

Privacy-Computation Trade-Offs in Private Repetition and Metaselection

Kunal Talwar  

Apple, Cupertino, CA, USA

Abstract

A Private Repetition algorithm takes as input a differentially private algorithm with constant success probability and boosts it to one that succeeds with high probability. These algorithms are closely related to private metaselection algorithms that compete with the best of many private algorithms, and private hyperparameter tuning algorithms that compete with the best hyperparameter settings for a private learning algorithm. Existing algorithms for these tasks pay either a large overhead in privacy cost, or a large overhead in computational cost. In this work, we show strong lower bounds for problems of this kind, showing in particular that for any algorithm that preserves the privacy cost up to a constant factor, the failure probability can only fall polynomially in the computational overhead. This is in stark contrast with the non-private setting, where the failure probability falls exponentially in the computational overhead. By carefully combining existing algorithms for metaselection, we prove computation-privacy tradeoffs that nearly match our lower bounds.

2012 ACM Subject Classification Security and privacy; Theory of computation → Theory and algorithms for application domains

Keywords and phrases Differential Privacy, Hyperparameter Tuning, Metaselection

Digital Object Identifier 10.4230/LIPIcs.FORC.2025.1

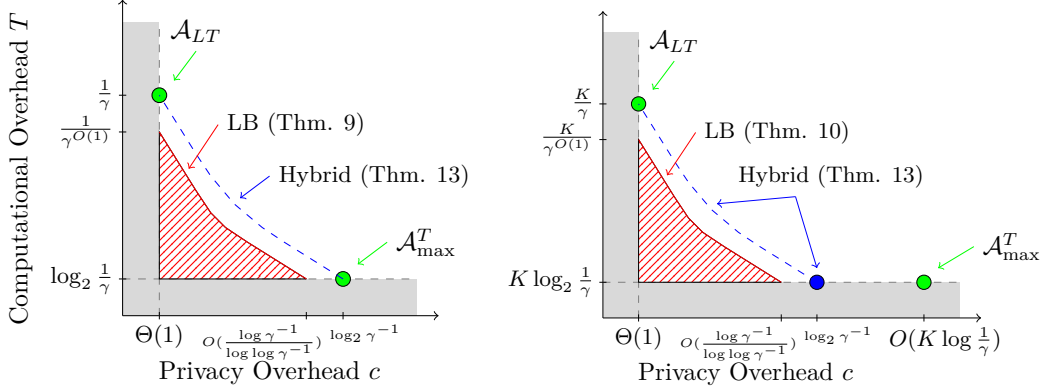
Related Version *Full Version*: <https://arxiv.org/abs/2410.19012>

1 Introduction

Randomized algorithms have probabilistic guarantees. Often one designs an algorithm that succeeds (for an appropriate notion of success) with constant probability. A standard approach to boosting the success probability of a randomized algorithm is to run the algorithm with fresh randomness multiple times and take the best of the results of individual runs. Standard tail inequalities then allow for proving bounds on the success probability of the resulting algorithm. Suppose that an algorithm \mathcal{A} produces an output with quality score at least m with probability at least $\frac{1}{2}$.¹ As an example, the algorithm of [13] for finding a maximum cut in the graph finds a cut with value within a small constant factor of the optimum with constant probability. Let \mathcal{A}_{\max}^T be the algorithm that runs \mathcal{A} T times to get outputs o_1, \dots, o_T and releases the o_i with the highest quality score. The basic *repetition theorem* says that \mathcal{A}_{\max}^T produces an output with quality score at least m except with probability $\frac{1}{2^T}$. An important aspect of this result is the relationship between the failure probability and the number of repetitions: logarithmically many repetitions suffice to make the failure probability polynomially small. In the example above, repeating the maximum cut algorithm 20 times ensures that the best cut amongst these is approximately optimal except with probability 10^{-6} . Various versions of repetition theorems have been studied in literature, e.g. improving on the amount of randomness needed [19], or allowing for parallelizability in case of multiple round protocols [27, 18].

¹ Without loss of generality, we assume our goal is to maximize a quality score; our results can be translated immediately to a setting where want to minimize a quality score. The constant $\frac{1}{2}$ can be replaced by any other constant.





■ **Figure 1** Existing and new results on Private Repetition (left) and Private Hyperparameter tuning and Metaselection (right). Green dots show the existing upper bounds \mathcal{A}_{LT} and \mathcal{A}_{\max}^T discussed above, and the gray regions depict the existing excluded regions. The red line shows our new lower bounds: we exclude the full region below the red line. The lower bound nearly matches the blue dashed line that depicts our analysis of hybrid algorithms.

In this work, we are interested in this question when the algorithm \mathcal{A} is differentially private (see Section 2 for a precise definition). We assume we have a differentially private algorithm that outputs an object (e.g. an ML model) and a score (e.g. approximate accuracy on a *test* dataset), where the score is “high” with constant probability. Private Repetition theorems allow for boosting the success probability, while controlling the privacy cost. A simple repetition theorem uses the \mathcal{A}_{\max}^T above. To get failure probability below γ , it would suffice to set $T = \log_2 \frac{1}{\gamma}$. This increases the privacy parameters (e.g. the ε in (ε, δ) -DP) of the algorithm by a factor of T which may be undesirable. Liu and Talwar [20] designed and analyzed a different algorithm for private repetition that we will refer to as \mathcal{A}_{LT} for the rest of introduction. The algorithm \mathcal{A}_{LT} allows for a constant ($3\times$) increase in the privacy cost, while allowing an arbitrarily small failure probability γ . However, it requires running the algorithm \mathcal{A} about $O(\frac{1}{\gamma})$ times, instead of the logarithmic dependence on γ^{-1} in the naive repetition theorem.

In other words, existing private repetition theorems either pay a $\log \frac{1}{\gamma}$ overhead in the privacy cost, or a $\text{poly}(\frac{1}{\gamma})$ overhead in the run time. A similar dichotomy exists for private hyperparameter selection, where one wants to privately select the best model amongst those resulting from different hyperparameter choices. Indeed private hyperparameter tuning is one of the standard uses of private repetition theorems [20, 26] and the \mathcal{A}_{LT} algorithm has been used in practice recently by Israel’s Ministry of Health in their release of 2014 live births microdata [17]. The private learning algorithms for a single hyperparameter setting can be computationally expensive to run. As an example, in Hod and Canetti [17], the algorithm \mathcal{A} can involve training a CTGAN [33, 34, 28] using DPSGD [1] or PATE [24, 25] for a certain set of hyperparameters. In such cases, the $\text{poly}(\frac{1}{\gamma})$ run time overhead can be quite significant.

As we show in Section 4, the two algorithms above can be combined to give a smooth trade-off between the privacy cost overhead and the computational overhead. It is natural to ask however if such a trade-off is necessary: is there a way to keep the privacy overhead to constant while paying only logarithmically in the run time?

In this work, we give a negative answer to this question. We show that for any algorithm with constant privacy overhead, the run time must necessarily be polynomial in $\frac{1}{\gamma}$. More generally, our lower bound shows that the trade-off between the computational overhead (denoted by T below) and the privacy overhead (denoted by c below) is nearly tight.

► **Theorem** (Informal version of Theorem 9). *Let \mathcal{A} be an oracle algorithm that satisfies the following properties:*

Privacy: For any ε -DP mechanism $\mathcal{M} : D^n \rightarrow (R \times \mathbb{R})$, $\mathcal{A}^\mathcal{M}$ is $(c\varepsilon, \delta)$ -DP.

Utility: For any input d , let m be such that $\Pr[\text{val}(\mathcal{M}(d)) \geq m] \geq \frac{1}{2}$. Then $\mathcal{A}^\mathcal{M}$ satisfies

$$\Pr[\text{val}(\mathcal{A}^\mathcal{M}(d)) \geq m] \geq 1 - \gamma.$$

Runtime: \mathcal{A} on input d makes at most T calls to \mathcal{M} (on any inputs) in expectation.

Then for $\delta < O(\frac{\varepsilon\gamma}{\ln T})$, $T \geq \gamma^{\Omega(1/c)}$, or equivalently, $c \geq \Omega(\frac{\ln \frac{1}{\gamma}}{\ln T})$.

These results are shown pictorially in Figure 1 (left). The two green filled dots denote the trade-offs achieved by \mathcal{A}_{LT} and \mathcal{A}_{\max}^T . The blue dashed line is the upper bound achieved by combining these algorithms (Section 4). The grey shaded region can be excluded by straightforward arguments. The lower bound above excludes the red striped region. In particular, it shows that for any constant c , T must be at least $\text{poly}(\frac{1}{\gamma})$. At the other extreme, any T that is polylogarithmic in $\frac{1}{\gamma}$ requires $c \geq \Omega(\frac{\ln \frac{1}{\gamma}}{\ln \ln \frac{1}{\gamma}})$. These results hold even when the target value m is public.

We extend our results to the setting where the target value m is achieved by the mechanism \mathcal{M} with a probability that is different from $\frac{1}{2}$. More generally, in the hyperparameter tuning setting, there are K possible settings of hyperparameters, and our utility bound asks that the algorithm be competitive with the median value of the best hyperparameter setting.

► **Theorem** (Informal version of Theorem 10). *Let \mathcal{A} be an oracle algorithm that satisfies the following properties:*

Privacy: For any ε -DP mechanism $\mathcal{M} : D^n \times [K] \rightarrow (R \times \mathbb{R})$, $\mathcal{A}^\mathcal{M} : D^n \rightarrow (R \times \mathbb{R})$ is $(c\varepsilon, \delta)$ -DP.

Utility: For any input d , let m be such that $\Pr_{\mathcal{M}} \Pr_{k \sim [K]}[\text{val}(\mathcal{M}(d, k)) \geq m] \geq \frac{1}{2K}$. Then $\Pr[\text{val}(\mathcal{A}^\mathcal{M}(d)) \geq m] \geq 1 - \gamma$.

Runtime: \mathcal{A} on input d makes at most T calls to \mathcal{M} (on any inputs) in expectation.

Then for $\delta < O(\frac{\varepsilon\gamma}{\ln T})$, $T \geq K\gamma^{\Omega(\frac{1}{c})}$, or equivalently, $c \geq \Omega(\frac{\ln \frac{1}{\gamma}}{\ln(T/K)})$.

We show these results in Figure 1 (right). As before, the two green filled dots denote the trade-offs achieved by \mathcal{A}_{LT} and \mathcal{A}_{\max}^T . Note the here \mathcal{A}_{\max}^T is significantly worse, and the better upper bounds is achieved by combining \mathcal{A}_{LT} with \mathcal{A}_{\max}^T . As before the grey shaded region is excluded by straightforward arguments, and the new lower bound exclude the red striped region. In particular, it shows that for any constant c , T must be at least $K \cdot \text{poly}(\frac{1}{\gamma})$. At the other extreme, making T/K polylogarithmic in $\frac{1}{\gamma}$ requires $c \geq \Omega(\frac{\ln \frac{1}{\gamma}}{\ln \ln \frac{1}{\gamma}})$. As before, the lower bound holds for known target value m .

This has strong implications for private hyperparameter tuning. Indeed our results imply that any private hyperparameter tuning algorithm that boosts the success probability to $(1 - \gamma)$ while paying only a constant overhead in privacy cost must pay a $K \cdot \text{poly}(\gamma^{-1})$ computational overhead.

We remark that both the upper bounds translate ε -DP algorithms to $c\varepsilon$ -DP algorithms, and (ε, δ) -DP algorithms to $(c\varepsilon, T\delta)$ -DP algorithms; similar results hold for Renyi DP algorithms as well [26]. Our lower bounds apply even when the input algorithm is ε -DP, and the output algorithm is allowed to be (ε, δ) -DP (or Renyi DP).

Other Related Work. Gupta et al. [14] first proved a repetition theorem for the case when the target threshold is known and the value is a low-sensitivity function. Their result required $T = O(1/\gamma^2)$ for repetition and $T = O(K^2/\gamma^2)$ for metaselection. As discussed, the random

stopping approach of [20] improves these to $T = O(1/\gamma)$ and $T = O(K/\gamma)$ without any restrictions. Subsequently, [26] studied these questions under Renyi Differential privacy, and showed positive results for a variety of distributions of stopping times.

There is also a beautiful line of work on meta-algorithms that run multiple private algorithms iteratively, and only pay for privacy cost of those that meet a certain criteria. The Sparse Vector Technique [10, 29, 15, 32] is the simplest version of such a meta-algorithm, where one only pays privacy cost for (numerical, bounded-sensitivity) queries whose answer is above a threshold. This paradigm was significantly expanded by Cohen and Lyu [6] who show a similar result for general target sets, under appropriate assumptions on the algorithms.

Decision vs. Optimization. Repetition theorems in complexity theory are normally stated for decision problems. This is justified by the fact that one can reduce optimization problems to decision problems with logarithmic (in the range) overhead. However a logarithmic overhead in privacy cost would be quite significant as one typically wants ε to be a small constant. For this reason, all the above works [14, 20, 26] directly address optimization problems. As decision problems can be reduced to optimization, our results also apply to decision problems.

The need for Private Repetition. Private algorithms come in many flavors. In many cases, the algorithm itself or a slight modification can be better analyzed to make its failure probability small, at a small cost to the utility criteria. For example, the Laplace mechanism [9] and the exponential mechanism [21] can directly give utility bounds that hold with probability $(1 - \gamma)$ for any γ . In these cases, we do not need private repetition algorithms that use a base algorithm as an oracle. However, for many other private algorithms, we do not have such sliding scale guarantees and can only control the expected utility, or median utility. This is often the case when the randomization plays a role in “non-private” part of the algorithm, e.g. in the use of tree embeddings [14, 12], hashing [3] or other more custom analyses [14, 7]. Finally, for some problems, (private) algorithms work well in practice but may not have theoretical guarantees (or work much better than their theoretical guarantees) and one does not have a knob to tune the failure probability. For these two kinds of algorithms, private repetition theorems are an invaluable tool for reducing the failure probability.

Metaselection and hyperparameter tuning algorithms are even more broadly useful. It is fairly common to have algorithms that take a hyperparameter as input and work well either provably or empirically. As an example, algorithms in the Propose-test-release framework [8] work well when the proposal is true for the distribution. In cases when the proposal can be parameterized (e.g. in [5]), it becomes a hyperparameter. Practical private optimization algorithms (e.g. [1]) typically involve multiple hyperparameters. Often the hyperparameters cannot be well-tuned on public proxy datasets, and can be a source of leakage of private information [26]. Private Hyperparameter tuning algorithms are practically used in such settings and the computational cost can be a significant concern.

2 Preliminaries

Differential Privacy [9] constrains the distribution of outcomes on neighboring datasets to be close. The Hamming distance $|d - d'|_H$ between two datasets $d, d' \in \mathcal{D}^n$ is the number of entries in which they differ. For our purposes, two datasets $d, d' \in \mathcal{D}^n$ are *neighboring* if they differ in one entry, i.e. if their Hamming distance is 1.

► **Definition 1.** A randomized algorithm $\mathcal{M} : \mathcal{D}^n \rightarrow \mathcal{R}$ is (ε, δ) -differentially private (DP) if for any pair of neighboring datasets d, d' and any $S \subseteq \mathcal{R}$,

$$\Pr[\mathcal{M}(d) \in S] \leq e^\varepsilon \cdot \Pr[\mathcal{M}(d') \in S] + \delta.$$

We abbreviate $(\varepsilon, 0)$ -DP as ε -DP and refer to it as pure differential privacy.

Sometimes it is convenient to define a mechanism on a subset of inputs, and establish its privacy properties on this restricted subset. The following extension lemma due to Borgs et al. [4] shows that any such partial specification can be extended to the set of all datasets at a small increase in the privacy cost.

► **Proposition 2 (Extension Lemma).** Let \mathcal{M}_B be an ε -differentially private algorithm restricted to $B \subseteq \mathcal{D}^n$ so that for any $d, d' \in B$, and any $S \subseteq \mathcal{R}$, $\frac{\Pr[\mathcal{M}_B(d) \in S]}{\Pr[\mathcal{M}_B(d') \in S]} \leq \exp(\varepsilon \cdot |d - d'|_H)$. Then there exists a randomized algorithm \mathcal{M} defined on the whole input space \mathcal{D} which is 2ε -differentially private and satisfies that for every $d \in B$, $\mathcal{M}(d)$ has the same distribution as $\mathcal{M}_B(d)$.

We will be interested in private mechanisms that output a tuple $(o, v) \in \mathcal{R} \subset \mathcal{R}_o \times \mathbb{R}$ where \mathcal{R}_o is some arbitrary output space. For example, if the mechanism is computing an approximate maximum cut in a graph, o could be a subset of vertices of a graph and v the (approximate) number of edges in the input graph that leave this subset o . In the case of hyperparameter tuning, o would be a model and v an estimate of its accuracy. We denote by $\text{val}(\mathcal{M}(d))$ the real value v when $\mathcal{M}(d) = (o, v)$. For a mechanism $\mathcal{M} : \mathcal{D}^n \rightarrow \mathcal{R}_o \times \mathbb{R}$, we let $\text{Median}(\mathcal{M}(d)) = \sup\{z : \Pr_{(o,v) \sim \mathcal{M}(d)}[v \geq z] \geq \frac{1}{2}\}$ denote the median of v when $(o, v) \sim \mathcal{M}(d)$. A private amplification algorithm takes as input a dataset d and a failure probability γ , and given oracle access to a mechanism \mathcal{M} , outputs an $(o, v) \in \mathcal{R}$ such that $v \geq \text{Median}(\mathcal{M}(d))$ with probability $(1 - \gamma)$. Here this probability is taken over both the internal randomness of the amplification algorithm as well as the randomness in \mathcal{M} . A private amplification algorithm may need to make T oracle calls to \mathcal{M} , and may be $c\varepsilon$ -DP whenever \mathcal{M} is ε -DP. Our goal is to understand the trade-off between the computational overhead T and the privacy overhead c .

We will assume that \mathcal{A} can only output an (o, v) tuple that is produced by a run of \mathcal{M} on some input. This is a natural restriction in all settings, and is needed to make the problem meaningful.² Thus in this paper, we will always make the assumption that the oracle algorithm \mathcal{A} selects a tuple (o, v) from the outputs it receives from calls to \mathcal{M} .

A private metaselection algorithm operates on a set of K private algorithms $\mathcal{M}_1, \dots, \mathcal{M}_K$, where each $\mathcal{M}_i : \mathcal{D}^n \rightarrow \mathcal{R} \subset \mathcal{R}_o \times \mathbb{R}$. It takes as input a dataset d and a failure probability γ , and given oracle access to mechanisms \mathcal{M}_i , outputs an $(o, v) \in \mathcal{R}$ such that $v \geq \max_i \text{Median}(\mathcal{M}_i(d))$ with probability $(1 - \gamma)$. As above, the probability is taken over both the internal randomness of the metaselection algorithm as well as the randomness in \mathcal{M}_i 's. The metaselection algorithm may need to make T oracle calls to the \mathcal{M}_i 's, and may be $c\varepsilon$ -DP whenever each \mathcal{M}_i is ε -DP. Our goal as before is to understand the trade-off between the computational overhead T and the privacy overhead c . Note that hyperparameter tuning can be phrased as metaselection, by treating the private training algorithm for each setting of hyperparameters as a separate algorithm amongst K options. More generally, we can phrase metaselection and hyperparameter tuning as special cases of repetition, where the target

² E.g., a trivial algorithm that computes $(o, v) \sim \mathcal{M}(d)$ and outputs (o, ∞) would not be useful in any application.

value is the $(1 - \frac{1}{2K})$ th quantile of the distribution, instead of the median. This variant can handle the case when the set of hyperparameter settings may be large (or even unbounded) but we expect a non-trivial measure of the hyperparameter settings to be good.

► **Proposition 3.** *Let $\beta > 0$ and \mathcal{A} be an oracle algorithm that satisfies the following properties:*

Privacy: For any ε -DP mechanism $\mathcal{M} : D^n \rightarrow (R \times \mathbb{R})$, $\mathcal{A}^\mathcal{M}$ is $(c\varepsilon, \delta)$ -DP.

Utility: For any input d , let m be such that $\Pr[\text{val}(\mathcal{M}(d)) \geq m] \geq \beta$. Then $\mathcal{A}^\mathcal{M}$ satisfies

$$\Pr[\text{val}(\mathcal{A}^\mathcal{M}(d)) \geq m] \geq 1 - \gamma.$$

Runtime: \mathcal{A} on input d makes at most $T = T(c, \beta)$ calls to \mathcal{M} (on any inputs) in expectation. Let H be a hyperparameter space, and let μ_H be a measure on H . Then there is an oracle algorithm $\tilde{\mathcal{A}}$ that satisfies the following properties:

Privacy: For any ε -DP mechanism $\mathcal{M} : D^n \times H \rightarrow (R \times \mathbb{R})$, $\tilde{\mathcal{A}}^\mathcal{M}$ is $(c\varepsilon, \delta)$ -DP.

Utility: For any input d , let m be such that $\Pr_{h \sim \mu_H}[\text{val}(\mathcal{M}(d)) \geq m] \geq \beta$. Then

$$\Pr[\text{val}(\tilde{\mathcal{A}}^\mathcal{M}(d)) \geq m] \geq 1 - \gamma.$$

Runtime: $\tilde{\mathcal{A}}$ on input d makes at most $T = T(c, \beta)$ calls to \mathcal{M} (on any inputs) in expectation. In particular, when $H = [K]$ and μ_H is uniform, then $\tilde{\mathcal{A}}$ competes with the largest median value $\max_{k \in [K]} \text{Median}(\text{val}(\mathcal{M}(d, k)))$ and makes $T = T(c, \frac{1}{2K})$ calls to \mathcal{M} .

Proof. Define \mathcal{M}' that on input d , samples a random $h \sim \mu_H$ and runs $\mathcal{M}(d, h)$. Running \mathcal{A} with \mathcal{M}' gives the first result. The second result uses the fact that the output of \mathcal{M}' when $H = [K]$ and μ is uniform is at least the largest median with probability at least $\frac{1}{2K}$. ◀

Differential privacy constrains the likelihood of events on neighboring datasets. This also implies constraints for datasets at some bounded distance. The following forms of these constraints will be useful. The proofs of the following consequences of group privacy [11] are elementary.

► **Proposition 4.** *Let \mathcal{M} satisfy ε -DP. Then for datasets d, d' such that $|d - d'|_H \leq \Delta$, and any event E ,*

$$\Pr[\mathcal{M}(d) \in E] \leq e^{\varepsilon\Delta} \Pr[\mathcal{M}(d') \in E].$$

In particular, if $\Pr[\mathcal{M}(d) \in E] \geq \frac{1}{4}$, then $\Pr[\mathcal{M}(d') \in E] \geq e^{-\varepsilon\Delta}/4$.

► **Proposition 5.** *Let \mathcal{M} satisfy (ε, δ) -DP. Then for data sets d, d' such that $|d - d'|_H = \Delta$, and any event E ,*

$$\Pr[\mathcal{M}(d) \in E] \leq e^{\varepsilon\Delta} (\Pr[\mathcal{M}(d') \in E] + \Delta\delta).$$

In particular, if $\delta < e^{-\varepsilon\Delta}/8\Delta$ and $\Pr[\mathcal{M}(d) \in E] \geq \frac{1}{4}$, then $\Pr[\mathcal{M}(d') \in E] \geq e^{-\varepsilon\Delta}/8$.

A similar statement holds for Renyi differential privacy.

► **Proposition 6.** *Let \mathcal{M} satisfy (α, ε) -Renyi DP for $\alpha > 1$. Then for data sets d, d' such that $|d - d'|_H = \Delta$, and any event E ,*

$$\Pr[\mathcal{M}(d) \in E] \leq e^{\varepsilon \sum_{i=1}^{\Delta} (1 - \frac{1}{\alpha})^i} \cdot (\Pr[\mathcal{M}(d') \in E])^{(1 - \frac{1}{\alpha})^\Delta}.$$

In particular, for $\alpha \geq \Delta + 1$, we get

$$\Pr[\mathcal{M}(d) \in E] \leq e^{(\alpha-1)\varepsilon} \cdot (\Pr[\mathcal{M}(d') \in E])^{1/e}.$$

It follows that for $\alpha \geq \Delta + 1$, if $\Pr[\mathcal{M}(d) \in E] \geq \frac{1}{4}$, then $\Pr[\mathcal{M}(d') \in E] \geq e^{-e(\alpha-1)\varepsilon}/44$.

Proof. Let $d' = d_0, d_1, \dots, d_\Delta = d$ be a sequence of datasets where d_i and d_{i+1} are adjacent. A consequence of (α, ε) -RDP [22, Prop. 10] is that for any event E ,

$$\Pr[\mathcal{M}(d_{i+1}) \in E] \leq e^{\varepsilon(1-\frac{1}{\alpha})} \cdot (\Pr[\mathcal{M}(d_i) \in E])^{(1-\frac{1}{\alpha})}.$$

This implies the base case ($k = 1$) of the claim that for all k ,

$$\Pr[\mathcal{M}(d_k) \in E] \leq e^{\varepsilon \sum_{i=1}^k (1-\frac{1}{\alpha})^i} \cdot (\Pr[\mathcal{M}(d_0) \in E])^{(1-\frac{1}{\alpha})^k}.$$

Suppose that the claim holds for d_{k-1} . We now inductively prove it for d_k . We write

$$\begin{aligned} \Pr[\mathcal{M}(d_k) \in E] &\leq e^{\varepsilon(1-\frac{1}{\alpha})} \cdot (\Pr[\mathcal{M}(d_{k-1}) \in E])^{(1-\frac{1}{\alpha})} \\ &\leq e^{\varepsilon(1-\frac{1}{\alpha})} \cdot \left(e^{\varepsilon \sum_{i=1}^{k-1} (1-\frac{1}{\alpha})^i} \cdot (\Pr[\mathcal{M}(d_0) \in E])^{(1-\frac{1}{\alpha})^{k-1}} \right)^{(1-\frac{1}{\alpha})} \\ &= e^{\varepsilon \sum_{i=1}^k (1-\frac{1}{\alpha})^i} \cdot (\Pr[\mathcal{M}(d_0) \in E])^{(1-\frac{1}{\alpha})^k}. \end{aligned}$$

This completes the proof of the first part. For the second part, we use the following two facts. Firstly, for $\alpha \geq \Delta + 1$, $(1 - \frac{1}{\alpha})^\Delta > \frac{1}{e}$, so that the exponent of $(\Pr[\mathcal{M}(d_0) \in E])$ is at least $(1/e)$ (when $k = \Delta$). Secondly, the geometric series $\sum_{i=1}^\Delta (1 - \frac{1}{\alpha})^i \leq \sum_{i=1}^\infty (1 - \frac{1}{\alpha})^i = \alpha - 1$. The third part follows immediately by rearrangement. ◀

3 Main Lower Bound

► **Theorem 7.** Let \mathcal{A} be an oracle algorithm that satisfies the following properties:

Privacy: For any ε -DP mechanism $\mathcal{M} : D^n \rightarrow (R \times \mathbb{R})$, $\mathcal{A}^\mathcal{M}$ is $(c\varepsilon, \delta)$ -DP.

Utility: For any input d , $\Pr[\text{val}(\mathcal{A}^\mathcal{M}(d)) \geq \text{Median}(\text{val}(\mathcal{M}(d)))] \geq 1 - \gamma$.

Runtime: \mathcal{A} on input d makes at most T calls to \mathcal{M} on input d in expectation for some $T \geq e^\varepsilon$.

Then for $\delta < \frac{\varepsilon\gamma}{4 \ln 4T}$, $T \geq (8\gamma)^{-\frac{1}{4c}}/4$, or equivalently, $c \geq \frac{\ln \frac{1}{8\gamma}}{2 \ln 4T}$.

Proof. With some foresight, we set $\Delta = \lceil \frac{\ln 4T}{\varepsilon} \rceil$, and set $q = e^{-\varepsilon\Delta} \leq \frac{1}{4T}$. Consider datasets in $\{0, 1\}^\Delta$ where $d_0 = 0^\Delta$, and $d_1 = 1^\Delta$. Now we define a mechanism \mathcal{M} such that

$$\mathcal{M}(d_1) = \begin{cases} (r, 1) & \text{w.p. } 1 - q \\ (r', 0) & \text{w.p. } q \end{cases} \quad \text{and} \quad \mathcal{M}(d_0) = \begin{cases} (r, 1) & \text{w.p. } q \\ (r', 0) & \text{w.p. } 1 - q \end{cases}.$$

Here r and r' are two arbitrary distinct elements of R . Note that this specification on $\{d_0, d_1\}$ satisfies ε -DP as $\frac{1-q}{q} \leq \frac{1}{q} = e^{\varepsilon|d_0-d_1|_H}$. By the extension lemma (Proposition 2), this partial mechanism can be extended to a 2ε -DP mechanism on $\{0, 1\}^\Delta$.

Consider a run of $\mathcal{A}^\mathcal{M}(d_0)$. Let E be the event that $\mathcal{A}^\mathcal{M}$ on input d_0 makes at most $2T$ calls to $\mathcal{M}(d_0)$. By Markov's inequality, $\Pr[E] \geq \frac{1}{2}$. Conditioned on this event E , the probability that $(r, 1)$ is the output of *any* of the at most $2T$ runs of $\mathcal{M}(d_0)$ is at most $2qT \leq \frac{1}{2}$. Since $\mathcal{A}^\mathcal{M}$ has never seen a $(r, 1)$ output in this case, it follows that $\Pr[\mathcal{A}^\mathcal{M}(d_0) = (r, 1) \mid E] \leq \frac{1}{2}$, so that $\Pr[\mathcal{A}^\mathcal{M}(d_0) = (r', 0) \mid E] \geq \frac{1}{2}$. It follows that $\Pr[\mathcal{A}^\mathcal{M}(d_0) = (r', 0)] \geq \frac{1}{4}$.

As \mathcal{M} is 2ε -DP, the privacy property of $\mathcal{A}^\mathcal{M}$ implies that it is $(2c\varepsilon, \delta)$ -DP. By Proposition 5, it follows that for $\delta < e^{-2c\varepsilon\Delta}/8\Delta$,

$$\Pr[\mathcal{A}^\mathcal{M}(d_1) = (r', 0)] \geq e^{-2c\varepsilon\Delta}/8.$$

By the utility property, the left hand side is at most γ . Since $e^{2c\varepsilon\Delta} \leq e^{4c \ln 4T}$, it follows that $\gamma \geq \frac{1}{8(4T)^{4c}}$. Rearranging, we get the claimed result. ◀

By using Proposition 6 in lieu of Proposition 5, we get a similar result for Renyi DP. We omit the straightforward proof, and similar corollaries of Theorem 9 and Theorem 10.

► **Corollary 8.** *In Theorem 7, if we replace the Privacy condition by*

Privacy (RDP): For any ε -DP mechanism $\mathcal{M} : D^n \rightarrow (R \times \mathbb{R})$, $\mathcal{A}^\mathcal{M}$ is $(\alpha, c\varepsilon)$ -RDP.

Then if $\alpha \leq 1 + \lceil \frac{\ln T}{\varepsilon} \rceil$, $T \geq (44\gamma)^{-\frac{1}{2ec}}/4$, or equivalently, $c \geq \frac{\ln \frac{1}{44\gamma}}{2e \ln 4T}$.

3.1 Allowing more general meta-algorithms

One of the assumptions in Theorem 7 is that when $\mathcal{A}^\mathcal{M}$ is run on input d , its oracle calls to \mathcal{M} are also on input d . A more general oracle algorithm may, given input d , run \mathcal{M} on additional inputs derived from d . We next show that the lower bound continues to hold, up to constants. At a high level, the lower bound in Theorem 7 comes from the inability of \mathcal{A} on input d_0 to see any sample that is unlikely in $\mathcal{M}(d_0)$ but likely in $\mathcal{M}(d_1)$. We show how to *embed* such an instance in a mechanism over a slightly larger dataset, in a way where on input $\mathcal{M}(d_0)$, finding an output that is good for $\mathcal{M}(d_1)$ is still difficult.

► **Theorem 9.** *Let \mathcal{A} be an oracle algorithm that satisfies the following properties:*

Privacy: For any ε -DP mechanism $\mathcal{M} : D^n \rightarrow (R \times \mathbb{R})$, $\mathcal{A}^\mathcal{M}$ is $(c\varepsilon, \delta)$ -DP.

Utility: For any input d , $\Pr[\text{val}(\mathcal{A}^\mathcal{M}(d)) \geq \text{Median}(\text{val}(\mathcal{M}(d)))] \geq 1 - \gamma$.

Runtime: \mathcal{A} on input d makes at most T calls to \mathcal{M} (on any inputs) in expectation for some $T \geq e^\varepsilon$.

Then for $\varepsilon \leq 3$, $\delta < \frac{\varepsilon\gamma}{40 \ln 8T}$, $T \geq (8\gamma)^{-\frac{1}{40c}}/8$, or equivalently, $c \geq \frac{\ln \frac{1}{8\gamma}}{40 \ln 8T}$.

Proof. With some foresight, we set $\Delta = \lceil \frac{\ln 8T}{\varepsilon} \rceil$, and set $q = e^{-\varepsilon\Delta} \leq \frac{1}{8T}$. Fix a vector $v \in \{0, 1\}^{10\Delta}$, and let $B_v = \{u : |u - v|_H \leq \Delta\}$ be the radius Δ Hamming ball centered at v . We now define a mechanism \mathcal{M}_v such that:

$$\mathcal{M}_v(v) = \begin{cases} (r, 1) & \text{w.p. } 1 - q \\ (r', 0) & \text{w.p. } q \end{cases} \quad \text{and} \quad \forall u \notin B_v : \mathcal{M}_v(u) = \begin{cases} (r, 1) & \text{w.p. } q \\ (r', 0) & \text{w.p. } 1 - q \end{cases}.$$

As before, r and r' are two arbitrary distinct elements of R . Note that this specification on $\{v\} \cup B_v^c$ satisfies ε -DP as $\frac{1-q}{q} \leq \frac{1}{q} \leq e^{\varepsilon|u-v|_H}$ for $u \notin B_v$. By the extension lemma (Proposition 2), this partial mechanism can be extended to a 2ε -DP mechanism on $\{0, 1\}^{10\Delta}$.

Consider a run of $\mathcal{A}^{\mathcal{M}_v}(d_0)$ where $d_0 = 0^{10\Delta}$. Let E be the event that $\mathcal{A}^{\mathcal{M}_v}$ on input d_0 makes at most $2T$ calls to \mathcal{M} on any inputs. By Markov's inequality, $\Pr[E] \geq \frac{1}{2}$. For v chosen uniformly at random, the probability that a single dataset d chosen by $\mathcal{A}^{\mathcal{M}_v}(d_0)$ lies in B_v is given by

$$\frac{|B_v|}{2^{10\Delta}} = \frac{\binom{10\Delta}{\Delta}}{2^{10\Delta}} \leq \left(\frac{10e}{1024}\right)^\Delta \leq e^{-3\Delta} \leq \frac{1}{8T}.$$

Thus the likelihood of seeing an $(r, 1)$ on any given call to $\mathcal{M}_v(d)$, when v is chosen at random is

$$\begin{aligned} \Pr[\mathcal{M}_v(d) = (r, 1)] &\leq \Pr[d \in B_v] + \Pr[\mathcal{M}_v(d) = (r, 1) \mid d \notin B_v] \\ &\leq \frac{1}{8T} + q \leq \frac{1}{4T}. \end{aligned}$$

Here and in the rest of the proof, the probability is taken over both the choice of v and the randomness of $\mathcal{A}^{\mathcal{M}_v}$ and \mathcal{M}_v . Conditioned on the event E , the probability that $(r, 1)$ is the output of *any* of the at most $2T$ runs of \mathcal{M} is at most $\frac{2T}{4T} \leq \frac{1}{2}$. Since $\mathcal{A}^\mathcal{M}$ has never seen an $(r, 1)$ output in this case, it follows that $\Pr[\mathcal{A}^{\mathcal{M}_v}(d_0) = (r, 1) \mid E] \leq \frac{1}{2}$, so that $\Pr[\mathcal{A}^\mathcal{M}(d_0) = (r', 0) \mid E] \geq \frac{1}{2}$. It follows that $\Pr[\mathcal{A}^\mathcal{M}(d_0) = (r', 0)] \geq \frac{1}{4}$.

The rest of the proof is now essentially identical to the earlier proof. As \mathcal{M}_v is 2ε -DP, the privacy property of $\mathcal{A}^{\mathcal{M}_v}$ implies that it is $(2c\varepsilon, \delta)$ -DP. By Proposition 5, it follows that for $\delta < e^{-20c\varepsilon\Delta}/80\Delta$,

$$\Pr[\mathcal{A}^{\mathcal{M}}(d_1) = (r', 0)] \geq e^{-20c\varepsilon\Delta}/8.$$

By the utility property, the left hand side is at most γ . Since $e^{20c\varepsilon\Delta} \leq e^{40c \ln 8T}$, it follows that $\gamma \geq \frac{1}{8(8T)^{40c}}$. Rearranging, we get the claimed result. \blacktriangleleft

3.2 Lower Bounds for Hyperparameter Tuning

We next show that the same essential arguments can be used to show a lower bound for private hyperparameter tuning.

► **Theorem 10.** *Let \mathcal{A} be an oracle algorithm that satisfies the following properties:*

Privacy: For any ε -DP mechanism $\mathcal{M} : D^n \times [K] \rightarrow (R \times \mathbb{R})$, $\mathcal{A}^{\mathcal{M}} : D^n \rightarrow (R \times \mathbb{R})$ is $(c\varepsilon, \delta)$ -DP.

Utility: For any input d , $\Pr[\text{val}(\mathcal{A}^{\mathcal{M}}(d)) \geq \max_{k \in [K]} \text{Median}(\text{val}(\mathcal{M}(d, k)))] \geq 1 - \gamma$.

Runtime: \mathcal{A} on input d makes at most TK calls to \mathcal{M} (on any inputs) in expectation for some $T \geq e^\varepsilon$.

Then for $\varepsilon \leq 3$, $\delta < \frac{\varepsilon\gamma}{40 \ln 8T}$, $T \geq (8\gamma)^{-\frac{1}{40c}}/8$, or equivalently, $c \geq \frac{\ln \frac{1}{8\gamma}}{40 \ln 8T}$.

Proof. We set $\Delta = \lceil \frac{\ln 8T}{\varepsilon} \rceil$, and set $q = e^{-\varepsilon\Delta} \leq \frac{1}{8T}$. For a vector $v \in \{0, 1\}^{10\Delta}$, and let $B_v = \{u : |u - v|_H \leq \Delta\}$ be the radius Δ Hamming ball centered at v . For a parameter setting $k \in [K]$, we now define a mechanism $\mathcal{M}_{v,k}$ such that:

$$\begin{aligned} \mathcal{M}_{v,k}(v, k) &= \begin{cases} (r, 1) & \text{w.p. } 1 - q \\ (r', 0) & \text{w.p. } q \end{cases}, \\ \forall u \notin B_v : \mathcal{M}_{v,k}(u, k) &= \begin{cases} (r, 1) & \text{w.p. } q \\ (r', 0) & \text{w.p. } 1 - q \end{cases} \\ \text{and } \forall j \neq k, \forall u \in \{0, 1\}^{10\Delta} : \mathcal{M}_{v,k}(u, j) &= (r', 0) \text{ w.p. } 1. \end{aligned}$$

Here, r and r' are two arbitrary distinct elements of R . By the extension lemma (Proposition 2), for each $j \in [K]$, the partial mechanism $\mathcal{M}_{v,k}(\cdot, j)$ can be extended to a 2ε -DP mechanism on $\{0, 1\}^{10\Delta}$.

Consider a run of $\mathcal{A}^{\mathcal{M}_{v,k}}(d_0)$ where $d_0 = 0^{10\Delta}$. Let E be the event that $\mathcal{A}^{\mathcal{M}_{v,k}}$ on input d_0 makes at most $2TK$ calls to \mathcal{M} on any inputs. By Markov's inequality, $\Pr[E] \geq \frac{1}{2}$. Each call to \mathcal{M} made by $\mathcal{A}^{\mathcal{M}_{v,k}}(d_0)$ consists of a single dataset d and a single $j \in [K]$. When v, k are chosen uniformly at random, the probability that $d \in B_v$ is

$$\frac{|B_v|}{2^{10\Delta}} = \frac{\binom{10\Delta}{\Delta}}{2^{10\Delta}} \leq \left(\frac{10e}{1024}\right)^\Delta \leq e^{-3\Delta} \leq \frac{1}{8T}.$$

If the j chosen by $\mathcal{A}^{\mathcal{M}_{v,k}}(d_0)$ is different from k , the likelihood of seeing an $(r, 1)$ is zero. Thus the likelihood of seeing an $(r, 1)$ on a given call to \mathcal{M} is

$$\begin{aligned} \Pr[\mathcal{M}_{v,k}(d, j) = (r, 1)] &\leq \Pr[j = k \wedge d \in B_v] \\ &\quad + \Pr[j = k] \cdot \Pr[\mathcal{M}_v(d) = (r, 1) \mid j = k \wedge d \notin B_v] \\ &\leq \frac{1}{8TK} + \frac{q}{K} \leq \frac{1}{4TK}. \end{aligned}$$

Conditioned on the event E , the probability that $(r, 1)$ is the output of *any* of the at most $2TK$ runs of \mathcal{M} is at most $\frac{2TK}{4TK} \leq \frac{1}{2}$. Since $\mathcal{A}^{\mathcal{M}}$ has never seen a $(r, 1)$ output in this case, it follows that $\Pr[\mathcal{A}^{\mathcal{M}_{v,k}}(d_0) = (r, 1) \mid E] \leq \frac{1}{2}$, so that $\Pr[\mathcal{A}^{\mathcal{M}}(d_0) = (r', 0) \mid E] \geq \frac{1}{2}$. It follows that $\Pr[\mathcal{A}^{\mathcal{M}}(d_0) = (r', 0)] \geq \frac{1}{4}$.

The rest of the proof is essentially identical. As $\mathcal{M}_{v,k}(\cdot, j)$ is 2ε -DP for each j , the privacy property of $\mathcal{A}^{\mathcal{M}_{v,k}}$ implies that it is $(2c\varepsilon, \delta)$ -DP. By Proposition 5, it follows that for small enough δ .

$$\Pr[\mathcal{A}^{\mathcal{M}_{v,k}}(v) = (r', 0)] \geq e^{-20c\varepsilon\Delta}/8.$$

By the utility property, the left hand side is at most γ . Since $e^{20c\varepsilon\Delta} \leq e^{40c \ln 8T}$, it follows that $\gamma \geq \frac{1}{8(8T)^{40c}}$. Rearranging, we get the claimed result. \blacktriangleleft

4 Upper Bounds Trade-offs

As mentioned earlier, two extreme points on the trade-off between the privacy overhead and the run time overhead are known. We show next the trade-off that is achievable by a simple combination of the basic approaches. We consider the goal of matching the median score, that is the common case for private repetition (labeled *Repetition* below), as well as the goal of matching the $(1 - \frac{1}{K})$ th quantile, which is the case for metaselection and hyperparameter tuning (labeled *Metaselection* below). We first state the results one gets from simple repetition.

► **Theorem 11.** *Suppose that $\mathcal{M} : D^n \rightarrow (R \times \mathbb{R})$ is (ε, δ) -DP. Then for an integer T , the algorithm \mathcal{A}_{\max}^T that on input d runs $\mathcal{M}(d)$ T times and releases the output with the highest quality score satisfies*

- *Privacy:* $\mathcal{A}_{\max}^T : D^n \rightarrow (R \times \mathbb{R})$ is $(T\varepsilon, T\delta)$ -DP, as well as $(T\varepsilon^2 + \varepsilon\sqrt{2T \ln \frac{1}{\delta'}}, T\delta + \delta')$ -DP, for any $\delta' \in (0, 1)$.
- *Runtime:* \mathcal{A} on input d makes exactly T calls to \mathcal{M} .
- *Utility (Repetition):* For any input d , if $T \geq \log_2 \gamma$, then

$$\Pr[\text{val}(\mathcal{A}_{\max}^T(d)) \geq \text{Median}(\text{val}(\mathcal{M}(d)))] \geq 1 - \gamma.$$

- *Utility (Metaselection):* For any input d , and $K > 1$, if $T \geq K \ln \gamma$, then

$$\Pr[\text{val}(\mathcal{A}_{\max}^T(d)) \geq q_{1-\frac{1}{K}}(\mathcal{M}(d))] \geq 1 - \gamma,$$

where the quantile $q_{1-\frac{1}{K}}$ is such that $\Pr[\mathcal{M}(d) \geq q_{1-\frac{1}{K}}] \geq \frac{1}{K}$.

Proof. The privacy follows from simple composition and advanced composition applied and post-processing. The utility and run time analyses are straightforward. \blacktriangleleft

The following result was shown by Liu and Talwar [20].

► **Theorem 12.** *Suppose that $\mathcal{M} : D^n \rightarrow (R \times \mathbb{R})$ is ε -DP and $\gamma > 0$. Then the algorithm $\mathcal{A}_{LT,\gamma}$ satisfies*

- *Privacy:* $\mathcal{A}_{LT,\gamma}^{\mathcal{M}} : D^n \rightarrow (R \times \mathbb{R})$ is 3ε -DP.
- *Runtime:* $\mathcal{A}_{LT,\gamma}^{\mathcal{M}}$ on input d makes in expectation $O(\frac{1}{\gamma})$ calls to \mathcal{M} .
- *Utility (Repetition):* For any input d , $\Pr[\text{val}(\mathcal{A}_{LT,\gamma}^{\mathcal{M}}(d)) \geq \text{Median}(\text{val}(\mathcal{M}(d)))] \geq 1 - \gamma$.

- *Utility (Metaselection):* For any input d , and $K > 1$, the algorithm $\mathcal{A}_{LT,\gamma}^{\mathcal{M}}$ satisfies $\Pr[\text{val}(\mathcal{A}_{LT,\gamma}^{\mathcal{M}}(d)) \geq q_{1-\frac{1}{K}}(\mathcal{M}(d))] \geq 1 - K\gamma$, where the quantile $q_{1-\frac{1}{K}}$ is such that $\Pr[\mathcal{M}(d) \geq q_{1-\frac{1}{K}}] \geq \frac{1}{K}$.

These two algorithms can be combined, by using Theorem 12 to boost the success probability to $1 - \gamma^{\frac{1}{c}}$, and then repeating that algorithm similarly to Theorem 11 to further boost the success probability to $(1 - \gamma)$.

► **Theorem 13.** Suppose that $\mathcal{M} : D^n \rightarrow (R \times \mathbb{R})$ is ε -DP and $c > 1$ be an integer. Then the algorithm \mathcal{A} that runs \mathcal{A}_{\max}^T (for $T = c$) with an $\mathcal{A}_{LT,\gamma^{\frac{1}{c}}}^{\mathcal{M}}$ oracle satisfies

- *Privacy:* $\mathcal{A} : D^n \rightarrow (R \times \mathbb{R})$ is $3c\varepsilon$ -DP.
- *Runtime:* \mathcal{A} on input d makes in expectation $O(\frac{c}{\gamma^{\frac{1}{c}}})$ calls to \mathcal{M} .
- *Utility:* For any input d , $\Pr[\text{val}(\mathcal{A}(d)) \geq \text{Median}(\text{val}(\mathcal{M}(d)))] \geq 1 - \gamma$.

Proof. By Theorem 12, $\mathcal{A}_{LT,\gamma^{\frac{1}{c}}}^{\mathcal{M}}$ is 3ε -DP. Theorem 11 now implies that \mathcal{A} is $3c\varepsilon$ -DP. For the chosen parameters, the utility bound in Theorem 12 implies that $\Pr[\text{val}(\mathcal{A}_{LT,\gamma^{\frac{1}{c}}}^{\mathcal{M}}(d)) \geq \text{Median}(\text{val}(\mathcal{M}(d)))] \geq 1 - \gamma^{\frac{1}{c}}$. Since this algorithm is repeated c times in \mathcal{A} , the failure probability gets reduced to γ . Finally, the run time bound follows by combining the run time results in Theorem 12 and Theorem 11. ◀

We note that for $c < \log \gamma^{-1} / \log \log \gamma^{-1}$, $c^c < \gamma^{-1}$ so that this bound of $O(\frac{c}{\gamma^{\frac{1}{c}}})$ is $O(\gamma^{-2/c})$. Thus it matches the lower bound in Theorem 9 up to constant factors in c .

An identical argument yields the following result for metaselection.

► **Theorem 14.** Suppose that $\mathcal{M} : D^n \rightarrow (R \times \mathbb{R})$ is ε -DP and let $c > 1$ be an integer. Then the algorithm \mathcal{A} that runs \mathcal{A}_{\max}^T (for $t = c$) with an $\mathcal{A}_{LT,\gamma^{\frac{1}{c}}/K}^{\mathcal{M}}$ oracle satisfies

- *Privacy:* $\mathcal{A} : D^n \rightarrow (R \times \mathbb{R})$ is $3c\varepsilon$ -DP.
- *Runtime:* \mathcal{A} on input d makes in expectation $O(\frac{Kc}{\gamma^{\frac{1}{c}}})$ calls to \mathcal{M} .
- *Utility:* For any input d , $\Pr[\text{val}(\mathcal{A}(d)) \geq q_{1-\frac{1}{K}}(\mathcal{M}(d))] \geq 1 - \gamma$, where the quantile $q_{1-\frac{1}{K}}$ is such that $\Pr[\mathcal{M}(d) \geq q_{1-\frac{1}{K}}] \geq \frac{1}{K}$.

Proof. By Theorem 12, $\mathcal{A}_{LT,\gamma^{\frac{1}{c}}/K}^{\mathcal{M}}$ is 3ε -DP. Theorem 11 now implies that \mathcal{A} is $3c\varepsilon$ -DP. For the chosen parameters, the utility bound in Theorem 12 implies that $\Pr[\text{val}(\mathcal{A}_{LT,\gamma^{\frac{1}{c}}/K}^{\mathcal{M}}(d)) \geq q_{1-\frac{1}{K}}(\mathcal{M}(d))] \geq 1 - K \cdot (\gamma^{\frac{1}{c}}/K) = 1 - \gamma^{\frac{1}{c}}$. Since this algorithm is repeated c times in \mathcal{A} , the failure probability gets reduced to γ . Finally, the run time bound follows by combining the run time results in Theorem 12 and Theorem 11. ◀

This result improves on \mathcal{A}_{\max}^T in terms of privacy cost and gives the points plotted on Figure 1(right). As in the case of repetition, this result is tight up to constants in c all the way up to $c = \log \gamma^{-1} / \log \log \gamma^{-1}$.

5 Conclusions

Differentially private repetition is a fundamental problem in the design of differentially private algorithms, and private hyperparameter tuning is of great practical importance. Our work shows new trade-offs between privacy overhead and computational overhead, and our main result is a lower bound showing that there is no general algorithm that can do significantly better. Indeed we show that for constant privacy overhead, the computational overhead must be polynomial in $\frac{1}{\gamma}$ for some private algorithms.

It is natural to ask if there are reasonable restrictions one can place on the inputs or the private algorithms of interest, for which better repetition theorems and/or hyperparameter tuning algorithms are possible. Such beyond-worst-case results are not uncommon in differential privacy [23, 8, 31, 2]. There are some assumptions that hyperparameter tuning algorithms either implicitly [30] or explicitly [16] make in the non-private setting and one may ask if such assumptions help with better trade-offs for private hyperparameter search. Even absent computational constraints, the $2\times$ or $3\times$ privacy cost overhead in \mathcal{A}_{LT} can be large. Once again, while it is unavoidable in the worst-case, one may hope to design algorithms that do better for non-worst-case instances. We leave these important questions for future work.

References

- 1 Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Conference on Computer and Communications Security*, pages 308–318, 2016. doi:10.1145/2976749.2978318.
- 2 Hilal Asi and John C Duchi. Instance-optimality in differential privacy via approximate inverse sensitivity mechanisms. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14106–14117. Curran Associates, Inc., 2020. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/a267f936e54d7c10a2bb70dbe6ad7a89-Paper.pdf.
- 3 Raef Bassily, Kobbi Nissim, Uri Stemmer, and Abhradeep Thakurta. Practical locally private heavy hitters. *Journal of Machine Learning Research*, 21(16):1–42, 2020. URL: <http://jmlr.org/papers/v21/18-786.html>.
- 4 Christian Borgs, Jennifer Chayes, Adam Smith, and Ilias Zadik. Revealing network structure, confidentially: Improved rates for node-private graphon estimation. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 533–543, 2018. doi:10.1109/FOCS.2018.00057.
- 5 Gavin Brown, Jonathan Hayase, Samuel Hopkins, Weihao Kong, Xiyang Liu, Sewoong Oh, Juan C Perdomo, and Adam Smith. Insufficient statistics perturbation: Stable estimators for private least squares extended abstract. In Shipra Agrawal and Aaron Roth, editors, *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 750–751. PMLR, 30 June–03 July 2024. URL: <https://proceedings.mlr.press/v247/brown24b.html>.
- 6 Edith Cohen and Xin Lyu. The target-charging technique for privacy analysis across interactive computations. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 62139–62168. Curran Associates, Inc., 2023. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/c3fe2a07ec47b89c50e89706d2e23358-Paper-Conference.pdf.
- 7 Michael Dinitz, Satyen Kale, Silvio Lattanzi, and Sergei Vassilvitskii. Almost tight bounds for differentially private densest subgraph. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 2908–2950. SIAM, 2025. doi:10.1137/1.9781611978322.94.
- 8 Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, STOC '09*, pages 371–380, New York, NY, USA, 2009. Association for Computing Machinery. doi:10.1145/1536414.1536466.
- 9 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. *Journal of Privacy and Confidentiality*, 7(3):17–51, 2017. doi:10.29012/jpc.v7i3.405.
- 10 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 Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, pages 381–390, New York, NY, USA, 2009. Association for Computing Machinery. doi:10.1145/1536414.1536467.
- 11 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3 & 4):211–407, 2014. doi:10.1561/04000000042.
 - 12 Vitaly Feldman, Audra McMillan, Satchit Sivakumar, and Kunal Talwar. Instance-optimal private density estimation in the wasserstein distance. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 90061–90131. Curran Associates, Inc., 2024. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/a406c9f8eb70032a21110a4d86735ab9-Paper-Conference.pdf.
 - 13 Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, November 1995. doi:10.1145/227683.227684.
 - 14 Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. Differentially private combinatorial optimization. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 1106–1125, USA, 2010. Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611973075.90.
 - 15 Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 61–70, 2010. doi:10.1109/FOCS.2010.85.
 - 16 Elad Hazan, Adam Klivans, and Yang Yuan. Hyperparameter optimization: a spectral approach. In *International Conference on Learning Representations*, 2018. URL: <https://openreview.net/forum?id=H1zriGeCZ>.
 - 17 Shlomi Hod and Ran Canetti. Differentially Private Release of Israel’s National Registry of Live Births . In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 100–100, Los Alamitos, CA, USA, May 2025. IEEE Computer Society. doi:10.1109/SP61157.2025.00101.
 - 18 Thomas Holenstein. Parallel repetition: simplifications and the no-signaling case. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 411–419, New York, NY, USA, 2007. Association for Computing Machinery. doi:10.1145/1250790.1250852.
 - 19 Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43:439–561, 2006.
 - 20 Jingcheng Liu and Kunal Talwar. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 298–309, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3313276.3316377.
 - 21 Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 94–103, USA, 2007. IEEE Computer Society. doi:10.1109/FOCS.2007.41.
 - 22 Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275, 2017. doi:10.1109/CSF.2017.11.
 - 23 Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 75–84, New York, NY, USA, 2007. Association for Computing Machinery. doi:10.1145/1250790.1250803.
 - 24 Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations*, 2017. URL: <https://openreview.net/forum?id=HkwoSDPgg>.

- 25 Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Ulfar Erlingsson. Scalable private learning with PATE. In *International Conference on Learning Representations*, 2018. URL: <https://openreview.net/forum?id=rkZB1XbRZ>.
- 26 Nicolas Papernot and Thomas Steinke. Hyperparameter tuning with renyi differential privacy. In *International Conference on Learning Representations*, 2022. URL: <https://openreview.net/forum?id=-70L81pp9DF>.
- 27 Ran Raz. A parallel repetition theorem. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '95, pages 447–456, New York, NY, USA, 1995. Association for Computing Machinery. doi:10.1145/225058.225181.
- 28 Lucas Rosenblatt, Xiaoyan Liu, Samira Pouyanfar, Eduardo de Leon, Anuj Desai, and Joshua Allen. Differentially private synthetic data: Applied evaluations and enhancements. *arXiv preprint arXiv:2011.05537*, 2020. arXiv:2011.05537.
- 29 Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, STOC '10, pages 765–774, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1806689.1806794.
- 30 Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.
- 31 Abhradeep Guha Thakurta and Adam Smith. Differentially private feature selection via stability arguments, and the robustness of the lasso. In Shai Shalev-Shwartz and Ingo Steinwart, editors, *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pages 819–850, Princeton, NJ, USA, 12–14 June 2013. PMLR. URL: <https://proceedings.mlr.press/v30/Guha13.html>.
- 32 Salil Vadhan. *The Complexity of Differential Privacy*, pages 347–450. Springer, Yehuda Lindell, ed., 2017. doi:10.1007/978-3-319-57048-8_7.
- 33 Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/254ed7d2de3b23ab10936522dd547b78-Paper.pdf.
- 34 Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019. URL: <https://openreview.net/forum?id=S1zk9iRqF7>.