l contains $\max\{1, 2^{l-1}\}$ columns (see Figure 3). The level L_0 only consists of the first column, the level L_1 contains the next column, and the level L_2 the following two columns and so on. Moreover, the absolute values of all non-zero elements in the level L_l in M_k are all equal to $\max\{1, 2^{l-1}\}/2^k$. Furthermore, a key property of transposed Haar matrices is that the columns are sparse; each column in the level L_l in M_k contains exactly $2^{\min\{k, k-l+1\}}$ non-zero elements. We visualize in Figure 3 the transposed Haar matrices M_1, M_2 and M_3. The pattern can be extended to general M_k. It is straight forward to verify that all the columns in any M_k are orthogonal. Consider any two columns in the same level; their support is disjoint and their scalar product vanishes. On the other hand, for any two columns in different levels, their support is either disjoint or the support of one is contained in an index set where the values in the other column is constant. Hence, as every column has an equal number of positive and negative values with equal magnitude, we can conclude that either way the scalar product is zero. Consequently, if we scale the columns appropriately, the Haar basis matrix would be orthogonal. M_k^{-1} is therefore equal to M_k^T with the columns scaled appropriately. Finally, we note that the appropriate scaling is such that all non-zero elements of the inverse have absolute value 1, as visualized in Figure 4.

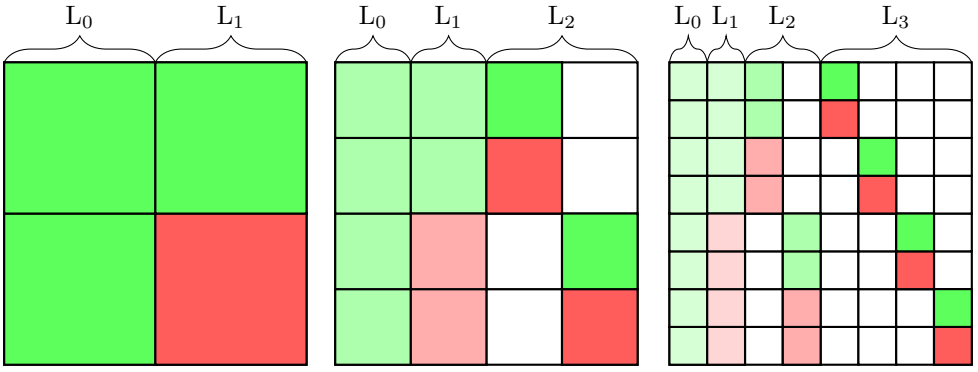


Figure 3: Evolution of the transposed Haar basis shwoing M_1, M_2 and M_3. Green cells contain positive values, red cells negative values while white cells contain 0. The intensity of a cell correspond to the magnitude of the value within it.

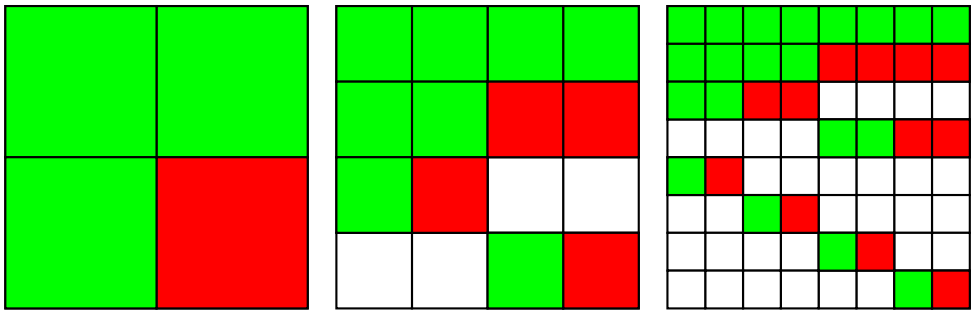


Figure 4: Evolution of the inverse of the transposed Haar basis showing M_1^{-1}, M_2^{-1} and M_3^{-1}. Green cells contain value 1, red cells -1 while white cells contain 0.

We can now state the following lemma which we use in the proof of Theorem 2. These properties of the transposed Haar matrix have already been implicitly used in the proofs in [Boedihardjo et al., 2022].

Lemma 3. *For any $m \geq 0$ and $\Phi = (\lceil \log_2(m) \rceil + 1) M_{\lceil \log_2(m) \rceil}$ with M_k the k-th transposed Haar Matrix, it holds that $\|\Phi^{-1}\|_1 \leq 1$ and*

$$\max_{i \in [m]} \left\| \sum_{j=1}^{i} \Phi_j \right\|_1 \leq (\lceil \log_2(m) \rceil + 1)^2 \quad \text{and} \quad \max_{i \in [m]} \left\| \sum_{j=1}^{i} \Phi_j \right\|_2 \leq (\lceil \log_2(m) \rceil + 1)^{3/2} \tag{22}$$

Proof of Lemma 3

- For the first property, we note that by definition, $\|\Phi^{-1}\|_1 = \max_l \|\Phi_l^{-1}\|_1$ where Φ_l^{-1} is the l-th column of Φ^{-1}. By the construction (see Figure 4), we have that for all k, $\|(M_k^{-1})_l\|_1 = k+1$, and thus $\|M_k^{-1}\|_1 = k+1$. Therefore, we get that $\|\Phi^{-1}\|_1 = \|M_k^{-1}\|_1/(k+1) = 1$.

- For the second and third property we notice that due to the disjoint support between the columns within a level, the contiguous support of each column and the equal number of positive and negative values within a column (see Figure 3), when summing the j-first columns of M_k there will be at most one non-zero element in each level. As the 1-norm of each column is 1, we know that the magnitude of this non-zero element is upper bounded by 1. Furthermore, in total we have $k+1$ levels. Hence, $\| \sum_{l=1}^{i} (M_k)_l \|_2 \leq \sqrt{k+1}$ for any $i \in [k+1]$. Thus, for any $i \in [2^k]$ we get that $\| \sum_{l=1}^{i} \Phi_l \|_2 \leq (k+1)^{\frac{3}{2}}$ and $\| \sum_{l=1}^{i} \Phi_l \|_1 \leq (k+1)^2$.

C Extension of Theorem 1 to general metric spaces

In this section we generalize the setting in Theorem 1 to general metric spaces in Theorem 2. Generally, we can ask for an algorithm $\mathcal{A}$, taking a data set $D \in T^n$ on some measurable space $(T, \mathcal{B})$ as input, to achieve a small utility loss (2) over a function class

$$\mathcal{F} = \bigcup_{i=1}^{K} \mathcal{F}^{(i)} := \bigcup_{i=1}^{K} \{ f^{(i)} \circ h^{(i)} | f^{(i)} \text{ is 1-Lipschitz w.r.t. } \rho^{(i)} \}, \tag{23}$$

with $(T^{(i)}, \rho^{(i)})_{i=1}^{K}$ being some set of metric spaces and surjective measurable functions $h^{(i)} : T \to T_{(i)}$ serving as "projections" of T to $T^{(i)}$. We assume that the push-forward σ-algebras generated by $h^{(i)}$ coincide with the σ-algebras generated by $\rho^{(i)}$ on $T^{(i)}$.

Analogous to Equation (1), by the Kantorovich-Rubinstein Duality Theorem [Villani et al., 2009], the utility loss is exactly the maximum of the Wasserstein distances w.r.t. the metrics $\rho_{(i)}$,

$$\mathcal{U}(\mu, \mu') = \max_{i \in [K]} W_1(h^{(i)}_{\#}\mu, h^{(i)}_{\#}\mu'). \tag{24}$$

To give an example, in the context of Theorem 1, we have $K = \binom{d}{s}$ and $T^{(i)} = [0,1]^s$ are the s-sparse marginals of T with $\rho^{(i)}(x,y) = \|x-y\|_\infty$. Furthermore, $h^{(i)}$ are the functions projecting $x \in [0,1]^d$ to the corresponding s-dimensional subspaces and $h^{(i)}_{\#}\mu$ are the marginal measures of μ on the corresponding s-dimensional subspaces.

We now turn to the main result of this section, Theorem 2, which provides a general upper bound for the utility loss over the function class $\mathcal{F}$. When $K = 1$ we can directly obtain the upper bounds in Theorem 2 from the results in [Boedihardjo et al., 2022]. The main technical contribution of this section is to show that the price to pay if $K > 1$ is at most linear in K (plus a logarithmic factor). Let $N(T, \rho, t)$ be the covering number of T, we have:

Theorem 2. *For the setting described above, there exists a universal constant $c > 0$ and a randomized algorithm $\mathcal{A}$ that takes a data set $D \in T^n$ of size n as input and returns a finitely-supported measure $\mathcal{A}(D) \in \mathcal{M}_{\mathbb{P}}(T)$ such that $\mathcal{A}$ is ϵ-DP private and has expected utility loss* (2) *over the function class $\mathcal{F}$, defined in Equation* (23)*, at most*

$$\mathbb{E}\,\mathcal{U}(\mu_D, \mathcal{A}(D)) \leq t + cK \max_{i\in[K]} \left[\frac{\left(\lceil \log_2 N\big(T^{(i)}, \rho^{(i)}, t\big)\rceil + 1 + \log(K)\right)^2}{n\epsilon} \int_{t/2}^{\operatorname{diam}\left(T^{(i)}\right)/2} N\Big(T^{(i)}, \rho^{(i)}, x\Big)\, dx \right]. \tag{25}$$

We now divide the proof of Theorem 2 into two parts. First, we prove in Appendix C.1 Theorem 2 for the known case when $K = 1$. While the proofs builds upon the ideas in Boedihardjo et al. [2022], we present the proof in a different structure based on Equation (15) and Definition 2 from Section 4. This structure is crucial since it then allows us in a second part in Section C.2 to extend the proofs to the case where $K > 2$

C.1 Proof of Theorem 2 when $K = 1$

We first present a proof for the case where $K = 1$, where we can assume w.l.o.g. that h_1 is the identity function. In this case, we obtain exactly the results from Section 7 in [Boedihardjo et al., 2022]. The proof consists of three parts, where we first construct a query operator $\mathbb{T}$ and a proxy utility loss $\mathcal{L}_{\mathbb{T}}$ (Definition 2).

Construction of the query operator $\mathbb{T}$ and the right-inverse $\mathbb{T}^{-1}$: Let $m = N(T, \rho, t)$ be the covering number of T and let T_m be the centers of any minimal t-covering of T. Furthermore, let $\mathbb{D}_{T_m} : \mathcal{M}_{\mathbb{P}}(T) \to \mathcal{M}_{\mathbb{P}}(T_m) \subset \mathcal{M}_{\mathbb{P}}(T)$ be the projection operator which constructs a probability measure on T_m by dividing the space T into m disjoint measurable neighborhoods around the points in T_m of diameter at most t. Given any indexing of the elements in T_m, we can straightforwardly define a bijection from $\mathcal{M}_{\mathbb{P}}(T_m)$ to the probability simplex $\mathcal{V} = \{z \in \mathbb{R}^m : \|z\|_1 = 1, z_i \geq 0\}$ and thus complete the construction of the operator $\mathbb{T}$. Furthermore, let the right-inverse $\mathbb{T}^{-1} : \mathcal{V} \to \mathcal{M}_{\mathbb{P}}(T_m) \subset \mathcal{M}_{\mathbb{P}}(T)$ be any operator such that $\mathbb{T}^{-1}\mathbb{T} = \mathbb{D}_{T_m}$.

To simplify the following analysis, we now describe how to choose a particular indexing of the elements in T_m based on the analysis in Boedihardjo et al. [2022]. Proposition 6.5 and Equation (7.4) in [Boedihardjo et al., 2022] together guarantee the existence of a finite set $\Omega = \{w_1, \cdots, w_m\} \subset [0, L]$ as well as a 1-Lipschitz (w.r.t. ρ) bijection $f : \Omega \to T_m$ with

$$L = 64 \int_{t/2}^{\operatorname{diam}(T)/2} N(T, \rho, x) dx. \tag{26}$$

Note that w.l.o.g. we can assume that $w_i \leq w_{i+1}$ which thereby implicitly induces a Hamiltonian path on T_m of length at most L. We can now define the indexing of the elements in T_m by $z_i = f(w_i)$, which allows us to complete the construction of the operator $\mathbb{T}$.

Step 3 in Algorithm 1: proxy utility loss $\mathcal{L}_{\mathbb{T}}$: One of the key ideas of Boedihardjo et al. [2022] is to reduce the the problem of constructing a private measure on T to that of constructing a private measure on an interval on $\mathbb{R}$. This is done via the bijection f introduced in the previous paragraph. Along these lines, we now show how we can make use of the bijection f to construct a proxy utility loss $\mathcal{L}_{\mathbb{T}}$ satisfying the conditions in Definition 2.

If $\mu_{T_m} \in \mathcal{M}_{\mathbb{P}}(T_m)$, then let $\mu_{T_m,f}$ be the push-forward measure of $\mu_{T_m} \in \mathcal{M}_{\mathbb{P}}(T_m)$, i.e. the measure on $\mathcal{M}_{\mathbb{P}}(\Omega)$ such that for any $A \subset \Omega$, $\mu_{T_m,f}(A) = \mu_{T_m}(f(A))$. Since f is a 1-Lipschitz continuous function, the Wasserstein

distance of any two measures $\mu_{T_m}, \mu'_{T_m} \in \mathcal{M}_{\mathbb{P}}(T_m)$ is upper bounded by

$$\begin{aligned}\mathcal{U}(\mu_{T_m}, \mu'_{T_m}) &= W_1(\mu_{T_m}, \mu'_{T_m}) \leq W_1(\mu_{T_m,f}, \mu'_{T_m,f}) \\ &= \|F_{\mu_{T_m,f}} - F_{\mu'_{T_m,f}}\|_{L^1(\mathbb{R})} \\ &= \sum_{j=1}^{m}(w_{j+1} - w_j)\left|\sum_{i=1}^{j}(\mu_{T_m}(f^{-1}(w_i)) - \mu'_{T_m}(f^{-1}(w_i)))\right| \\ &= \sum_{j=1}^{m}(w_{j+1} - w_j)\left|\sum_{i=1}^{j}(v_i - v'_i)\right|\end{aligned} \tag{27}$$

where the second equality follows from the identity in [Vallender, 1974], $F_{\mu_{T_m,f}}$ is the cumulative distribution function of the measure $\mu_{T_m,f}$, and we use the notation $v = \mathbb{T}\mu_{T_m}$, $v' = \mathbb{T}\mu'_{T_m}$, and $w_{m+1} = L$. Using the RHS of Equation (27) we can now define the utility proxy utility loss function $\mathcal{L}_{\mathbb{T}}$ on $\mathbb{R}^m$

$$\mathcal{L}_{\mathbb{T}}(v, v') := \sum_{j=1}^{m}(w_{j+1} - w_j)\left|\sum_{i=1}^{j}(v_i - v'_i)\right|, \tag{28}$$

and therefore have $\mathcal{U}(\mu_{T_m}, \mu'_{T_m}) \leq \mathcal{L}_{\mathbb{T}}(\mathbb{T}\mu_{T_m}, \mathbb{T}\mu'_{T_m})$ for all $\mu_{T_m}, \mu'_{T_m} \in \mathcal{M}_{\mathbb{P}}(T_m)$. Hence, we conclude that $\mathcal{L}_{\mathbb{T}}$ satisfies the conditions in Definition 2.

Upper bound for the utility loss $\mathcal{U}$: Finally, we can prove the result by upper bounding the utility loss using Equation (15). First, the "projection" error term in Equation (15) can be upper bounded by

$$\sup_{\mu\in\mathcal{M}_{\mathbb{P}}(T)} \mathcal{U}(\mu, \mathbb{T}^{-1}\mathbb{T}\mu) = \sup_{\mu\in\mathcal{M}_{\mathbb{P}}(T)} \sup_{S\subset[d];|S|=s} W_1(P^S_{\#}\mu, P^S_{\#}\mathbb{T}^{-1}\mathbb{T}\mu) = \sup_{\mu\in\mathcal{M}_{\mathbb{P}}(T)} \sup_{S\subset[d];|S|=s} W_1(P^S_{\#}\mu, P^S_{\#}\mathbb{D}_{T_m}\mu) \leq t, \tag{29}$$

which is a consequence of the fact that $\mathbb{D}_{T_m}$ moves every point mass at most distance t.

Next, we bound second term in Equation (15), 2 $\mathcal{L}_{\mathbb{T}}(\mathbb{T}\mu_D, v_{\mathrm{DP}})$. To do so, we first need to choose the matrix Φ in Equation (7). As first suggested in [Xiao et al., 2010b] and also in [Boedihardjo et al., 2022], we can choose $\Phi = (\lceil \log_2(m) \rceil + 1)M_{\lceil \log_2(m) \rceil}$ where M_k is the k-th Haar Matrix (see Appendix B) and then apply the Laplace mechanism as in Lemma 2. First note that $\Delta_{\mathbb{T}} \leq \frac{2}{n}$ where $\Delta_{\mathbb{T}}$ is defined in Lemma 2. Indeed, the vectors $\mathbb{T}\mu_D$ and $\mathbb{T}\mu_{D'}$ are representations for the discretized measures $\mathbb{D}_{T_m}\mu_D$ and $\mathbb{D}_{T_m}\mu_{D'}$, which differ at most in two points. By Lemma 2 we therefore need to draw $\tilde{\eta} \sim \left(\mathrm{Lap}\left(\frac{2\|\Phi^{-1}\|_1}{n\epsilon}\right)\right)^{m_\Phi}$. A straightforward calculation then yields the following upper bound:

$$\begin{aligned}&\mathbb{E}\ \mathcal{L}_{\mathbb{T}}(\mathbb{T}\mu_D, v_{\mathrm{DP}}) = \mathbb{E}\ \mathcal{L}_{\mathbb{T}}(\mathbb{T}\mu_D, \mathbb{T}\mu_D + [\Phi\tilde{\eta}]_{1:m}) \\ &= \sum_{j=1}^{m}(w_{j+1} - w_j)\mathbb{E}_{\tilde{\eta}\sim\left(\mathrm{Lap}\left(\frac{2\|\Phi^{-1}\|_1}{n\epsilon}\right)\right)^{m_\Phi}}\left|\sum_{i=1}^{j}(\Phi\tilde{\eta})_i\right| \\ &\overset{\text{(Jensen)}}{\leq} L\max_{j=1}^{m}\mathbb{E}_{\tilde{\eta}\sim\left(\mathrm{Lap}\left(\frac{2\|\Phi^{-1}\|_1}{n\epsilon}\right)\right)^{m_\Phi}}\left[\mathrm{std}\left(\sum_{i=1}^{j}(\Phi\tilde{\eta})_i\right)\right] = \frac{2\sqrt{2}L\|\Phi^{-1}\|_1}{n\epsilon}\cdot\max_{j=1}^{m}\left\|\sum_{i=1}^{j}\Phi_i\right\|_2,\end{aligned} \tag{30}$$

where Φ_i is the i-th row of Φ. We then obtain the desired upper bounds when applying Lemma 3 in Appendix B.

C.2 Full proof of Theorem 2 for arbitrary $K > 1$

We now show how we can extend the result for the case where $K = 1$ (in Section C.2) to the case of $K > 1$ by changing the query operator $\mathbb{T}$. We construct $\mathbb{T}$ with $m = \sum_i m_{(i)}$ with $m_{(i)} = N\big(T^{(i)}, \rho^{(i)}, t\big)$, by simply stacking the operators $\mathbb{T}_{(i)} \circ h^{(i)}_{\#} : \mathcal{M}_{\mathbb{P}}(T) \to \mathbb{R}^{m_{(i)}}$ where $\mathbb{T}_{(i)}$ are the query operators described in Section C.1 for the metric spaces $(T_{(i)}, \rho_{(i)})$. It is then straightforward to verify that for any right inverse, the "projection" error $\mathcal{U}(\mu_D, \mathbb{T}^{-1}\mathbb{T}\mu_D)$ from Equation (15) is upper bounded by t.

In Step 2 of Algorithm 1, we apply the transformed Laplace mechanism to every block $v_{(i)} \in \mathbb{R}^{m_{(i)}}$ as described in Section C.1. However, this requires increasing the sensitivity to $\Delta_{\mathbb{T}} \leq \frac{2K}{n}$ due to the increased total number of

measurements. Since the proxy utility loss function $\mathcal{U}$ is simply the maximum over the Wasserstein distances over the projected measures of every subspace $T_{(i)}$ (see Equation (24)), we can define the proxy utility loss function $\mathcal{L}_{\mathbb{T}}$ dominating the utility loss $\mathcal{U}$ (see Definition 2) to be the maximum loss $\mathcal{L}_{\mathbb{T}}(v, v') = \max_{i\in[K]} \mathcal{L}_{\mathbb{T}_{(i)}}(v_{(i)}, v'_{(i)})$ where $v_{(i)} \in \mathbb{R}^{m_{(i)}}$ is the i-th block of the vector $v \in \mathbb{R}^m$ and $\mathcal{L}_{\mathbb{T}_{(i)}}$ is the corresponding proxy utility loss function as constructed in Section C.1. We then obtain the desired result when bounding the term $\mathbb{E}\mathcal{L}_{\mathbb{T}}(\mathbb{T}\mu_D, v_{\mathrm{DP}})$ from Equation (15) using Lemma 4.

Lemma 4. *Assume that v_{DP} is generated as in described in the proof of Theorem 2. We can upper bound $\mathcal{L}_{\mathbb{T}}(\mathbb{T}\mu, v_{DP})$ from Equation* (15) *by:*

$$\mathbb{E}\ \mathcal{L}_{\mathbb{T}}(\mathbb{T}\mu_D, v_{DP}) \lesssim \max_{i\in[K]} \frac{2K(\lceil \log_2(m_{(i)})\rceil + 1 + \log(K))^2}{n\epsilon} L_{(i)}. \tag{31}$$

Proof of Lemma 4 Since the sensitivity for every block $i \in [K]$ is $\frac{2}{n}$ (see Section C.1), we get $\Delta_{\mathbb{T}} \leq \frac{2K}{n}$. We can then upper bound the expected privacy error when applying the Laplace mechanism:

$$\begin{aligned}
\mathbb{E}\ \mathcal{L}_{\mathbb{T}}(\mathbb{T}\mu_D, v_{\mathrm{DP}}) &= \mathbb{E} \max_{i=[K]} \mathcal{L}_{\mathbb{T}_{(i)}}(v_{(i)}, v_{(i)} + [\Phi_{(i)}\tilde{\eta}_i]_{1:m_{(i)}}) \\
&= \mathbb{E}\ \max_{i=[K]} \sum_{j_{(i)}=1}^{m_{(i)}} (w_{j_{(i)}+1} - w_{j_{(i)}}) \left| \sum_{l=1}^{j_{(i)}} (\Phi_{(i)}\tilde{\eta}_{(i)})_l \right| \\
&\leq \mathbb{E}\ \max_{i=[K]}\ L_{(i)}\ \max_{j_i\in[m_{(i)}]} \left| \sum_{l=1}^{j_{(i)}} (\Phi_{(i)}\tilde{\eta}_{(i)})_l \right|.
\end{aligned} \tag{32}$$

Next, recall that by the construction of the noise η in the proof for the case where $K = 1$ we have $\tilde{\eta}_{(i)} \sim \left(\mathrm{Lap}\left(\frac{2K\|\Phi_{(i)}^{-1}\|_1}{n\epsilon}\right)\right)^{m_{\Phi_{(i)}}}$. In particular, as in Section C.1, we choose $\Phi_{(i)}$ to be the rescaled Haar matrix as described in Lemma 3, and hence $\tilde{\eta}_{(i)} \sim \left(\mathrm{Lap}\left(\frac{2K}{n\epsilon}\right)\right)^{m_{(i)}}$ with $m_{\Phi_{(i)}} = 2^{\lceil \log_2(m_{(i)})\rceil}$. We can now use essentially the following standard argument as in Section 3.3 in [Boedihardjo et al., 2022]:

Since for every $i, j_{(i)}$, $\frac{n\epsilon}{2K}\tilde{\eta}_{(i),j}$ has sub-exponential norm $\|\tilde{\eta}_{(i),j}\|_{\phi_1} \leq 2$ (see Section 2 in Vershynin [2018]), we can apply Bernstein's inequality, which gives together with Lemma 3 and the fact that $\|\Phi\|_\infty \leq k_{(i)}/2$, for all $i, j_{(i)}$:

$$\begin{aligned}
\mathbb{P}\left(\left| \frac{n\epsilon}{2K} \sum_{l=1}^{j_{(i)}} (\Phi_{(i)}\eta_{(i)})_l \right| \geq t \right) &\leq 2\exp\left(-c\min\left(\frac{t^2}{(k_{(i)}+1)^3}, \frac{t}{k_{(i)}+1} \right) \right) \\
&\leq 2\exp\left(-c\min\left(\frac{t^2}{(k_{\max}+1)^3}, \frac{t}{k_{\max}+1} \right) \right)
\end{aligned} \tag{33}$$

where $k_{\max} = \max_{i\in[K]} k_{(i)}$. We can then upper bound the term in Equation (32) when taking the union bound over at most $K \max_{i\in[K]} m_{(i)} \leq \exp(\log(K) + k_{\max} + \log(2))$ elements. Thus, we obtain the following upper bound for the expectation:

$$\begin{aligned}
&\mathbb{E}\ \max_{i\in[K]} L_{(i)}\ \max_{j_{(i)}\in[m_{(i)}]} \left| \frac{2K}{n\epsilon} \sum_{l=1}^{j_{(i)}} (\Phi_{(i)}\tilde{\eta}_{(i)})_l \right| \\
\lesssim\ &\max_{i\in[K]\}} \frac{2K(\lceil \log_2(m_{(i)})\rceil + 1 + \log(K))^2}{n\epsilon} L_{(i)}.
\end{aligned} \tag{34}$$

C.3 Proof of Theorem 1

We now discuss how Theorem 2 implies Theorem 1. We recall that in this case, $T_{(i)} = [0,1]^s$ and $\rho_{(i)} = \|.\|_\infty$, and thus, for any $k \in \mathbb{N}_{>0}$, we can simply upper bound the covering numbers by $N(T_{(i)}, \|.\|_\infty, t) \leq k^s$. with $t = 1/2k$. Plugging this upper bound into Equation (25) in Theorem 2, we obtain $\mathbb{E}\,\mathcal{U}(\mu_D, \mathcal{A}(D)) \lesssim_s \frac{1}{2k} + \frac{\binom{d}{s}\log(k)^2}{n\epsilon} k^{s-1}$, where $\lesssim_s$ is hiding constants depending on s. We then obtain the desired result when optimizing over k. Finally, we note that the procedure described in the proof of Theorem 2 in Appendix C.2 and C.1 agrees exactly with the Algorithm 1 from Section 3 when $T = [0,1]^d$ equipped with the ℓ_∞ distance function.

D Other types of sparsity: low dimensional data

As shown in [He et al., 2023, Boedihardjo et al., 2022], the expected utility loss when $\mathcal{F}$ is the set of all 1-sparse functions (see Section 6) is of order $\left(\frac{1}{n\epsilon}\right)^{1/d}$. In Section 4 we showed that we can overcome this curse of dimensionality when restricting $\mathcal{F}$ to sparse functions. In this section we consider the case where $\mathcal{F}$ is the set of all 1-Lipschitz functions, and thus $\mathcal{U} = W_1$, and show how this curse of dimensionality can also be overcome when the data lives on a sparse (although unknown) subspace. As we show, this is the case even when we do not have access to any oracle knowledge about the data set, nor the "dimension" of the subspace, as the algorithm is capable of "adapting" to the data set.

Special case: rates on the hyper cube Consider the same setting as in Theorem (1), where the underlying space is the d-dimensional hypercube $T = [0,1]^d$ equipped with the ℓ_∞-metric. Let $T_{k,d} = \{1/2k, \cdots, (2k-1)/2k\}^d$ be the centers of a minimal $1/2k$-covering of T of size $N = k^d$. We have:

Theorem 3. *Let $c \geq 0$ be any constant and let $\mathcal{F}$ be the set of all 1-Lipschtiz continuous functions on $[0,1]^d$ with respect to the ℓ_∞-metric. For any $n\epsilon \geq d+1$, there exists an ϵ-DP algorithm $\mathcal{A}$ such that for any data set D,*

$$\mathbb{E}\ W_1(\mu_D, \mathcal{A}(D)) \lesssim_s \left(\frac{d^3 \log(\epsilon n)}{n\epsilon}\right)^{1/(s+1)} + \frac{d^2 \log(n\epsilon)^2}{n\epsilon^2}, \tag{35}$$

where $s \in \{0, \cdots, d\}$ is the smallest integer such that for all $k \geq 1$, D is contained in at most ck^s ℓ_∞-balls of radius $1/2k$ with centers in $T_{k,d}$.

The proof is a consequence of Theorem 4 below. Note that the definition of s in Theorem 3 resembles the definition of the Minkowski dimension when the data set $D \subset T_s$ lives on a subspace T_s of Minkowski dimension s. The algorithm in Theorem 3 is "adaptive" in the sense that it adjusts to the characteristic s of the data set without relying on any prior information or oracle knowledge of s. When the worst case scenario occurs and $s = d$, the rate in Equation (35) includes an additional term with an exponent of $+1$ compared to Equation (18). This raises the question of whether the cost of adaptivity can be reduced further.

General result Generally, for any $t > 0$ and metric space space (T, ρ), fix a minimal t-covering T_m of size $N(T, \rho, t)$ and let $q_t : T \to T_m$ be any measurable discretization map such that for any point $x \in T$, $\rho(q_t(x), x) \leq t$. Further, let $|q_t(D)|$ be the size of the support of $q_t(D)$ (which captures the "sparsity" of the subspace the data is lying on). We have:

Theorem 4. *In the setting described above, there exists a randomized algorithm $\mathcal{A}$ that takes a data set $D \in T^n$ of size n as input on a metric space (T, ρ) and returns a finitely-supported probability measure $\mathcal{A}(\mu_D)$ on (T, ρ) such that $\mathcal{A}$ is ϵ-DP private and has expected utility loss (2) over the set of all 1-Lipschitz continuous functions with respect to ρ at most:*

$$\mathbb{E}\ W_1(\mu_D, \mathcal{A}(D)) \leq t + \frac{64\ \mathrm{diam}(T) \log(m+1)}{n\epsilon}\ |q_t(D)| \tag{36}$$

D.1 Proof of Theorem 4

As in Section C.1, let $\mathbb{D}_{T_m} : \mathcal{M}_{\mathbb{P}}(T) \to \mathcal{M}_{\mathbb{P}}(T_m)$ be any projection operator and let $\mathbb{T}\mu$ be any vector on the probability simplex representing the measure $\mathbb{D}_{T_m}\mu$. Unlike in Section C.1, we can choose any random indexing of the elements in T_m to represent the vector $\mathbb{T}\mu$.

Data sanitization, Step 2 in Algorithm 1: We are now going to construct a DP vector v_{DP} as in Step 2 in Algorithm 1. A simple way to construct a sparse private variant of v is: first apply the standard Laplace mechanism (with $\Phi = I_m$) to obtain a differential private copy of $\tilde{v}_{\mathrm{DP}} = v + \eta$ of v with η as in Lemma 2 and then solve the convex optimization problem

$$v_{\mathrm{DP}} = \arg\min_{v'} \|v' - \tilde{v}_{\mathrm{DP}}\|_2 \quad \text{s.t. } \|v'\|_1 \leq 1 \tag{37}$$

Standard results for the constrained ℓ_1-norm ERM solution (see e.g., Theorem 7.13 in Wainwright [2019]) then yield the following upper bound on the ℓ_1-error $\|v_{\mathrm{DP}} - v\|_1 \leq 16|q_t(D)|\|\eta\|_\infty$.

Optimization, Step 3 in Algorithm 1 For the proxy utility loss we can simply choose $\mathcal{U}_{\mathbb{T}}^d$ form Section 5.4 (where we set $s = d$). Note that since $\mathcal{F}$ is the set of all 1-Lipschitz queries, we can solve the minimization problem in Step 3 by solving a linear program (see also [He et al., 2023]).

Upper bound for the utility loss W_1: Recall from Section C.1 that the projection error $W_1(\mu_D, \mathbb{T}^\dagger\mathbb{T}\mu_D) \leq t$ from Equation (15) is upper bounded by t. By the same reasoning as in Equation (15), it suffices to upper bound the (expected) privacy error term $\mathbb{E}\, \mathcal{U}_{\mathbb{T}}^d(\mathbb{T}\mu_D, v_{\mathrm{DP}})$ (with $s = d$) in Equation (15) (where we replace $\mathcal{L}_{\mathbb{T}}$ with $\mathcal{U}_{\mathbb{T}}^d$) by:

$$\begin{aligned}\mathbb{E}\, \mathcal{U}_{\mathbb{T}}^d(\mathbb{T}\mu_D, v_{\mathrm{DP}}) &\leq \mathbb{E} \sup_{f\in F; f(0)=0} \sum_{z_i \in T_m} |f(z_i)(v_{\mathrm{DP},i} - v_i)| \overset{\text{Hölder}}{\leq} \mathbb{E} \sup_{f\in F; f(0)=0} \|f\|_{L_\infty} \|v_{\mathrm{DP}} - v\|_1 \\ &\leq \mathrm{diam}(T)\ 16\ |q_t(D)|\ \mathbb{E}\|\eta\|_\infty.\end{aligned} \tag{38}$$

We then obtain the desired result when using the upper bound $\mathbb{E}\ \|\eta\|_\infty \leq \frac{4}{n\epsilon}\log(m+1)$ where we used Example 2.19 in [Wainwright, 2019] in the last line and the fact that $\|\eta\|_{\psi_1} = 2$ for $\eta \sim \mathrm{Lap}(1)$.

D.2 Proof of Theorem 3

Finally, we discuss how we obtain Theorem 3 form Theorem 4. Unlike in the proof of Theorem 1 we can no longer simply optimize over t because we do not have access to $|q_t(D)|$, nor do we assume to have access to the smallest integer s from Theorem 3 such that for all $t = 1/2k$, $|q_{1/2k}(D)| \leq ck^s$.

Instead, we need to "adaptively" optimize over all $s' \in \{0, \cdots, d\}$. For this, in the first step, we want to find for every s' the optimal $t_{s'}$ which minimizes the RHS in Equation (36) in Theorem 4, assuming that for all $t = 1/2k$, $|q_{1/2k}(D)| \leq ck^{s'}$. We choose $t_{s'} = 1/2k_{s'}$ and since $N([0,1]^d, \|.\|_\infty, t_{s'}) = k_{s'}^d$, we can upper bound the RHS in Equation (36) in Theorem 4 by:

$$1/2k_{s'} + \frac{64\log(k_{s'}^d + 1)}{n\epsilon}\ |q_{1/2k_{s'}}(D)| \leq 1/2k_{s'} + 1\frac{64d\log(k_{s'})}{n\epsilon}\ ck_{s'}^{s'}. \tag{39}$$

We can now minimize the RHS by choosing $k_{s'} \asymp \left(\frac{d^2\log(\epsilon n)}{n\epsilon}\right)^{1/(s'+1)}$, which gives

$$1/2k_{s'} + \frac{64\log(k_{s'}^d + 1)}{n\epsilon} \lesssim \left(\frac{d\log(\epsilon n)}{n\epsilon}\right)^{1/(s'+1)}. \tag{40}$$

We run the algorithm in Theorem 4 for every choice of $s' \in \{0, \cdots, d\}$ with $t_{s'}$, resulting in the measures $\mathcal{A}_{s'}(D)$, and by Theorem 4 we have

$$\mathbb{E}\ W_1(\mu_D, \mathcal{A}_{s'}(D)) \leq 1/2k_{s'} + \frac{64\ \log(k_{s'}^d + 1)}{n\epsilon}\ |q_{1/2k_{s'}}(D)|. \tag{41}$$

The problem remains which measure $\mathcal{A}_{s'}(D)$ to return. The idea is to estimate $|q_{1/2k_{s'}}(D)|$ using the estimates $\hat{S}_{s'}$ for the support. More precisely, we release the $d+1$ ϵ-DP estimates for the sizes of the supports:

$$\hat{S}_{s'} = |q_{1/2k_{s'}}(D)| + \frac{1}{\epsilon}\xi_{s'} \quad \text{with} \quad \xi_{s'} \sim \mathrm{Lap}(1) \tag{42}$$

where we used that the sensitivity of the support function is trivially 1. We can now return the measure $\mathcal{A}_{s_{\mathrm{opt}}}(D)$ with

$$s_{\mathrm{opt}} = \arg\min_{s'} 1/2k_{s'} + \frac{64\ \log(k_{s'}^d + 1)}{n\epsilon}\ \hat{S}_{s'}. \tag{43}$$

Note that by the composition theorem for differential privacy [Dwork et al., 2006], the overall algorithm is therefore $2(d+1)\epsilon$-DP, and we obtain an ϵ-DP algorithm by simply replacing ϵ with $\tilde{\epsilon} = \epsilon/(2d+1)$.

Upper bound for the expected utility loss: To prove the result in Theorem 3, we need to upper bound the expected utility loss. We divide the upper bound into two parts, where we let $\mathcal{E}$ be the event where

$$\mathcal{E}: \quad \max_{s' \in \{0,\cdots,d\}} |\xi_{s'}| \leq 4\log(n\epsilon), \tag{44}$$

and note that (using $n\epsilon \geq d+1$), $\mathbb{P}(\mathcal{E}^c) \leq \frac{1}{(\epsilon n)^2}$. Since the utility loss is at most 1 (because the transportation cost is at most $\text{diam}(T) = 1$), we have that $\mathbb{E}\left[W_1(\mu_D, \mathcal{A}_{s_{\text{opt}}}(D))|\mathcal{E}^c\right] \leq 1$. Moreover, we can bound:

$$\begin{aligned}\mathbb{E}_{\xi,\eta}\ W_1(\mu_D, \mathcal{A}_{s_{\text{opt}}}(D)) &\leq \mathbb{E}_{\xi,\eta}\ \left[W_1(\mu_D, \mathcal{A}_{s_{\text{opt}}}(D))|\mathcal{E}\right] + P(\mathcal{E}^c) \\ &\leq \mathbb{E}_{\xi,\eta}\ \left[W_1(\mu_D, \mathcal{A}_{s_{\text{opt}}}(D))|\mathcal{E}\right] + \frac{1}{(n\epsilon)^2}.\end{aligned}$$

Thus, we are only left with bounding the expected utility loss conditioning on $\mathcal{E}$. Note that the expectation in Equation (41) is only over η, and thus:

$$\begin{aligned}&\mathbb{E}_{\xi,\eta}\ \left[W_1(\mu_D, \mathcal{A}_{s_{\text{opt}}}(D))|\mathcal{E}\right] \\ &\quad\leq \mathbb{E}_{\xi}\left[1/2k_{s_{\text{opt}}} + \frac{64\ \log(k^d_{s_{\text{opt}}}+1)}{n\tilde{\epsilon}}\ |q_{1/2k_{s_{\text{opt}}}}(D)|\big|\mathcal{E}\right] \\ &\quad\leq 1/2k_s + \frac{64\ \log(k^d_s+1)}{n\tilde{\epsilon}}\ \left(|q_{1/2k_s}(D)| + 4\frac{\log(\epsilon n)}{\tilde{\epsilon}}\right) \\ &\quad\lesssim \left(\frac{d^2\log(\epsilon n)}{n\epsilon}\right)^{1/(s+1)} + \frac{d^3\log^2(n\epsilon)}{n\epsilon^2},\end{aligned} \tag{45}$$

where we used in the last line the assumption that for all k, $|q_{1/2k_{s'}}(D)| \leq ck^s$ and Equation (39) and recall that $k_s \asymp \left(\frac{d^2\log(\epsilon n)}{n\epsilon}\right)^{1/(s+1)}$.

E Proof sketch for the lower bound in Equation (14)

We now present a proof sketch for the lower bound in Equation (14), which follows from a standard geometric argument pioneered in [Hardt and Talwar, 2010] and used in [Boedihardjo et al., 2022]. The key idea is to apply the following corollary of Proposition 8.1 in [Boedihardjo et al., 2022]. Adapted to this setting, the proposition states:

Corollary 1 (Proposition 8.1 in [Boedihardjo et al., 2022])**.** *Let $M_0 = T^n$ be the set of all datasets of size n, and let $\mathcal{M}_n(T)$ be the corresponding set of all empirical measures constructed from datasets in M_0. Further, let $M_1 = \mathcal{M}_{\mathbb{P}}(T)$ be the set of all probability measures. Let ρ_1 be any metric on M_1, and assume that for some $t, \epsilon > 0$ the packing number is lower bounded by*

$$N_{pack}(\mathcal{M}_n(T), \rho_1, t) > 2e^{\epsilon n}.$$

Then, for any randomized algorithm $\mathcal{A}: M_0 \to M_1$ that is ϵ-differentially private, there exists $D \in M_0$ such that

$$\mathbb{E}_{\mathcal{A}}\ \ \rho_1(\mathcal{A}(D), \mu_D) > t/4.$$

Thus, to obtain the minimax lower bound from Equation (14), it suffices to lower bound the packing number $N_{\text{pack}}(\mathcal{M}_n(T), \rho_1, t)$ and apply the corollary with $\rho_1 = \mathcal{U}$. In the case where $s = d$, and thus $\mathcal{U} = W_1$, Boedihardjo et al. [2022] construct datasets of size $l \leq n$ (we let n be a multiple of l) by drawing uniform samples from T. Leveraging concentration bounds, we then obtain from Proposition 8.2 and 8.6 in [Boedihardjo et al., 2022] that whenever $N_{\text{pack}}(T, \|.\|_\infty, t) \geq 2l$, we have

$$N_{\text{pack}}(\mathcal{M}_l(T), W_1, t/3) \geq \exp(cl), \tag{46}$$

where W_1 is the Wasserstein-1 distance. To extend the lower bound to the case where the metric $\rho_1 = \mathcal{U}$ is the maximum Wasserstein distance over all s-sparse subspaces, as defined Equation (2), we split the dimensions

$\{1, \cdots, d\}$ into $\lfloor d/s \rfloor$ disjoint sets of size s. By the argument above, assuming that $N_{\text{pack}}([0,1]^s, \|.\|_\infty, t) \geq 2l$, we can then construct $\exp(cl)$ many empirical measures on each subspace. Since the dimensions are distinct, we have shown that

$$N_{\text{pack}}(\mathcal{M}_l(T), \mathcal{U}, t/3) \geq \exp(\lfloor d/s \rfloor cl), \tag{47}$$

The bound then follows from $N_{\text{pack}}([0,1]^s, \|.\|_\infty, t) \asymp (1/t)^s$ and choosing $t \asymp \left(\lfloor d/s \rfloor \frac{1}{\epsilon n}\right)^{1/s}$, as in Theorem 9.4 in [Boedihardjo et al., 2022]

F Comparison with approaches minimizing the Euclidean distance

In this section we present a proof sketch for the rate in the last paragraph in Section 4.1. More precisely, we discuss the utility loss of a modified version of Algorithm 1 based on the Euclidean distance, as used in previous works (see McKenna et al. [2021, 2022], Zhang et al. [2017] and references therein). While the mentioned papers present approximate, efficient algorithms, they all rely on the same patter where, similarly to Algorithm 1, we

Step 2 in Alg. 1: draw i.i.d. Laplace (resp. Gaussian) noise $\eta \sim \left(\text{Lap}\left(\frac{\triangle_{\mathbb{T}}}{\epsilon}\right)\right)^m$ and set $v_{\text{DP}} = v + \eta = \mathbb{T}\mu_D + \nu$

Step 3 in Alg. 1: construct a DP measure $\mathcal{A}(D) = \arg\min_{\mu \in \mathcal{M}_{\mathbb{P}}(T)} \sum_{S \subset \{1,\cdots d\}; |S|=s} \|v^S_{\text{DP}} - (\mathbb{T}\mu)^S\|_2^2$

We slightly abuse the notation in Step 3 above by treating v^S_{DP} as a vector. Following the same argument as in the proof of Theorems 2-4, we can bound the expected utility loss as follows. First, similar to Equation (15), we can upper bound

$$\mathbb{E}\left[\mathcal{U}(\mu_D, \mathcal{A}(D))\right] \leq \mathcal{U}(\mu_D, \mathbb{T}^{-1}\mathbb{T}\mu_D) + \mathbb{E}\left[\mathcal{U}(\mathbb{T}^{-1}\mathbb{T}\mu_D, \mathbb{T}^{-1}\mathbb{T}\mathcal{A}(D))\right]$$

where we made use of the fact we can choose $\mathcal{A}(D)$ such that $\mathcal{A}(D) = \mathbb{T}^{-1}\mathbb{T}\mathcal{A}(D)$ without increasing the utility loss. We recall that by construction, the first term is at most $1/2k$ (see proof of Theorem 1 and 2), where k is the grid size of the ℓ_∞-covering of $T = [0,1]^d$ used to construct $\mathbb{T}$. To bound the second term, note that we have (using the notation from Equation (17))

$$\begin{aligned}
\mathcal{U}(\mathbb{T}^{-1}\mathbb{T}\mu_D, \mathbb{T}^{-1}\mathbb{T}\mathcal{A}(D)) &= \max_{\substack{S\subset[d]\\|S|=s}} \sup_{\substack{f\in\mathcal{F}(T_{k,s})\\f(0)=0}} \sum_{z_i \in T_m} |f(z_i)((\mathbb{T}\mathcal{A}(D))^S_i - v^S_i)| \\
&\overset{\text{Hölder}}{\leq} \max_{\substack{S\subset[d]\\|S|=s}} \sup_{\substack{f\in\mathcal{F}(T_{k,s})\\f(0)=0}} \sqrt{m}\|f\|_{L_\infty} \|(\mathbb{T}\mathcal{A}(D))^S - v^S\|_2 \\
&\leq \max_{\substack{S\subset[d]\\|S|=s}} \sup_{\substack{f\in\mathcal{F}(T_{k,s})\\f(0)=0}} 2\sqrt{m}\ \text{diam}(T)\|v_{\text{DP}} - v^S\|_2 = \max_{\substack{S\subset[d]\\|S|=s}} 2\sqrt{m}\ \text{diam}(T)\|\eta^S\|_2,
\end{aligned}$$

where we recall that v^S is a vector of size m. Using the fact that $\triangle_T = 2\binom{d}{s}/n$, we can apply standard concentration bounds to show that $\mathbb{E}\,\mathcal{U}(\mathbb{T}^{-1}\mathbb{T}\mu_D, \mathbb{T}^{-1}\mathbb{T}\mathcal{A}(D)) = \tilde{O}_{s,d}(\frac{m}{n\epsilon})$ and thus $\mathbb{E}\,\mathcal{U}(\mathcal{A}(D), \mu_D) \leq 1/(2k) + \tilde{O}_{s,d}(\frac{k^d}{n\epsilon})$, where we used that $m = k^d$. Optimizing over k yields $\mathbb{E}\,\mathcal{U}(\mathcal{A}(D), \mu_D) = \tilde{O}_{d,s}((\epsilon n)^{-1/(s+1)})$, as desired.