DIFFERENTIAL PRIVACY

BEYOND THE CENTRAL MODEL

Matthew Joseph

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2020

Supervisor of Dissertation

Aaron Roth
Associate Professor
Computer and Information Science

Graduate Group Chairperson

Rajeev Alur, Professor
Computer and Information Science

Dissertation Committee
Michael Kearns, Professor and National Center Chair, Computer and Information Science
Anindya De, Assistant Professor, Computer and Information Science
Sampath Kannan, Henry Salvatori Professor, Computer and Information Science
Toniann Pitassi, Professor and Bell Canada Chair in Information Systems, Department of Computer Science at University of Toronto

Acknowledgements

Many people have helped make this dissertation possible. I first thank Aaron Roth for being an outstanding advisor. Spending five years learning from and working with Aaron has been one of the great bits of luck in my life.

I thank the rest of my Dissertation Committee: Anindya De, Sampath Kannan, Michael Kearns, and Toniann Pitassi. I especially thank Michael for his additional mentorship.

I thank Jamie Morgenstern for providing all the one-on-one research meetings, helpful advice, and vegetarian cooking information that a first-year graduate student could want. I thank Bo Waggoner for teaching me the basics of differential privacy and how to throw a frisbee. I thank Jieming Mao for teaching me about probability, information theory, programming problems, and other things he learned in primary school.

I thank both Jana Kulkarni and Kareem Amin for hosting me for fun and productive summer internships at, respectively, Microsoft Research and Google.

I thank John Boller, Risi Kondor, and Maryanthe Malliaris for encouraging me to pursue graduate school during my time at UChicago. I particularly thank Risi for patiently working with such an ill-prepared undergraduate as a research collaborator.

I thank those of my collaborators not mentioned above: Shahin Jabbari, Seth Neel, Jonathan Ullman, and Zhiwei Steven Wu.

More personally, I thank my parents and sister for their continuous support. Many other people have helped me, but in the spirit of this dissertation, I leave these hard-for-an-adversary-to-infer individuals unnamed. However, I must thank Marcella.

ABSTRACT

DIFFERENTIAL PRIVACY
BEYOND THE CENTRAL MODEL

Matthew Joseph

Aaron Roth

A differentially private algorithm adds randomness to its computations to ensure that its output reveals little about its input. This careful decoupling of output and input provides privacy for users that contribute input data, but the nature of this privacy depends on the model of differential privacy used. In the most common model, a differentially private algorithm receives a raw database and must produce a differentially private output. This privacy guarantee requires several assumptions. There must exist a secure way of sending the data to the algorithm; the algorithm must maintain a secure state while carrying out its computations; and data contributors must trust the algorithm operator to responsibly steward their raw data in the future. When these three assumptions hold, differential privacy offers both meaningful utility and privacy. In this dissertation, we study what is possible when these assumptions fail.

Pan-privacy weakens the first two assumptions and removes the third. *Local* differential privacy removes all three. Unfortunately, this flexibility comes at a cost. Pan-privacy often introduces more random noise, and local differential privacy adds more noise still. This reduces utility in the forms of worse accuracy and higher sample complexity. Motivated by this trade-off between privacy and utility, it is important to understand the relative powers of these models. We approach this question in two ways. The first part of this dissertation focuses on *connections* between different models: we show that in some settings, it is possible to convert algorithms in one model to algorithms in another. The second part of this dissertation complements these connections with *separations*: we construct problems where algorithms in different models must obtain different performance guarantees.

Table of Contents

List of Tables

List of Illustrations

Chapter 1

Introduction

Differential privacy [34] is a mathematical guarantee that forces an algorithm's output to be relatively insensitive to small changes in its input. This obscures the presence or absence of any one data contributor. Rigorous privacy guarantees and practical solutions for a variety of problems have in turn driven adoption of differential privacy by industry [9, 14, 30, 43], government [41], and researchers [52, 54].

In this work, we consider three models of differential privacy: central differential privacy, pan-privacy, and local differential privacy[1]. *Central* differential privacy [34] grants an algorithm free access to a raw database and only requires the algorithm to produce a differentially private output. Of the three models, central differential privacy always offers the best utility — any algorithm in the other two models can always be simulated by a centrally differentially private algorithm — but this flexibility comes at the cost of three assumptions.

First, the algorithm's "central" access to the raw database requires users to trust the algorithm's data collection process. Second, the algorithm's internal state during computation is unconstrained. As a result, intrusion on the algorithm's internal state mid-computation may reveal arbitrary information about the data. This further requires users to trust the security of the algorithm operator. Third, differential privacy makes no promises about future stewardship of data. A differentially private algorithm may faithfully produce differentially private outputs, but nothing prevents the algorithm operator from using the algorithm's raw data for other purposes in the future. It follows that users must trust the algorithm operator to use their data responsibly even after the computation has finished. In total, central differential privacy offers an algorithm operator the highest utility, but it also requires the most trust from the users providing data.

[1] For brevity, we often refer to these models as simply central, pan-, and local privacy.

A *pan-private* [36] algorithm receives a stream of raw data, one element at a time, and maintains a differentially private internal state while processing that stream. This models an algorithm operator that acquires data over time. Gradual data acquisition weakens the first assumption of central differential privacy. However, pan-privacy does assume that the stream itself is transmitted through a secure channel, and differential privacy is only guaranteed against an adversary who intrudes on the internal state at most once. The differential privacy of the internal state also weakens central differential privacy's second assumption and enables pan-privacy to drop the third assumption completely. This is because differential privacy's post-processing guarantee (see Fact 1) ensures that a pan-private algorithm only preserves a differentially private summary of the data it has seen. This prohibits the storage of any raw data, so users need not trust the algorithm operator to steward data in the future. Pan-privacy thus requires less trust from users, but it also places more restrictions on the algorithm operator.

A locally differentially private [34, 38, 50] algorithm does not see any raw data. Instead, the database remains distributed among users on their devices. The algorithm must learn by interacting with these users in a public yet privacy-preserving way. Because the interaction is public, local privacy requires none of central differential privacy's three assumptions. In local privacy, secure channels are unnecessary, algorithmic state may equivalently be public as well, and there is no raw data that the operator must safeguard in the future. Moreover, because users add randomness to their communications from their own devices, they are in full control of their own privacy and need not trust any other party.

Central, pan-, and local privacy are thus ordered by decreasing user trust. A user who trusts the algorithm operator both today and in the future will accept a central privacy guarantee; a user who only trusts the algorithm operator today will accept a pan-privacy guarantee; and a user who does not trust the algorithm operator at all will only accept a local privacy guarantee.

1.1. Our Contributions

In this dissertation, we study the relationships between these three models and their subclasses. We primarily focus on pan-privacy and local privacy. At a high level, we divide our results into *connections* and *separations*. Connections (Chapters 3 and 4) are transformations between algorithms in different model classes. They provide ways of porting certain kinds of algorithms with one privacy guarantee to algorithms with another. Separations (Chapters 5 through 8) instead provide problems where these transformations are impossible, and the models must obtain different sample complexity guarantees. Informally, we show:

1. Pure pan-privacy against multiple intrusions is equivalent to the local privacy where each user participates at most once ("sequential interaction") [8] (Chapter 3).

2. At a controlled cost, one can convert any pure locally private algorithm into a sequentially interactive equivalent [47] (Chapter 4).

3. The problems of simple hypothesis testing [47] and one-dimensional Gaussian estimation [46] polynomially separate central and local privacy (Chapter 5).

4. By a connection between communication complexity and local privacy, the hidden layers problem exponentially separates fully interactive (each user may participate arbitrarily many times) and sequentially interactive local privacy [48] (Chapter 6).

5. A second lower bound, which does not rely on the aforementioned communication connection and instead uses a direct proof for a similar problem, shows that the cost of the full-sequential conversion of Chapter 4 is tight (Chapter 7).

6. The problem of uniformity testing polynomially separates central, pure pan-, and sequentially interactive local privacy [8] (Chapter 8).

Note that our separations fall into two rough categories. Chapters 5 and 8 give separations

for the *learning* problems of, respectively, simple hypothesis testing and uniformity testing. Chapters 6 and 7 instead use *communication* problems, which — though still valid tasks for locally private protocols — have more artificial structures.

We conclude with some folklore results in pan-privacy and possible directions for future work (Chapter 9).

1.2. Related Work

We begin with basic background information relating the three models, with a focus on pan-privacy and local privacy. More problem-specific related work appears in later sections.

Dwork, Naor, Pitassi, Rothblum, and Yekhanin [36] introduced pan-privacy and gave algorithms for several different counting problems over streams. This original definition of pan-privacy is different than the one we consider here. In particular, their definition promises user-level rather than event-level privacy (for details, see the discussion in Chapter 2.2). This makes their lower bounds weaker and their upper bounds stronger relative to ours. Since we focus on event-level privacy, we shorthand their formulation as "user-level pan-privacy" and ours as "pan-privacy". With this definition, Dwork et al. [36] separated user-level pan-privacy against one and multiple intrusions by showing that estimating the number of distinct elements in a stream is much harder with multiple intrusions. Second, they showed that inner-product counting is easier under user-level pan-privacy than non-interactive local privacy. Mir, Muthukrishnan, Nikolov, and Wright [53] extended these results to new counting problems and dynamic streams permitting element deletions. They also showed that user-level pan-private algorithms cannot additively approximate distinct element count to $o(\sqrt{|X|})$ accuracy for data universe X (in fact, this lower bound extends to our notion of pan-privacy as well). This improved the previous $\Omega\left(\frac{\sqrt{|X|}}{\log(|X|)}\right)$ lower bound given by McGregor, Mironov, Pitassi, Reingold, Talwar, and Vadhan [51] for two-party privacy (a weaker notion of privacy than pan-privacy), which may be viewed as the first separation between central and pan-privacy. Dwork, Naor, Pitassi, and Rothblum [35] also

studied a variant of pan-privacy with the additional constraint of continual observation, which requires the algorithm to provide accurate answers after every stream element. They and Chan, Shi, and Song [25] gave both upper and lower bounds for counting problems under continual observation.

The first result in this dissertation differs from those above by showing that pure pan-privacy against multiple intrusions is equivalent to a different model, sequentially interactive local privacy (Chapter 3). In contrast, Dwork et al. [36] showed that user-level pan-privacy against multiple intrusions is worse for the specific problem of counting distinct elements. We also separate pan-privacy from sequentially interactive local privacy for the problem of uniformity testing (Chapter 8).

We now turn to local privacy. A series of works [34, 38, 50] introduced and formalized the basics of local privacy, and Duchi, Jordan, and Wainwright [33] first defined noninteractive, sequentially interactive, and fully interactive algorithms. Most local privacy lower bounds apply only to the noninteractive or sequentially interactive models, but there are a few exceptions. For the problem of summing n bits, Beimel, Nissim, and Omri [13] showed that $o(\sqrt{n})$ additive accuracy is impossible in a limited number of rounds. Chan, Shi, and Song [26] extended this result to an arbitrary number of rounds. The aforementioned lower bound of Mir et al. [53] also separates central and local privacy for the problem of counting distinct elements. Kasiviswanathan, Lee, Nissim, Raskhodnikova, and Smith [50] constructed an equivalence between local privacy and SQ learning and used it to show that learning d-parity is hard for sequentially interactive protocols with $\text{poly}(d)$ users and fully interactive algorithms using $2^{o(d^{1/3})}$ randomizer calls. They also separated noninteractive and sequentially interactive learning for the similar problem of d-masked parity. Daniely and Feldman [28] gave a similar but more general separation for concept classes with high margin complexity. Most recently, Duchi and Rogers [31] extended the results of Duchi et al. [32] to fully interactive algorithms for problems admitting strong data processing inequalities. These results certify the optimality of certain noninteractive solutions, but

they do not give any examples where noninteractivity is necessarily suboptimal.

The second result in this dissertation shows how to (at a cost) convert fully interactive algorithms into sequentially interactive equivalents (Chapter 4). This differs from the results above, which come closest when showing that interactivity offers no benefits for certain problems. We also separate centrally and locally private hypothesis testing (Chapter 5) and, as a corollary, one-dimensional Gaussian estimation (Chapter 5). Because it applies even to approximate local privacy and incorporates a dependence on ε, this separation is more general than those given above. We also give the first separations between fully and sequentially interactive local privacy (Chapters 6 and 7).

Chapter 2

Preliminaries

This section covers basic definitions for the rest of this work. Where appropriate, specific additional definitions appear in later sections.

2.1. Differential Privacy

A randomized algorithm is *differentially private* if its output distribution is relatively insensitive to small changes in its input. We define this insensitivity at the granularity of replacing a single element in the database. Because the output distribution of differentially private $\mathcal{A}$ is agnostic to the presence or absence of any one datum (as parameterized by ε and δ), an adversary can infer only a carefully prescribed amount of information about any one record from the output of $\mathcal{A}$.

Definition 1 ([34])**.** *Given data universe $\mathcal{X}$ and two databases $D, D' \in \mathcal{X}^n$, D and D' are* neighbors *if they differ in ≤ 1 element. Given algorithm $\mathcal{A} : \mathcal{X}^n \to Y$, $\mathcal{A}$ is* (ε, δ)-differentially private *if for all subsets $S \subset Y$,*

$$\mathbb{P}_{\mathcal{A}}\left[\mathcal{A}(D) \in S\right] \leq e^{\varepsilon}\mathbb{P}_{\mathcal{A}}\left[\mathcal{A}(D') \in S\right] + \delta.$$

When $\delta = 0$, we say $\mathcal{A}$ is ε-differentially private*.*

The case where $\delta > 0$ is also called *approximate* privacy, with the case where $\delta = 0$ called *pure*. Two fundamental lemmas about differential privacy will be useful. References for these results appear in Chapters 2 and 3 of the survey of Dwork and Roth [37]. First, differential privacy is resilient to post-processing: any function applied to a differentially private output inherits the same privacy guarantee.

Fact 1. *For (ε, δ)-differentially private algorithm $\mathcal{A} : \mathcal{X}^n \to Y$ and arbitrary random function $f : Y \to Y'$, $f \circ \mathcal{A}$ is (ε, δ)-differentially private.*

Informally, this is because post-processing cannot "undo" the randomness added in the computation of $\mathcal{A}(D)$[1]. Second, differential privacy guarantees "add up" when composed.

Fact 2. *For $(\varepsilon_1, \delta_1)$-differentially private $\mathcal{A}_1$ and $(\varepsilon_2, \delta_2)$-differentially private $\mathcal{A}_2$, $\mathcal{A}_3$ defined by $\mathcal{A}_3(D) = (\mathcal{A}_1(D), \mathcal{A}_2(D))$ is $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-differentially private.*

Together, post-processing and composition guarantees enable us to handle differentially private algorithms in a modular way: we can cut and paste them together in a pipeline and obtain a privacy guarantee for the pipeline as a whole by appropriately adding up individual privacy guarantees.

Note that Definition 1 makes a few assumptions. First, including D as an input to $\mathcal{A}$ means that the algorithm has secure access to raw data. Second, because the constraint applies only to the output of $\mathcal{A}$, it assumes that algorithm's internal state during computation is safe from intrusion. Third, differential privacy does not on its own prevent the operator of $\mathcal{A}$ from using D for some other purpose in the future. For example, the operator could repeatedly compute $\mathcal{A}$ on D and — as the privacy parameters grow through composition — eventually make the privacy guarantee vacuous. We will primarily differentiate pan-privacy and local privacy from central privacy using these three assumptions.

2.2. Pan-Privacy

In the *pan-private* model, the algorithm $\mathcal{A}$ receives the database D as a stream of elements, typically denoted S for clarity. Upon receiving an element S_t, $\mathcal{A}$ updates its internal state, deletes the element, proceeds to the next element in the stream S_{t+1}, and eventually publishes some output when the stream ends. This models a gradual data acquisition process where the database is acquired over time.

Pan-privacy strengthens central differential privacy by requiring privacy of $\mathcal{A}$'s internal state. An adversary that intrudes upon the internal state of $\mathcal{A}$ at any single time t should not be able to determine if $\mathcal{A}$ received stream S_t or neighboring stream S'_t. We formalize

[1] For an alternative view, this is conceptually similar to propagation of measurement error.

this below.

Definition 2 ([36])**.** *Let $\mathcal{X}$ be a data universe, and let $\mathcal{S} = \mathcal{X}^{\mathbb{N}}$ be the set of streams from $\mathcal{X}$. Two streams $S, S' \in \mathcal{S}$ are* neighbors *if there exists index t such S and S' differ only at index t.*

A pan-private algorithm consists of an internal algorithm $\mathcal{A}_{\mathcal{I}}$ and an output algorithm $\mathcal{A}_{\mathcal{O}}$. $\mathcal{A}$ maps streams to internal states by repeated application of $\mathcal{A}_{\mathcal{I}}$, which maps an internal state and element of $\mathcal{X}$ to an internal state, $\mathcal{A}_{\mathcal{I}} : \mathcal{I} \times \mathcal{X} \to \mathcal{I}$. At some time the stream ends and $\mathcal{A}$ publishes a final output $\mathcal{A}_{\mathcal{O}}(i)$ where i is the internal state of $\mathcal{A}$ at the end of the stream. For stream S, let $\mathcal{A}_{\mathcal{I}}(S)$ denote the internal state of $\mathcal{A}$ after processing S, and let $S_{\leq t}$ denote the first t elements of stream S. $\mathcal{A}$ is (ε, δ)-pan-private *if, for any neighboring streams S and S', any time t, any $E \subset \mathcal{I} \times \mathcal{O}$,*

$$\mathbb{P}_{\mathcal{A}}\left[(\mathcal{A}_{\mathcal{I}}(S_{\leq t}), \mathcal{A}_{\mathcal{O}}(\mathcal{A}_{\mathcal{I}}(S))) \in E\right] \leq e^{\varepsilon}\mathbb{P}_{\mathcal{A}}\left[(\mathcal{A}_{\mathcal{I}}(S'_{\leq t}), \mathcal{A}_{\mathcal{O}}(\mathcal{A}_{\mathcal{I}}(S'))) \in E\right] + \delta. \tag{2.1}$$

When $\delta = 0$, we say $\mathcal{A}$ is ε-pan-private.

Pan-privacy thus protects against an adversary that sees any single internal state of $\mathcal{A}$ as well as its final output. The second requirement implies that any pan-private algorithm is also differentially private; the key additional contribution of pan-privacy is the maintenance of the differentially private internal state[2].

We also note that this definition of pan-privacy differs from past definitions [36, 53] by guaranteeing event-level privacy (uncertainty about the presence of any single stream element) whereas previous works guarantee user-level privacy (uncertainty about the presence of any one data universe element). The main reason for this change is that Dwork et al. [36] originally intended pan-privacy for streams where users might contribute many data points. We instead assume that each user contributes at most one data point. This is the

[2]Pan-privacy also assumes that the process of receiving a stream element, updating internal state, and deleting the stream element from memory is atomic. This means that an adversary cannot interrupt the process. Without this assumption, nothing prevents an adversary from seeing a stream element in the clear, and privacy is impossible.

most common assumption in most works on differentially privacy (for example, it is implicit in Definition 1). We therefore shorthand the "event-level" pan-privacy of Definition 2 as simply"pan-privacy".

Recall the three assumptions made for central privacy. By gradually acquiring the raw database D in a stream S, pan-privacy weakens the first assumption of secure access to D in its entirety. Pan-privacy's tolerance of any one intrusion on its internal state also weakens the second assumption of a completely secure internal state. Finally, by the post-processing guarantee of Fact 1, the operator of pan-private $\mathcal{A}$ may only access a differentially private summary of past data after proceeding to the next stream element. Users therefore need not trust the operator after it has moved on to acquiring other data.

Finally, to generalize Definition 2 to $c > 1$ intrusions, we can replace inequality 2.1 with

$$\mathbb{P}_{\mathcal{A}}\left[(\mathcal{A}_{\mathcal{I}}(S_t)_{t=t_1}^{t_c}, \mathcal{A}_{\mathcal{O}}(\mathcal{A}_{\mathcal{I}}(S))) \in E\right] \leq e^{\varepsilon}\mathbb{P}_{\mathcal{A}}\left[(\mathcal{A}_{\mathcal{I}}(S'_t)_{t=t_1}^{t_c}, \mathcal{A}_{\mathcal{O}}(\mathcal{A}_{\mathcal{I}}(S'))) \in E\right] + \delta$$

where $E \subset \mathcal{I}^c \times \mathcal{O}$. This generalization will only be relevant when proving our equivalence between multi-intrusion pan-privacy and sequentially interactive local privacy (Chapter 3).

2.3. Local Differential Privacy

Our final privacy model is *local differential privacy*. Variants on this model date back to the simplest form of Warner's randomized response [62] and more generally the notion of γ-amplification defined by Evfimieski, Gehrke, and Srikant [38]. Dwork et al. [34] also touched on this local restriction, but the first focused study of local differential privacy is the work of Kasiviswanathan et al. [50].

Local differential privacy does not assume any access to raw data or any limit on intrusions into an algorithm's internal state. Moreover, it strictly limits even the algorithm operator's knowledge of the database D. Rather than view $\mathcal{A}$ as an algorithm receiving input and producing output, we view $\mathcal{A}$ as a protocol that coordinates public communication among

users. $\mathcal{A}$ thus has no special information access privileges relative to users.

The primary role of the protocol is to assign *randomizers* to users. A randomizer is a differentially private function of the user's input. By communicating through randomizers, users add randomness to all of their communications, and this randomness ensures privacy. By choosing randomizers carefully, the protocol can still extract useful aggregate information from the users as a whole.

Definition 3. *An* (ε, δ)-randomizer $R : X \to Y$ *is an* (ε, δ)*-differentially private function taking a single data point as input.*

A simple but useful $(\varepsilon, 0)$-randomizer is *randomized response* [62], denoted $\mathsf{RR}(\cdot, \varepsilon)$. Given a bit x and privacy parameter ε, randomized response outputs x with probability $\frac{e^\varepsilon}{e^\varepsilon+1}$ and outputs $1 - x$ with probability $\frac{1}{e^\varepsilon+1}$.

Since the protocol learns from users through randomizer calls, we need to formalize the way we represent this information. The record of users, randomizers, and messages over time is a *transcript*.

Definition 4. *A* transcript π *is a vector of tuples* (i_t, R_t, y_t) *indicating the user queried, randomizer used, and output produced at each time* t*.*

We can therefore define a protocol as a function that assigns randomizers to users based on the transcript so far or halts. We typically refer to locally private algorithms as protocols to emphasize their distributed and cooperative nature.

Definition 5. *Let* S_π *denote the collection of transcripts,* $[n]$ *the collection of users, and* S_R *the collection of randomizers. Then a* protocol $\mathcal{A}$ *is a function* $\mathcal{A} : S_\pi \to [n] \times S_R \cup \{\perp\}$ *that maps transcripts to users and randomizers (or halts (denoted* $\perp$*).*

We now have all the definitions to define a locally differentially private protocol.

Definition 6. *A protocol* $\mathcal{A} : S_\pi \to [n] \times S_R \cup \{\perp\}$ *is* (ε, δ)-locally differentially private *if*

for all neighboring (distributed) databases D and D' and subsets $S \subset S_\pi$,

$$\mathbb{P}\left[T(\mathcal{A}(D)) \in S\right] \leq e^{\varepsilon}\mathbb{P}\left[T(\mathcal{A}(D')) \in S\right] + \delta$$

where T is the random variable for the transcript generated by $\mathcal{A}$. If $\delta = 0$, we say $\mathcal{A}$ is ε-locally differentially private.

Note that while this definition views a protocol $\mathcal{A}$ as a function on the data D, $\mathcal{A}$ may only interact with D by querying randomizer outputs from users holding the data from D. In this sense, $\mathcal{A}$ never "sees" D.

Since $\mathcal{A}$ is a protocol that communicates with users, we can also distinguish between different classes of locally private protocols based on interaction. A *noninteractive* protocol makes all randomizer assignments before seeing any responses.

Definition 7. *A protocol $\mathcal{A}$ is* noninteractive *if, at each round t, as random variables, (i^t, R_t) is conditionally independent of $\Pi_{<t}$ given t.*

Note that some works [4, 6] have studied an even stronger private-coin model of noninteraction. Informally, our public-coin model allows for an additional "half step" of interaction over the private-coin model because the protocol may coordinate randomizer choices across users. In contrast, the private-coin model forbids this. Some of our noninteractive protocols (for example, the simple hypothesis tester of Chapter 5) fit directly into the private-coin model, but in general this distinction makes our upper bounds relatively weaker and our lower bounds relatively stronger.

More generally, a *sequentially interactive* protocol can make randomizer assignments adaptively, but may still only query each user at most once.

Definition 8 ([32])**.** *An algorithm $\mathcal{A}$ is* sequentially interactive *if, at each round t, $i^t \neq i^{t-1}, \ldots, i^1$.*

Most generally, a *fully interactive* protocol may make adaptive randomizer assignments and

query each user arbitrarily many times; its only constraint is the local privacy guarantee of Definition 6.

For any locally private protocol, we refer to the number of users n that it queries as its *sample complexity*. For fully interactive protocols, the total number of rounds — which we denote by T — may greatly exceed n. In contrast, for both noninteractive and sequentially interactive protocols, $T \leq n$. Illustrations of these models appear in Figure 1.

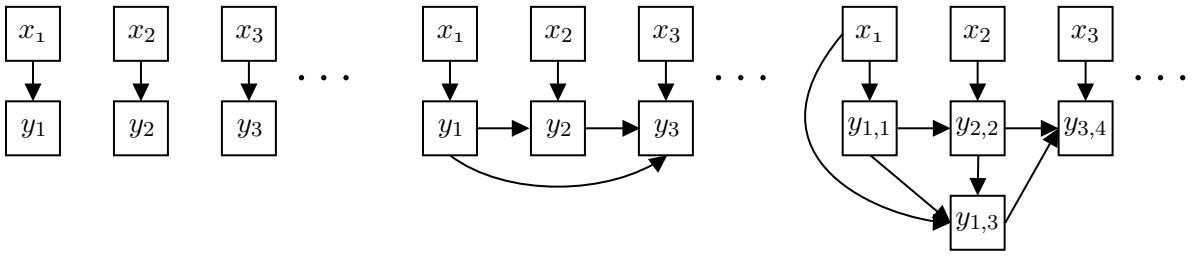

Figure 1: From left to right, examples of noninteractive, sequential, and full interaction. In each illustration, x variables are user data, and y variables are privatized user responses. In the noninteractive model, each privatized response y_i is a function only of the user's data x_i (and their internal randomness). In the sequential model, each y_i is a function of x_i and previous responses $y_1, \ldots, y_{i-1}$. In the full model, each $y_{i,t}$ is a function of x_i and any $y_{i',t'}$ for any $t' < t$.

Chapter 3

Connection: Pan-Privacy and Local Privacy

We now give the first of two connections by connecting pan-privacy and local privacy [8]. Specifically, we show that any algorithm that is (pure) pan-private against multiple intrusions has a sequentially interactive locally private equivalent (Theorem 1). The main idea is that the operator of a pan-private algorithm $\mathcal{A}_{2P}$ cannot know when two intrusions will occur. In particular, if the two intrusions occur at times t and $t+1$ — respectively, immediately after $\mathcal{A}_{2P}$ processes the t^{th} and $(t+1)^{th}$ stream elements — then failure to randomize the internal state between t and $t+1$ may reveal element s_{t+1}. The operator must therefore re-randomize the state at *every* time step.

We briefly sketch the proof of Theorem 1. First, we show that any $\mathcal{A}_{2P}$ that is ε-pan-private against two intrusions can be modified into an algorithm $\mathcal{A}_{1P}$ that maintains all of its internal states thus far and still remain ε-pan-private against *one* intrusion (Lemma 1). Because this single intrusion may come at the end of the stream, the complete list of internal states during the stream must be an ε-differentially private function of the stream. We can therefore simulate this procedure in the sequentially interactive local model and have the transcript generate this complete list of internal states (Lemma 2).

In the other direction, we convert any sequentially interactive ε-locally private protocol $\mathcal{A}_L$ to $\mathcal{A}_{2P}$, which is ε-pan-private against two intrusions. $\mathcal{A}_{2P}$ simulates $\mathcal{A}_L$ and stores the transcript so far as its internal state. Since the $\mathcal{A}_L$ is ε-locally private, its public transcript is an ε-differentially private function of the data. $\mathcal{A}_{2P}$ is therefore ε-pan-private against an arbitrary number of intrusions onto its internal state.

Before stating and proving the result, note that in this section we may distinguish between random variables and realizations of streams. Thus we typically denote the stream random variables by S and the stream realizations by s.

Theorem 1. *For every $\mathcal{A}_{2P}$ that is ε-pan-private against two intrusions and generates output distribution O given input stream s, there exists $\mathcal{A}_L$ that is sequentially interactive ε-locally private and generates transcript distribution O given s, and vice-versa.*

Proof. $\Rightarrow$ (pan to local): We start by converting from pan-privacy against two intrusions to pan-privacy against one intrusion while preserving all internal states.

Lemma 1. *Suppose $\mathcal{A}_{2P}$ is ε-pan-private against two intrusions, and let $I_{2,t}$ be the random variable for the internal state of $\mathcal{A}_{2P}$ after stream element t. Then there exists $\mathcal{A}_{1P}$ that is ε-pan-private against one intrusion such that, for analogously defined $I_{1,t}$, for any stream $s_{\leq t}$, the concatenation $I_{2,1} \circ I_{2,2} \cdots \circ I_{2,t}$ is distributed identically to $I_{1,t}$.*

Proof. We first define $\mathcal{A}_{1P}$. For $j \in \{1, 2\}$, define $i_{j,t}$ to be the realized internal state of $\mathcal{A}_{jP}$ after seeing the t^{th} stream element. Each internal state $i_{1,t}$ of $\mathcal{A}_{1P}$ is a concatenation of internal states $i_{2,1} \circ \cdots \circ i_{2,t}$, and for any internal state i of $\mathcal{A}_{1P}$ we let i^{-1} denote the most recently concatenated state. For example, for $i = i_{2,1} \circ \cdots \circ i_{2,t}$, $i^{-1} = i_{2,t}$[1]. We then define the internal algorithm of $\mathcal{A}_{1P}$ by $\mathcal{A}_{1P,\mathcal{I}}(i, x) = i \circ \mathcal{A}_{2P,\mathcal{I}}(i^{-1}, x)$. Finally, we define the output algorithm of $\mathcal{A}_{1P}$ by $\mathcal{A}_{1P,\mathcal{O}}(i) = \mathcal{A}_{2P,\mathcal{O}}(i^{-1})$. As a result, $\mathcal{A}_{1P,\mathcal{O}}(\mathcal{A}_{1P,\mathcal{I}}(s)) = \mathcal{A}_{2P,\mathcal{O}}(\mathcal{A}_{2P,\mathcal{I}}(s))$, and $\mathcal{A}_{1P}$ and $\mathcal{A}_{2P}$ have identical output distributions.

We will prove this result for discrete state spaces; a similar approach works for continuous state spaces if we replace probability mass functions with densities. To prove ε-pan-privacy of $\mathcal{A}_{1P}$ against one intrusion, it suffices to fix neighboring streams s and s', internal state set i, output state set o, stream position t, and show

$$\frac{\mathbb{P}_{\mathcal{A}_{1P}}\left[\mathcal{A}_{1P,\mathcal{I}}(s_{\leq t}) = i\right] \mathbb{P}_{\mathcal{A}_{1P}}\left[\mathcal{A}_{1P,\mathcal{O}}(\mathcal{A}_{1P,\mathcal{I}}(s)) = o \mid \mathcal{A}_{1P,\mathcal{I}}(s_{\leq t}) = i\right]}{\mathbb{P}_{\mathcal{A}_{1P}}\left[\mathcal{A}_{1P,\mathcal{I}}(s'_{\leq t}) = i\right] \mathbb{P}_{\mathcal{A}_{1P}}\left[\mathcal{A}_{1P,\mathcal{O}}(\mathcal{A}_{1P,\mathcal{I}}(s')) = o \mid \mathcal{A}_{1P,\mathcal{I}}(s'_{\leq t}) = i\right]} \leq e^{\varepsilon}.$$

[1]We assume that it is possible to separate a concatenation into states of $\mathcal{A}_{2P}$ after the fact. This assumption is easily (but less neatly) removed using a separator character $\perp$.

First, by the definition of $\mathcal{A}_{1P}$, it suffices to show

$$\frac{\mathbb{P}_{\mathcal{A}_{1P}}\left[\mathcal{A}_{1P,\mathcal{I}}(s_{\leq t}) = i\right]\mathbb{P}_{\mathcal{A}_{2P}}\left[\mathcal{A}_{2P,\mathcal{O}}(\mathcal{A}_{2P,\mathcal{I}}(s)) = o \mid \mathcal{A}_{2P,\mathcal{I}}(s_{\leq t}) = i^{-1}\right]}{\mathbb{P}_{\mathcal{A}_{1P}}\left[\mathcal{A}_{1P,\mathcal{I}}(s'_{\leq t}) = i\right]\mathbb{P}_{\mathcal{A}_{2P}}\left[\mathcal{A}_{2P,\mathcal{O}}(\mathcal{A}_{2P,\mathcal{I}}(s')) = o \mid \mathcal{A}_{2P,\mathcal{I}}(s'_{\leq t}) = i^{-1}\right]} \leq e^{\varepsilon}. \tag{3.1}$$

Suppose streams s and s' differ at time t^*, i.e. $s_{t^*} \neq s'_{t^*}$. If $t^* > t$, then we immediately have $\mathbb{P}_{\mathcal{A}_{1P}}\left[\mathcal{A}_{1P,\mathcal{I}}(s_{\leq t}) = i\right] = \mathbb{P}_{\mathcal{A}_{1P}}\left[\mathcal{A}_{1P,\mathcal{I}}(s'_{\leq t}) = i\right]$, and $\frac{\mathbb{P}_{\mathcal{A}_{2P}}\left[\mathcal{A}_{2P,\mathcal{O}}(\mathcal{A}_{2P,\mathcal{I}}(s))=o|\mathcal{A}_{2P,\mathcal{I}}(s_{\leq t})=i^{-1}\right]}{\mathbb{P}_{\mathcal{A}_{2P}}\left[\mathcal{A}_{2P,\mathcal{O}}(\mathcal{A}_{2P,\mathcal{I}}(s'))=o|\mathcal{A}_{2P,\mathcal{I}}(s'_{\leq t})=i^{-1}\right]} \leq e^{\varepsilon}$ follows from the ε-pan-privacy of $\mathcal{A}_{2P}$. Thus Inequality 3.1 holds.

The remaining case is when $t^* \leq t$. Here, $\frac{\mathbb{P}_{\mathcal{A}_{2P}}\left[\mathcal{A}_{2P,\mathcal{O}}(\mathcal{A}_{2P,\mathcal{I}}(s))=o|\mathcal{A}_{2P,\mathcal{I}}(s_{\leq t})=i^{-1}\right]}{\mathbb{P}_{\mathcal{A}_{2P}}\left[\mathcal{A}_{2P,\mathcal{O}}(\mathcal{A}_{2P,\mathcal{I}}(s'))=o|\mathcal{A}_{2P,\mathcal{I}}(s'_{\leq t})=i^{-1}\right]} = 1$, and we need to upper bound $\frac{\mathbb{P}_{\mathcal{A}_{1P}}\left[\mathcal{A}_{1P,\mathcal{I}}(s_{\leq t})=i\right]}{\mathbb{P}_{\mathcal{A}_{1P}}\left[\mathcal{A}_{1P,\mathcal{I}}(s'_{\leq t})=i\right]}$. Since $\mathcal{A}_{1P,\mathcal{I}}(s_{\leq t})$ is conditionally independent of $\mathcal{A}_{1P,\mathcal{I}}(s_{\leq t^*-1})$ given $\mathcal{A}_{1P,\mathcal{I}}(s_{\leq t^*})$, it suffices to show that $\frac{\mathbb{P}_{\mathcal{A}_{1P}}\left[\mathcal{A}_{1P,\mathcal{I}}(s_{\leq t^*})=i\right]}{\mathbb{P}_{\mathcal{A}_{1P}}\left[\mathcal{A}_{1P,\mathcal{I}}(s'_{\leq t^*})=i\right]} \leq e^{\varepsilon}$. Recall that $I_{j,t}$ is the random variable for the internal state of $\mathcal{A}_j$ after seeing the t^{th} stream element. Then it is equivalent to show $\frac{\mathbb{P}_{\mathcal{A}_{1P}}\left[I_{1,t^*}=i|S_{\leq t*}=s_{\leq t*}\right]}{\mathbb{P}_{\mathcal{A}_{1P}}\left[I_{1,t^*}=i|S_{\leq t^*}=s'_{\leq t*}\right]} \leq e^{\varepsilon}$.

We introduce some additional notation to prove this claim. i is an internal state for $\mathcal{A}_{1P}$ and is therefore a concatenation of internal states for $\mathcal{A}_{2P}$. Let i_a denote the a^{th} state in the concatenation i, and let $i_{a:b} = i_a \circ i_{a+1} \circ \cdots \circ i_b$, the concatenation of states i_a through i_b. Then $\frac{\mathbb{P}_{\mathcal{A}_{1P}}\left[I_{1,t^*}=i|S_{\leq t^*}=s_{\leq t^*}\right]}{\mathbb{P}_{\mathcal{A}_{1P}}\left[I_{1,t^*}=i|S_{\leq t^*}=s'_{\leq t^*}\right]}$

$$\begin{aligned}
&= \frac{\mathbb{P}_{\mathcal{A}_{1P}}\left[I_{1,t^*-1} = i_{1:t^*-1} \mid S_{\leq t^*} = s_{\leq t^*}\right] \cdot \mathbb{P}_{\mathcal{A}_{2P}}\left[I_{2,t^*} = i_{t^*} \mid S_{\leq t^*} = s_{\leq t^*}, I_{2,t^*-1} = i_{t^*-1}\right]}{\mathbb{P}_{\mathcal{A}_{1P}}\left[I_{1,t^*-1} = i_{1:t^*-1} \mid S_{\leq t^*} = s'_{\leq t^*}\right] \cdot \mathbb{P}_{\mathcal{A}_{2P}}\left[I_{2,t^*} = i_{t^*} \mid S_{\leq t^*} = s'_{\leq t^*}, I_{2,t^*-1} = i_{t^*-1}\right]} \\
&= \frac{\mathbb{P}_{\mathcal{A}_{2P}}\left[I_{2,t^*} = i_{t^*} \mid S_{\leq t^*} = s_{\leq t^*}, I_{2,t^*-1} = i_{t^*-1}\right]}{\mathbb{P}_{\mathcal{A}_{2P}}\left[I_{2,t^*} = i_{t^*} \mid S_{\leq t^*} = s'_{\leq t^*}, I_{2,t^*-1} = i_{t^*-1}\right]} \\
&= \frac{\mathbb{P}_{\mathcal{A}_{2P}}\left[I_{2,t^*} = i_{t^*} \mid S_{t^*} = s_{t^*}, I_{2,t^*-1} = i_{t^*-1}\right]}{\mathbb{P}_{\mathcal{A}_{2P}}\left[I_{2,t^*} = i_{t^*} \mid S_{t^*} = s'_{t^*}, I_{2,t^*-1} = i_{t^*-1}\right]}
\end{aligned}$$

where the second equality uses the fact that $s_{<t^*} = s'_{<t^*}$, and the third equality uses I_{2,t^*}'s conditional independence from $S_{\leq t^*-1}$ given I_{2,t^*-1}. Now, since I_{2,t^*-1} and S_{t^*} are

independent, we multiply by $1 = \frac{\mathbb{P}_{\mathcal{A}_{2P}}\left[I_{2,t^*-1}=i_{t^*-1} | S_{t^*}=s_{t^*}\right]}{\mathbb{P}_{\mathcal{A}_{2P}}\left[I_{2,t^*-1}=i_{t^*-1} | S_{t^*}=s'_{t^*}\right]}$ to get

$$\frac{\mathbb{P}_{\mathcal{A}_{2P}}\left[I_{2,t^*} = i_{t^*} \mid S_{t^*} = s_{t^*}, I_{2,t^*-1} = i_{t^*-1}\right]}{\mathbb{P}_{\mathcal{A}_{2P}}\left[I_{2,t^*} = i_{t^*} \mid S_{t^*} = s'_{t^*}, I_{2,t^*-1} = i_{t^*-1}\right]} = \frac{\mathbb{P}_{\mathcal{A}_{2P}}\left[I_{2,t^*} = i_{t^*}, I_{2,t^*-1} = i_{t^*-1} \mid S_{t^*} = s_{t^*}\right]}{\mathbb{P}_{\mathcal{A}_{2P}}\left[I_{2,t^*} = i_{t^*}, I_{2,t^*-1} = i_{t^*-1} \mid S_{t^*} = s'_{t^*}\right]}$$
$$\leq e^{\varepsilon}$$

since $\mathcal{A}_{2P}$ is ε-pan-private against two intrusions. □

Next, we show how to convert this pan-private algorithm $\mathcal{A}_{1P}$ into an equivalent locally private algorithm $\mathcal{A}_L$.

Lemma 2. *Let $\mathcal{A}_{1P}$ be an ε-pan-private algorithm as described in Lemma 1. Then there exists sequentially interactive ε-locally private algorithm $\mathcal{A}_L$ whose transcript distribution Π_t is identical to the $\mathcal{A}_{1P}$'s state distribution I_t at each time t.*

Proof. At each time t, $\mathcal{A}_{1P}$ computes a function $\mathcal{A}_{1P}(i_{t-1}, s_t)$ of its current state and the current element in the stream and concatenates it to its current state. We define $\mathcal{A}_L$ to use $\mathcal{A}_{1P}(i_{t-1}, \cdot)$ as a randomizer, add the result $\mathcal{A}_{1P}(i_{t-1}, s_t)$ to the transcript, and continue.

$\mathcal{A}_L$ is sequentially interactive because we take a single pass through the stream. Furthermore, because $\mathcal{A}_{1P}$ is ε-pan-private and maintains all previous states, the transcript Π_t of $\mathcal{A}_L$ is an ε-differentially private function of the user data. Thus $\mathcal{A}_L$ is ε-locally private. Finally, recalling that Definition 4 defined a transcript to record not only outputs but the randomizers used as well, let Π_t^{-R} denote Π_t with the randomizers omitted. Then for any input stream s, Π_t^{-R} is distributed identically to I_t. □

We now combine Lemma 1 and Lemma 2: any $\mathcal{A}_{2P}$ that is ε-pan-private against two intrusions yields a sequentially interactive ε-locally private $\mathcal{A}_L$ such that for any input stream s and time t, $I_{2,t}$ is distributed identically to $\Pi_t^{-R,-1}$, the most recent addition to the transcript.

$\Leftarrow$ (local to pan): Let $\mathcal{A}_L : \Pi \to R$ be a sequentially interactive ε-locally private protocol mapping transcripts to randomizers, and let $\mathcal{A}_{\mathcal{I}} : \mathcal{I} \times \mathcal{X} \to \mathcal{I}$ be $\mathcal{A}_{2P}$'s internal algorithm with initial state $\emptyset$. We define $\mathcal{A}_{\mathcal{I}}(\emptyset, x_1) = (\emptyset, \mathcal{A}_L(\emptyset), \mathcal{A}_L(\emptyset)(x_1))$ and define other internal states i by $\mathcal{A}_{\mathcal{I}}(i, x) = i \circ (\mathcal{A}_L(i), \mathcal{A}_L(i)(x))$, the concatenation of the existing state i and the (randomizer, output) pair $(\mathcal{A}_L(i), \mathcal{A}_L(i)(x))$. Thus $I_t = \Pi_t$ at each time t. Finally, we define the output algorithm to be the identity function $\mathcal{A}_{\mathcal{O}}(i) = i$.

Since $\mathcal{A}_L$ is ε-locally private, its final transcript Π is an ε-differentially private function of the stream: for any transcript realization π and neighboring streams s and s', $\frac{\mathbb{P}_{\mathcal{A}_L}[\Pi=\pi|S=s]}{\mathbb{P}_{\mathcal{A}_L}[\Pi=\pi|S=s']} \leq e^{\varepsilon}$. Letting I^* be a random variable for the final internal state of $\mathcal{A}_{2P}$, it follows that $\frac{\mathbb{P}_{\mathcal{A}_{2P}}[I^*=\pi|S=s]}{\mathbb{P}_{\mathcal{A}_{2P}}[I^*=\pi|S=s']} \leq e^{\varepsilon}$. Thus the final internal state I of $\mathcal{A}_{2P}$ is also an ε-differentially private function of the stream. Moreover, because it is a transcript, I^* includes a record of all previous internal states. Thus the additional view of any two internal states (in fact, any number of internal states) is still an ε-differentially private function of the stream: fixing times $t_1, \ldots, t_c$ and corresponding internal states $\pi_1, \ldots, \pi_c$,

$$\frac{\mathbb{P}_{\mathcal{A}_{2P}}\left[I_{t_1} = \pi_1, \ldots, I_{t_c} = \pi_c, I^* = i \mid S = s\right]}{\mathbb{P}_{\mathcal{A}_{2P}}\left[I_{t_1} = \pi_1, \ldots, I_{t_c} = \pi_c, I^* = i \mid S = s'\right]} \leq e^{\varepsilon}.$$

Finally, since the output of $\mathcal{A}_{2P}$ is the final state I^*, $\mathcal{A}_{2P}$ is ε-pan-private against arbitrarily many (and, in particular, two) intrusions. □

We view this result as dictating the scope of when pan-privacy is reasonable. If a user requires privacy against multiple intrusions, then the operator suffers no utility loss (and users enjoy privacy gains) by using an algorithm that is locally private instead of an algorithm that is pan-private against multiple intrusions. However, there are cases where a user may be satisfied with pan-privacy against a single intrusion. To see why, we use the following simple fact.

Fact 3. *Suppose a user's datum is element s_t of an (ε, δ)-pan-private algorithm $\mathcal{A}$'s stream. We say an intrusion occurs at time t if the intrusion occurs immediately after $\mathcal{A}$ updates*

its internal state to i_t after seeing element s_t. If

1. *the first intrusion (possibly of many) occurs at time $t' \geq t$, or*

2. *all intrusions occur at times $t' < t$,*

then the intruder's view is an (ε, δ)-differentially private function of s_t.

Proof. Pan-privacy guarantees that i_t is an (ε, δ)-differentially private function of s_t. In Case 1, the adversary only sees a post-processing of i_t. Differential privacy's resilience to post-processing (Fact 1) implies that this view is (ε, δ)-differentially private in s_t. In Case 2, the adversary's view is independent of s_t, so (ε, δ)-differential privacy is immediate. □

By Fact 3, if $\mathcal{A}$ is pan-private against a single intrusion, then it guarantees privacy for users who either contribute data before the first intrusion or after all intrusions. However, pan-privacy is not sufficient to protect a user's privacy if the operator has already been compromised and may be compromised again. The key parameter for pan-privacy is therefore the user's trust in the operator when the user contributes their data. This motivates the trust model described in the introduction: if a user trusts the operator today, but wants to "future-proof" themselves for tomorrow, then pan-privacy is a reasonable privacy guarantee.

Chapter 4

Connection: Fully and Sequentially Interactive Local Privacy

We now turn to our second connection, between fully and sequentially interactive local privacy [47]. We show that for any ε-locally private protocol, we can exhibit a sequentially interactive 3ε-locally private protocol inducing exactly the same transcript distribution[1]. Thus for any task for which the original protocol was useful, the sequentially interactive protocol is just as useful. The cost of this transformation is an increase in sample complexity proportional to both ε and the *compositionality* of the original fully interactive protocol. Informally, a protocol with high compositionality (Definition 10) solicits many outputs from users but still guarantees a low privacy parameter because few of the outputs reveal information.

Our proof is constructive; given an arbitrary k-compositional ε-locally private protocol we show how to simulate it using a sequentially interactive protocol that induces the same joint distribution on transcripts. The "simulation" is driven by three main ideas:

1. **Bayesian Resampling**: The dataset used in a locally differentially private protocol does not change after the protocol starts. However, we consider the following thought experiment: whenever a user sends data through a randomizer, they first *resample* their datum from the posterior distribution on their datum conditioned on the transcript so far. This induces exactly the same joint distribution on datasets and transcripts upon completion of the mechanism. The remainder of this experiment therefore seeks to simulate this Bayesian resampling mechanism.

[1] Formally, for any loss function defined over a data distribution $\mathcal{D}$ and a transcript Π, when data points x_i are drawn i.i.d. from $\mathcal{D}$, the two protocols induce exactly the same distribution over transcripts, and hence the same distribution over losses. This is the sense in which the two protocols are equivalent.

2. **Rejection Sampling**: Local privacy ensures that the posterior on a user's datum conditioned on the transcript so far must be close to their prior (otherwise, the transcript reveals too much information). Thus, it is possible to sample from this posterior distribution by first sampling from the prior, and then applying a rejection sampling step that is both a) likely to succeed, and b) private. It is easy to sample from the prior by simply querying a new user, but it is not obvious how users can rejection sample without knowing the underlying data distribution $\mathcal{D}$. We show that an application of Bayes rule, together with a data independent rescaling, can be used to re-write the required rejection probability using only quantities that each user can compute from her own data point and the transcript. A similar use of rejection sampling appears in the simulation of locally private algorithms by statistical query algorithms in Kasiviswanathan et al. [50].

3. **Randomizer Decomposition**: The two ideas above suffice to transform a fully interactive mechanism into a sequentially interactive mechanism, with a blowup in sample complexity from the original sample complexity n to the original *round* complexity T. To mitigate this, we generalize a recent result of Balle, Bell, Gascón, and Nissim [11] to show that any ε_i-private randomizer can be described as a mixture between a *data independent* distribution and an ε-private andomizer for any $\varepsilon > \varepsilon_i$. The key element of this decomposition is that the weight on the data independent distribution is roughly (for small constant ε) $1 - \varepsilon_i/\varepsilon$. Thus we can simulate each local randomizer while only needing to query a new user with probability $\varepsilon_i/\varepsilon$. This dependence on $\varepsilon_i\varepsilon$ — roughly, the privacy of the isolated randomizer output relative to the privacy of the protocol as a whole — leads to our use of compositionality. We show that if the original protocol is k-compositional, then k users are required in expectation to carry out the sequential simulation, and the realized sample complexity concentrates sharply around its expectation.

4.1. Additional Preliminaries

We start with some additional formalism that will be useful in constructing our transformation. We will speak separately of protocols and *experiments*. While the protocol $\mathcal{A}$ is a function mapping transcripts to users and randomizers, the experiment is the interactive process that maps a protocol and collection of users drawn from a distribution $\mathcal{D}$ to a finished transcript. This will allow us to modify protocols in transformations.

In the simplest case, FollowerExpt (Algorithm 1), the experiment exactly follows the outputs of its protocol. However, experiments may in general heed, modify, or ignore the outputs of their input protocol. We delineate the privacy characteristics of experiment-protocol pairs and protocols in isolation below. Here and throughout, the dataset is not viewed as an input to an experiment, but is drawn from $\mathcal{D}$ by the experiment-protocol pair. Drawing a fresh user $\sim \mathcal{D}$ corresponds to adding an additional data point, and so the sample complexity of an experiment-protocol pair is the number of draws from $\mathcal{D}$ over the run of the algorithm. For the simple algorithm $\mathsf{FollowerExpt}(\mathcal{A})$ defined above, the sample complexity is always n. Although the distribution $\mathcal{D}$ and the sample complexity n are inputs to the experiment, for brevity we typically omit them and focus on the protocol $\mathcal{A}$; e.g. writing $\mathsf{Expt}(\mathcal{A})$ rather than $\mathsf{Expt}(\mathcal{A}, \mathcal{D}, n)$.

Algorithm 1

procedure $\mathsf{FollowerExpt}(\mathcal{A}, \mathcal{D}, n)$
    Draw n users $\{x_i\} \sim \mathcal{D}^n$
    Initialize transcript $\pi_0 \leftarrow \emptyset$
    for $t = 1, 2, \ldots$ **do**
        if $\mathcal{A}(\pi_{<t}) = \perp$ **then**
            Output transcript $\pi_{<t}$
        else
            $(i^t, R_t, \varepsilon_t, \delta_t) \leftarrow \mathcal{A}(\pi_{<t})$
            User i^t publishes $y_t \sim R_t(x_{i^t}, \varepsilon_t, \delta_t)$

Definition 9. *Experiment-protocol pair* $\mathsf{Expt}(\mathcal{A})$ *satisfies* (ε, δ)-local privacy *if it is* (ε, δ)-*differentially private in its transcript outputs.*

This experiment-protocol formalism will be useful in constructing the full-to-sequential re-

duction later in this section; elsewhere, we typically elide the distinction and simply reason about $\mathsf{FollowerExpt}(\mathcal{A})$ as "protocol $\mathcal{A}$".

At each round t of a fully interactive ε-locally private protocol, we know that the privacy parameter ε_t of any randomizer used in that round is $\leq \varepsilon$. For many protocols, we can say more about how the ε_t parameters relate to ε:

Definition 10. *Consider an ε-locally private protocol $\mathcal{A}$ where $\mathcal{A}$ is not ε'-locally private for $\varepsilon' < \varepsilon$. Let $\{\varepsilon_t\}_{t=1}^T$ denote the minimal privacy parameters of the local randomizers R_t selected at round t considered as random variables. We say the protocol $\mathcal{A}$ is* k-compositionally private *if for all $i \in [n]$, with probability 1 over the randomness of the transcript,*

$$\sum_{t:i_t=i} \varepsilon_t \leq k\varepsilon.$$

If $k = 1$, a protocol is simply compositional private*.*

In fact, all of our results hold without modification even under the weaker condition of *average* k-compositionality. For a protocol $\mathcal{A}$ with sample complexity n, $\mathcal{A}$ is k-compositional on average if

$$\sum_{t} \varepsilon_t \leq k\varepsilon n.$$

For brevity, we often shorthand "k-compositionally private" as simply "k-compositional".

Informally, a compositionally private protocol is one in which the privacy parameters for each user "just add up." Almost every locally private protocol studied in the literature (and in particular, every protocol whose privacy analysis follows from the composition theorem for pure differential privacy) is compositionally private[2]. They are so ubiquitous that it is tempting to guess that all ε-locally private protocols are compositional. However, this is false: for every k and ε, there are ε-locally private protocols that fail to be k-compositionally private. The following example shows that by taking advantage of special structure in the

[2]This simple compositionality applies even if $\{\varepsilon_t\}_{t=1}^T$ are chosen adaptively in each round (see Theorem 3.6 in the work of Rogers, Roth, Ullman, and Vadhan [58]).

data domain and choice of randomizers it is possible to achieve ε-local privacy, even as the sum of the round-by-round privacy parameters greatly exceeds ε.

Example 1 (Informal)**.** *Let the data universe $\mathcal{X}$ consist of the canonical basis vectors $e_1, \ldots, e_d \in \{0,1\}^d$, and let each $x_1, \ldots, x_n$ be an arbitrary element of $\mathcal{X}$. Consider the d round protocol where, for each round $j \in [d]$, every user i with $x_i = e_j$ outputs a sample from $\mathsf{RR}(1, \varepsilon)$, and the remaining users output a sample from $\mathsf{Ber}(0.5)$. As $\mathsf{RR}(\cdot, \varepsilon)$ is an ε-local randomizer which each user employs only once, and remaining outputs are data-independent, this protocol is ε-locally private. But the protocol fails to be k-compositional for $k < d/2$.*

The preceding example demonstrates that the careful choice of local randomizers based on the data universe structure can strongly violate compositional privacy. Seen another way, when multiple queries are asked of the same user, there are situations in which the correlation in privatized responses induced by being run on the same data element can lead to arbitrarily sub-compositional privacy costs. The main result of our paper is that the additional power of a full interactivity on top of sequential interactivity is characterized by its compositionality.

Finally, since it will be useful to explicitly specify privacy parameters in these transformations, in this section we view the parameters as explicit outputs of $\mathcal{A}$, rather than encoding them in each randomizer R. $\mathcal{A}$ thus outputs a tuple $(i^t, R_t, \varepsilon_t, \delta)$ rather than just $(i^t, R_t$ s in Definition 5.

4.2. Step 1: Bayesian View

The first step of our construction is to observe that for any locally private protocol $\mathcal{A}$, the experiment-protocol pair $\mathsf{BayesExpt}(\mathcal{A})$ induces exactly the same distribution over transcripts as $\mathsf{FollowerExpt}(\mathcal{A})$. The difference is that $\mathsf{BayesExpt}(\mathcal{A})$ does not follow $\mathcal{A}$ as $\mathsf{FollowerExpt}(\mathcal{A})$ does. Instead, between each interaction with a given user i, $\mathsf{BayesExpt}(\mathcal{A})$ has user i resample x_i from the posterior distribution on their data conditioned on the

transcript so far. We prove in Lemma 3 that the two experiments produce exactly the same transcript distribution.

Algorithm 2

procedure BayesExpt($\mathcal{A}, \mathcal{D}, n$)
  Initialize transcript $\pi_0 = \emptyset$
  for $t = 1, 2, \ldots$ **do**
    if $\mathcal{A}(\pi_{<t}) = \perp$ **then**
      Output transcript $\pi_{<t}$
    else
      $(i^t, R_t, \varepsilon_t, \delta_t) \leftarrow \mathcal{A}(\pi_{<t})$
      Redraw $x_{i^t} \sim Q_{i,t}$ ▷ $Q_{i,t}$ is the posterior on x_{i^t} given $\pi_{<t}$
      User i^t publishes $y_t \sim R_t(x_{i^t})$

Lemma 3. *For any protocol $\mathcal{A}$, Let Π^F be the transcript random variable that is output by* FollowerExpt($\mathcal{A}$) *and let Π^B be the transcript output by* BayesExpt($\mathcal{A}$)*. Then*

$$\Pi^F \stackrel{d}{=} \Pi^B$$

where $\stackrel{d}{=}$ denotes equality of distributions.

Proof. We show this by (strong) induction on rounds in the transcript. The base case $t = 1$ is immediate: for any index i^1 selected by BayesExpt($\mathcal{A}$), the posterior distribution $Q_{i,1}$ is the same as the prior $\mathcal{D}$.

Now suppose it is true through time t, i.e. $\Pi^F_{\leq t} \stackrel{d}{=} \Pi^B_{\leq t}$. Then since the joint distributions $\Pi_{\leq t+1}$ factor as $(i^{t+1}, R_{t+1}, \varepsilon_{t+1}, \delta_{t+1}, Y_{t+1} | \Pi_{\leq t}) \cdot \Pi_{\leq t}$, it suffices to show that the conditional distributions $i^{t+1}, R_{t+1}, \varepsilon_{t+1}, \delta_{t+1}, Y_{t+1} | \Pi_{\leq t}$ coincide. Moreover, the conditional distribution on $i^{t+1}, R_{t+1}, \varepsilon_{t+1}, \delta_{t+1} | \Pi_{<t+1}$ is given by $\mathcal{A}(\Pi_{<t+1})$ under both algorithms, and so it remains only to show that $Y_{t+1} | i^{t+1}, R_{t+1}, \varepsilon_{t+1}, \delta_{t+1}, \Pi_{\leq t}$ is identically distributed under both algorithms. Under FollowerExpt($\mathcal{A}$),

$$Y_{t+1} | i^{t+1}, R_{t+1}, \varepsilon_{t+1}, \delta_{t+1}, \Pi_{\leq t} \sim R_{t+1}(x_{i^{t+1}}, \varepsilon_{t+1}, \delta_{t+1} | \Pi_{\leq t}) \stackrel{d}{=} R_{t+1}(u, \varepsilon_{t+1}, \delta_{t+1}),$$

where $u \stackrel{d}{=} x_{i^{t+1}} | \Pi_{\leq t} \stackrel{d}{=} Q_{i,t+1}$ by definition, and we use the fact that after conditioning

on $\Pi_{\leq t}$, $x_{i^{t+1}}$ is independent of ε_{t+1} and δ_{t+1}. Redrawing $u \sim Q_{i,t+1}$ does not change the marginal distribution of $R_{t+1}(u, \varepsilon_{t+1}, \delta_{t+1})$, which is exactly the distribution under $\mathsf{BayesExpt}(\mathcal{A})$, as desired. □

With Lemma 3, our new goal will be to simulate the transcript distribution induced by $\mathsf{BayesExpt}(\mathcal{A})$. This is useful because, intuitively, the repeated sampling of $\mathsf{BayesExpt}(\mathcal{A})$ for each randomizer call is closer to a sequentially interactive protocol, which must query a new user for each randomizer call. However, we still require a method for actually redrawing each $x_{i^t} \sim Q_{i,t}$ in step 8 of Algorithm 2.

4.3. Step 2: Rejection Sampling

We now show how to replace step 8 in Algorithm 2 by selecting a new datapoint (drawn from $\mathcal{D}$) at every round and using rejection sampling to simulate a draw from $Q_{i,t}$. The result is a sequentially interactive mechanism that preserves the transcript distribution of Algorithm 2 (and, by Lemma 3, of Algorithm 1), albeit one with a potentially very large increase in sample complexity (from n to T). The rejection sampling step increases the privacy cost of the protocol by at most a factor of 2.

We first review why it is non-obvious that rejection sampling can be performed in this setting. We want to sample from the target distribution $Q_{i,t}$, the posterior $x_i^t | \pi_{<t}$, using samples from the original data distribution $\mathcal{D}$. Let p_Q denote the density function of $Q_{i,t}$ and let p denote the density function of $\mathcal{D}$. In rejection sampling, we would typically sample $u \sim \mathcal{D}$, and with probability $\propto \frac{p_Q(u)}{p(u)}$ we would accept u as a sample drawn from $Q_{i,t}$, or else redraw another u and continue.

This is not immediately possible in our setting, since the individuals (who must perform the rejection sampling computation) do not know the prior density p and hence do not know the posterior p_Q. As a result, they cannot compute either the numerator or denominator of the expression for the acceptance probability. We solve this problem by using the fact

that we are simulating a posterior with a prior distribution, and formulate the rejection sampling probability ratio as a quantity depending only on a user's private data point and the transcript. Users may then compute this quantity themselves. Crucially, this formulation relies on the local privacy of the transcript so far: because privacy bounds the contribution of any one user's datum to the transcript, any posterior data distribution conditioned on the transcript is not far from the prior.

To define our transformed rejection sampler we set up some new notation: given a user i and round t, let $\pi_{<t,i}$ denote the subset of the realized transcript up to time t that corresponds to user i's data, i.e. $\pi_{<t,i} = \{(i^{t'}, R_{t'}, \varepsilon_{t'}, \delta_{t'}, y_{t'}) : t' < t, i^{t'} = i\}$. Let $\mathbb{P}_{x_i}[\pi_{<t,i}]$ denote the conditional probability of the messages corresponding to user i given the choices of privacy parameters and randomizers up to time t:

$$\mathbb{P}[\pi_{<t,i}] = \prod_{t':i^{t'}=i} \mathbb{P}_{R_{t'}}[R_{t'}(x_i, \varepsilon_{t'}, \delta_{t'}) = y_{t'}].$$

Using this notation, we define our rejection sampling procedure RejSamp in Algorithm 3.

Algorithm 3 Rejection Sampling

procedure RejSamp$(i, \pi_{<t}, \varepsilon, \varepsilon_t, R_t(\cdot), \mathcal{D})$ ▷ Publishing $\Pi_{<t}$ is ε-private
  Initialize indicator accept $\leftarrow 0$
  while accept $= 0$ **do**
    Draw a new user $x \sim \mathcal{D}$
    User x computes $p_x \leftarrow \frac{\mathbb{P}_x[\pi_{<t,i}]}{\max_{x^*} \mathbb{P}_{x^*}[\pi_{<t,i}]}$
    User x publishes accept $\sim$ Ber$(p_x/2)$
    if accept $= 1$ **then**
      User x outputs $Y'_t \sim R_t(x, \varepsilon_t)$

We now prove that RejSamp is private and does not need to sample many users.

Lemma 4. *Let $Y_t \stackrel{d}{=} R_t(x')$, where $x' \sim Q_{i,t}$ and let Y'_t be defined by the rejection sampling algorithm* RejSamp. *Let the sample complexity N be the total number of new users x drawn in step 4 of* RejSamp. *Then* RejSamp *is $(\varepsilon + \varepsilon_t, 0)$-locally private, $Y_t \stackrel{d}{=} Y'_t$, and $\mathbb{E}[N] \leq 2e^{\varepsilon}$.*

Proof of Lemma 4. We start with the privacy guarantee.

Claim 1. RejSamp *is* $(\varepsilon + \varepsilon_t)$*-locally private.*

Proof. We first show that publishing a draw from $\mathsf{Ber}\left(p_x/2\right)$ is ε-locally private. For any input x,

$$\mathbb{P}\left[\text{output } 1 \mid x\right] = \frac{p_x}{2} = \frac{\mathbb{P}_x\left[\pi_{<t,i}\right]}{2\max_{x^*}\mathbb{P}_{x^*}\left[\pi_{<t,i}\right]} \in \left[\frac{e^{-\varepsilon}}{2}, \frac{1}{2}\right]$$

where the last step uses the ε-local privacy of $\pi_{<t,i}$. Therefore for any inputs x, x',

$$\mathbb{P}\left[\text{output } 1 \mid x\right] \leq e^{\varepsilon}\mathbb{P}\left[\text{output } 1 \mid x'\right].$$

Similarly,

$$\mathbb{P}\left[\text{output } 0 \mid x\right] = 1 - \frac{p_x}{2} \in \left[\frac{1}{2}, \frac{2e^{\varepsilon} - 1}{2e^{\varepsilon}}\right]$$

and by $1 + x \leq e^x$, we get $1 - \varepsilon \leq e^{-\varepsilon}$, so $2 - e^{-\varepsilon} \leq 1 + \varepsilon \leq e^{\varepsilon}$ and

$$\frac{2e^{\varepsilon} - 1}{2e^{\varepsilon}} \leq \frac{e^{\varepsilon}}{2}.$$

Thus for any inputs x, x',

$$\mathbb{P}\left[\text{output } 0 \mid x\right] \leq e^{\varepsilon}\mathbb{P}\left[\text{output } 0 \mid x'\right].$$

Finally, releasing $R_t(x, \varepsilon_t)$ is ε_t-locally private, so by composition the whole process is $(\varepsilon + \varepsilon_t)$-locally private. □

Next, we show that our rejection sampling induces the correct distribution.

Claim 2. $Y_t \overset{d}{=} Y_t'$

Proof. It suffices to show that $x \mid \text{ accept } = 1 \overset{d}{=} Q_{i,t}$. Fix any input x_0. Then by Bayes'

rule in the first and second-to-last equalities,

$$\begin{aligned}
\mathbb{P}\left[x = x_0 \mid \mathsf{accept} = 1\right] &= \mathbb{P}\left[\mathsf{accept} = 1 \mid x = x_0\right] \cdot \frac{\mathbb{P}\left[x = x_0\right]}{\mathbb{P}\left[\mathsf{accept} = 1\right]} \\
&= \frac{\mathbb{P}_{x_0}\left[\pi_{<t,i}\right]}{2\max_{x^*} \mathbb{P}_{x^*}\left[\pi_{<t,i}\right]} \cdot \frac{\mathbb{P}\left[x = x_0\right]}{\sum_{x'} \mathbb{P}\left[x = x'\right] \frac{\mathbb{P}_{x'}[\pi_{<t,i}]}{2\max_{x^*} \mathbb{P}_{x^*}[\pi_{<t,i}]}} \\
&= \frac{\mathbb{P}_{x_0}\left[\pi_{<t,i}\right] \mathbb{P}\left[x = x_0\right]}{\sum_{x'} \mathbb{P}\left[x = x'\right] \mathbb{P}_{x'}\left[\pi_{<t,i}\right]} \\
&= \frac{\mathbb{P}_{x_0}\left[\pi_{<t,i}\right] \mathbb{P}\left[x = x_0\right]}{\mathbb{P}\left[\pi_{<t,i}\right]} \\
&= \mathbb{P}\left[x = x_0 \mid \pi_{<t,i}\right] \stackrel{d}{=} Q_{i,t}.
\end{aligned}$$

□

Finally, since $p_x/2 \geq \frac{1}{2e^\varepsilon}$, the expected number of samples until $\mathsf{accept} = 1$ is $\leq 2e^\varepsilon$. □

4.4. Step 3: Randomizer Decomposition

The preceding sections enable us to simulate a fully interactive k-compositional ε-locally private protocol with a sequentially interactive $(2\varepsilon, 0)$-locally private protocol. However, our solution so far requires sampling (in expectation) multiple new users for each query in the original protocol. Since a fully interactive protocol's query complexity may greatly exceed its sample complexity, this is undesirable. To address this problem, we decompose each local randomizer in a way that substantially reduces the number of queries that actually require samples.

Let $R : \mathcal{X} \to \mathcal{Y}$ be an ε'-local randomizer, fix one arbitrary input element x_0, and let x be a given private input to R. Then Lemma 5.2 from Balle et al. [11] shows that we can write $R(x)$ as a mixture

$$R(x) \stackrel{d}{=} (1 - \gamma)w + \gamma d_x$$

where w is a data-independent distribution, d_x is a data-dependent distribution, and $1 - \gamma \geq e^{-\varepsilon'}$. This suggests that we can use decomposition to answer a portion of queries from data-

independent distributions and reduce the final sample complexity of our solution.

Unfortunately, this approach encounters a problem: the data dependent distribution need not be differentially private. In fact, the data dependent distribution often corresponds to a point mass on the private data point. Thus the privacy of the mechanism above relies on not releasing *which* of the two mixture distributions the output was sampled from.

We fix this problem by generalizing the result of Balle et al. [11]. We show that for any $\varepsilon \geq \varepsilon'$, we can write

$$R(x) = (1-\gamma)w + \gamma\tilde{R}(x)$$

where $\tilde{R}$ is now a 2ε-local randomizer, and $\gamma = \frac{e^{-\varepsilon'}-1}{e^{-\varepsilon}-1}$ (Lemma 5). The upshot of this generalization is that even if we make public which part of the mixture distribution was used, the resulting privacy loss is still bounded by 2ε. Larger values of ε increase our chance of sampling from a data-independent distribution when simulating a local randomizer, while increasing the privacy cost incurred by a user in the event that we sample from the data-dependent mixture component. This trade-off between the number of new samples and the privacy guarantee for the new samples will be crucial for us in the proof of our main result.

Lemma 5 (Data Independent Decomposition)**.** *Let $R : \mathcal{X} \to \mathcal{Y}$ be an ε'-local randomizer and let $\varepsilon \geq \varepsilon'$. Then there exists a mapping $\tilde{R}$ and fixed data-independent distribution w such that $\tilde{R}(\cdot)$ is a $2\varepsilon-$local randomizer and*

$$R(x) \stackrel{d}{=} (1-\gamma)w + \gamma\tilde{R}(x),$$

where $\gamma = \frac{e^{-\varepsilon'}-1}{e^{-\varepsilon}-1}$.

Proof. Let $\varepsilon \geq \varepsilon' > 0$, fix single arbitrary input x_0, let $\gamma = \frac{e^{-\varepsilon'}-1}{e^{-\varepsilon}-1}$, and let $r(x)$ denote the density function of the local randomizer R with input x implicitly evaluated at some arbitrary point in the range, which we suppress. Since $\varepsilon \geq \varepsilon' > 0$, we get $e^{-\varepsilon} - 1 \leq$

$e^{-\varepsilon'} - 1 < 0$, so $\gamma \in [0, 1]$ is a valid mixture probability. Thus we can write

$$r(x) = (r(x) - (1-\gamma)r(x_0)) + (1-\gamma)r(x_0)$$

and rewrite the first term as

$$r(x) - (1-\gamma)r(x_0) = \gamma(r(x_0) + \frac{1}{\gamma}(r(x) - r(x_0))) = \gamma\tilde{r}(x).$$

$\tilde{r}$ defines a new mapping $\tilde{R}(\cdot)$ by mapping x to the random variable $\tilde{R}(x)$ with density function $\tilde{r}(x) = r(x_0) + \frac{1}{\gamma}(r(x) - r(x_0))$. Thus, it suffices to show that the mapping $\tilde{R}(x)$ is a 2ε-local randomizer.

We first show that for any x, $\tilde{r}(x)$ is a well-defined density function. Since R is an ε'-local randomizer, $r(x) - r(x_0) \geq (e^{-\varepsilon'} - 1)r(x_0)$, and so

$$\begin{aligned}
\tilde{r}(x) &= r(x_0) + \frac{1}{\gamma}(r(x) - r(x_0)) \\
&\geq r(x_0)\left(1 + \frac{e^{-\varepsilon'} - 1}{\gamma}\right) \\
&= r(x_0)e^{-\varepsilon}.
\end{aligned}$$

This establishes that $\tilde{r}(x)$ is non-negative. Then since

$$\int_\Omega \tilde{r}(x) = \int_\Omega r(x_0) + \frac{1}{\gamma}\int_\Omega (r(x) - r(x_0)) = 1 + \frac{1}{\gamma}(1-1) = 1,$$

$\tilde{r}(x)$ defines a valid density function for any x.

To see that $\tilde{r}$ is also a 2ε-local randomizer, fix any outcome $o \in \mathcal{Y}$ and any other input x'.

Since r is an ε'-local randomizer, $r(x) - r(x_0) \leq r(x_0)(e^{\varepsilon'} - 1)$ and we get

$$\begin{aligned}
\tilde{r}(x) &= r(x_0) + \frac{1}{\gamma}(r(x) - r(x_0)) \\
&\leq r(x_0)\left[1 + \frac{1}{\gamma}\left(e^{\varepsilon'} - 1\right)\right] \\
&= r(x_0)\left[1 + \frac{1 - e^{-\varepsilon}}{1 - e^{-\varepsilon'}}\left(e^{\varepsilon'} - 1\right)\right] \\
&= r(x_0)\left[1 + e^{\varepsilon'} \cdot \frac{1 - e^{-\varepsilon}}{1 - e^{-\varepsilon'}}\left(1 - e^{-\varepsilon'}\right)\right] \\
&= r(x_0)\left[1 + e^{\varepsilon'} \cdot \left(1 - e^{-\varepsilon}\right)\right] \\
&\leq r(x_0)\left[1 + e^{\varepsilon} \cdot \left(1 - e^{-\varepsilon}\right)\right] = r(x_0)e^{\varepsilon}.
\end{aligned}$$

We already showed $\tilde{r}(x') \geq r(x_0)e^{-\varepsilon}$, so

$$\frac{\tilde{r}(x)(o)}{\tilde{r}(x')(o)]} \leq \frac{e^{\varepsilon} r(x_0)(o)}{e^{-\varepsilon} r(x_0)(o)} \leq e^{2\varepsilon}.$$

□

4.5. Complete Simulation

Finally, we combine rejection sampling and decomposition to give our complete reduction, Algorithm 4. We use rejection sampling to convert from a fully interactive mechanism to a sequentially interactive one and use our data-independent decomposition of local randomizers to reduce the sample complexity of the converted mechanism.

We now prove that Reduction has the desired interactivity, privacy, transcript, and sample complexity guarantees. We again denote by N the number of samples drawn from the prior $\mathcal{D}$ over the run of the algorithm, noting that sampling from the prior $\mathcal{D}$ simply corresponds to using a new datum drawn from $\mathcal{D}$. Fixing a protocol $\mathcal{A}$, let Π^R denote the transcript random variable generated by $\mathsf{Reduction}(\mathcal{A})$, and let Π^B denote the transcript random variable generated by $\mathsf{BayesExpt}(\mathcal{A})$.

Algorithm 4 Reduction

1: **procedure** Reduction(Fully interactive $\varepsilon-$LDP Protocol $\mathcal{A}, \mathcal{D}, n$)
2: Initialize $s_1, \ldots, s_n \leftarrow 0$. ▷ indicator if user i has been selected yet
3: **for** $t = 1 \ldots$ **do**
4: **if** $\mathcal{A}(\pi_{<t}) = \perp$ **then**
5: Output transcript $\pi_{<t}$
6: **else**
7: $(i^t, R_t, \varepsilon_t) \leftarrow \mathcal{A}(\pi_{<t})$
8: **if** $s_i^t = 1$ **then**
9: Let $\gamma \leftarrow \frac{e^{-\varepsilon_t}-1}{e^{-\varepsilon}-1}$
10: Let $R_t = (1-\gamma)R_t(x_0) + \gamma\tilde{R}_t$ ▷ Randomizer decomposition
11: Draw $\rho \sim \mathsf{Unif}(0,1)$
12: **if** $\rho \leq \gamma$ **then**
13: Draw $Y_t \sim \mathsf{RejSamp}(i^t, \pi_{<t}, \varepsilon, 2\varepsilon, \tilde{R}(\cdot), \mathcal{D})$
14: **else**
15: Draw $Y_t \sim R_t(x_0, \varepsilon_t)$ ▷ Data independent distribution
16: **else**
17: Draw $x_{i^t} \sim Q_{i,t} = \mathcal{D}$, then draw $Y_t \sim R_t(x_{i^t}, \varepsilon_t)$ ▷ $Q_{i,t} = \mathcal{D}$ since $s_{i^t} = 0$
18: Let $s_{i^t} \leftarrow 1$

Theorem 2. *Let $\mathcal{A}$ be a k-compositional ε-locally private protocol. Then*

1. $\mathsf{Reduction}(\mathcal{A})$ *is sequentially interactive,*
2. $\mathsf{Reduction}(\mathcal{A})$ *is 3ε-locally private,*
3. $\Pi^R \overset{d}{=} \Pi^B$,
4. $\mathbb{E}[N] \leq n(\frac{2e^{\varepsilon}\cdot\varepsilon}{1-e^{-\varepsilon}}k + 1)$, *and with probability $1-\beta$, $N = O(e^{\varepsilon}[nk + \sqrt{nk\log\frac{1}{\beta}}])$.*

Proof of Theorem 2. **1. Interactivity:** Since each user i's data is only used once (before s_i is set to 1), $\mathsf{Reduction}(\mathcal{A})$ is sequentially interactive.

2. Privacy: Consider a data point x corresponding to an arbitrary user over the run of $\mathsf{Reduction}(\mathcal{A})$. Then either x is drawn in line 18, or x is drawn during a rejection sampling step. In the first case, x is only used once in step 18 as input to an ε_t-local randomizer. This is ε-locally private since $\varepsilon_t \leq \varepsilon$. If x is drawn during the rejection sampling step, then it is used to simulate a draw from a 2ε-local randomizer $\tilde{R}(\cdot)$, where the input transcript

$\pi_{<t}$ has been generated ε-privately. The privacy of the input transcript is relevant because it bounds the ε-local privacy of the user's rejection sampling step. By composition and Lemma 4, this is 3ε-private.

3. Transcripts: We prove this claim by a similar argument as that of Lemma 3: we show by induction that the transcript distribution at each step t is the same for $\mathsf{Reduction}(\mathcal{A})$ and $\mathsf{BayesExpt}(\mathcal{A})$. This is trivially true at $t = 1$. Now suppose it is true through time t, i.e. $\Pi^R_{\leq t} \stackrel{d}{=} \Pi^B_{\leq t}$. Then since the joint distributions $\Pi_{\leq t+1}$ factor as $(i^{t+1}, R_{t+1}, \varepsilon_{t+1}, Y_{t+1}|\Pi_{\leq t}) \cdot \Pi_{\leq t}$, it suffices to show that the conditional distributions on $i^{t+1}, R_{t+1}, \varepsilon_{t+1}, Y_{t+1}|\Pi_{\leq t}$ coincide.

Under both $\mathsf{Reduction}(\mathcal{A})$ and $\mathsf{BayesExpt}(\mathcal{A})$, protocol $\mathcal{A}$ is used to select $i^{t+1}, R_{t+1}, \varepsilon_{t+1}$ as a function of $\Pi_{\leq t}$, so we can condition on $i^{t+1}, R_{t+1}, \varepsilon_{t+1}$ as well, and need only show that the distribution on Y_{t+1} is the same. Under $\mathsf{BayesExpt}(\mathcal{A})$, Y_{t+1} is drawn from $R_{t+1}(u, \varepsilon_{t+1}), u \sim Q_{i,t+1}$. There are two cases for $\mathsf{Reduction}(\mathcal{A})$:

- If $s_i^{t+1} = 0$, then under $\mathsf{Reduction}(\mathcal{A}), Y_{t+1}$ is drawn in line 18 from $R_{t+1}(u, \varepsilon_{t+1}), u \sim Q_{i,t+1}$, as desired.
- If $s_i^{t+1} = 1$, then $\mathsf{Reduction}(\mathcal{A})$ uses Lemma 5 to write $R_{t+1}(\cdot)$ as a mixture. Hence if we sample from the mixture with input $u \sim Q_{i,t+1}$, we sample from $R_{t+1}(u)$, which is the desired sampling distribution. To see that $\mathsf{Reduction}(\mathcal{A})$ does sample from the target, we need only show that Y_{t+1} drawn in line 13 is sampled from $\tilde{R}_t(u)$ where $u \sim Q_{i,t+1}$. This is true by Lemma 4.

4. Sample Complexity: We first bound the expected sample complexity.

Claim 3. $\mathbb{E}[N] \leq n(\frac{2e^\varepsilon \cdot \varepsilon}{1-e^{-\varepsilon}}k + 1)$.

Proof. Let N_i be the number of fresh samples drawn over all rounds t where $i^t = i$, i.e. the number of samples drawn when simulating follow-up queries to i in $\mathcal{A}$. Let N_i^t be the number of samples drawn during rejection sampling in round t (we imagine that regardless

of the coin-flip in line 11 of Reduction, N_i^t is always drawn). Then by the expected rejection sampling sample complexity from Lemma 4

$$\mathbb{E}[N_i] = \sum_{t=1}^{T} \gamma_t \mathbb{E}\left[N_i^T\right] \leq \sum_{t=1}^{T} \gamma_t 2e^{\varepsilon} = \frac{2e^{\varepsilon}}{1-e^{-\varepsilon}} \sum_{t=1}^{T} (1 - e^{-\varepsilon_t}).$$

Since $1 - x \leq e^{-x}$, we get that $1 - \varepsilon_t \leq e^{-\varepsilon_t}$ and so $1 - e^{-\varepsilon_t} \leq \varepsilon_t$. Hence

$$\mathbb{E}[N_i] \leq \frac{2e^{\varepsilon}}{1-e^{-\varepsilon}} \sum_{t=1}^{T} \varepsilon_t \leq \left(\frac{2e^{\varepsilon} \cdot \varepsilon}{1-e^{-\varepsilon}}\right) k$$

by k-compositionality. Summing over i and including the $\leq n$ samples drawn in line 18 bounds the expected sample complexity by $n((\frac{2e^{\varepsilon} \cdot \varepsilon}{1-e^{-\varepsilon}})k + 1)$. □

Next, we give the high probability sample complexity bound.

Claim 4. *With probability* $1 - \beta$, $N = O(e^{\varepsilon}[nk + \sqrt{nk \log \frac{1}{\beta}}])$.

Proof. We start with a multiplicative Azuma-Hoeffding Inequality which will drive the high probability bound. The result is folklore, but we prove it here for completeness.

Lemma 6. *Let* $\{\gamma_t\}_{t=1}^T$ *be a collection of random variables in* $[0, 1]$, *and let* $(\mathcal{F}_t)_{t=1}^T$ *be a filtration such that* $\sigma(\gamma_1, \ldots \gamma_{t-1}) \subset \mathcal{F}_{t-1}$. *Suppose there exists* $\{\mu_t\}_{t=1}^T$ *such that for every* $t \in [T]$, $\mathbb{E}\left[\gamma_t \mid \mathcal{F}_{t-1}\right] \leq \mu_t$ *where* $\sum_{t=1}^T \mu_t \leq \mu$. *Then for* $\beta \in [e^{-3\mu/4}, 1]$

$$\mathbb{P}\left[\sum_{t=1}^{T} \gamma_t > \sqrt{3\mu \log(1/\beta)} + \mu\right] \leq \beta.$$

Proof. Let $\ell \geq 0$. Then

$$\begin{aligned}
\mathbb{E}\left[e^{\ell\gamma_t} \mid \mathcal{F}_t\right] &= \mathbb{E}\left[(e^{\ell})^{\gamma_t} \mid \mathcal{F}_t\right] \\
&\leq 1 + (e^{\ell} - 1)\mathbb{E}\left[\gamma_t \mid \mathcal{F}_{t-1}\right] \\
&\leq 1 + (e^{\ell} - 1)\mu_t \\
&\leq e^{e^{\ell}-1)\mu_t} \qquad (4.1)
\end{aligned}$$

where the first inequality uses the fact $a^x \leq 1 + (a-1)x$ and the last inequality uses $1 + x \leq e^x$. Next, define $S_j = \sum_{t=1}^{j} \gamma_t$. Then

$$\begin{aligned}
\mathbb{E}\left[e^{\ell S_j}\right] &= \mathbb{E}_{\mathcal{F}_{j=1}}\left[\mathbb{E}\left[e^{\ell S_j} \mid \mathcal{F}_{j-1}\right]\right] \\
&= \mathbb{E}\left[e^{\ell\gamma_j} \mid \mathcal{F}_{j-1}\right] \cdot \mathbb{E}_{\mathcal{F}_{j-1}}\left[e^{\ell S_{j-1}} \mid \mathcal{F}_{j-1}\right] \\
&\leq \mathbb{E}\left[e^{\ell S_{j-1}}\right] e^{(e^{\ell}-1)\mu_t}
\end{aligned}$$

where the last step uses Inequality 4.1. Inducting on j then gives

$$\mathbb{E}\left[e^{\ell S_T}\right] \leq e^{(e^{\ell}-1)\sum_t \mu_t} \leq e^{(e^{\ell}-1)\mu}$$

by $\sum_{t=1}^{T} \mu_t \leq \mu$. Now for $\alpha > 0$ we take $\ell = \log(1+\alpha) > 1$ and $a = (1+\alpha)\mu$ and get

$$\begin{aligned}
\mathbb{P}\left[S \geq a\right] &= \mathbb{P}\left[e^{\ell S} \geq e^{\ell a}\right] \\
&\leq e^{-\ell a}\mathbb{E}\left[e^{\ell S}\right] \\
&\leq e^{-\ell a + (e^{\ell}-1)\mu} \\
&= e^{-\mu([1+\alpha]\log(1+\alpha)-\alpha)}
\end{aligned}$$

where the first inequality uses Markov's inequality. Then since

$$[1+\alpha]\log(1+\alpha) - \alpha \geq \alpha^2/3$$

for $\alpha \in [0, 3/2]$, we get

$$\mathbb{P}\left[S \geq (1+\alpha)\mu\right] \leq e^{-\mu\alpha^2/3}$$

and setting $\alpha = \sqrt{\frac{3\log(1/\beta)}{\mu}}$ gives the desired bound. Because $\alpha \in [0, 3/2]$, $e^{-\mu\alpha^2/3}$ is minimized at $\alpha = 3/2$, and we require $\beta \geq e^{-\mu(3/2)^2/3} = e^{-3\mu/4}$. □

We now use Lemma 6 to prove our overall claim. As in the proof for the expected sample complexity, $\leq n$ users drawn in line 18 of Reduction, so it suffices to control the sample complexity contributed by rejection sampling in line 13. For a given user i from $\mathcal{A}$, the sample complexity contributed in rounds where i is selected can be written as $\sum_{t:i_t=i}^{T} \gamma_t N_t$ where $\gamma_t \sim \mathsf{Ber}(\frac{e^{-\epsilon_t}-1}{e^{-\varepsilon}-1})$ and $N_t \overset{ind}{\sim} \mathsf{Geom}(P_t)$ for random variables $p_t \geq \frac{e^{-2\epsilon}}{2}$ and $\varepsilon_t \leq \varepsilon$ depending on the current transcript $\Pi_{<t}$. Thus we can write the total sample complexity random variable for rejection sampling rounds as

$$S = \sum_{t=1}^{T} \gamma_t N_t.$$

We now control S. First consider $\sum_{t=1}^{T} \gamma_t$. Let $\mathcal{F}_t$ be the σ-algebra generated as $\mathcal{F}_t = \sigma(\Pi_{<t}, \varepsilon_t, \{\gamma_\ell\}_{\ell=1}^{t-1})$. Then

$$\mathbb{E}\left[\gamma_t \mid \mathcal{F}_{t-1}\right] = \frac{1-e^{-\varepsilon_t}}{1-e^{-\varepsilon}}.$$

Define $\mu_t = \frac{1-e^{-\varepsilon_t}}{1-e^{-\varepsilon}}$ and $\mu = \sum_{t=1}^{T} \mu_t \leq \frac{nk\epsilon}{1-e^{-\varepsilon}}$ by $1 - e^{-x} \leq x$ and k-compositionality. Then by Lemma 6, with probability $1 - \beta/2$, for $\beta \geq 2e^{-3\mu/4}$

$$\mathbb{P}\left[\sum_{t=1}^{T} \gamma_t > \sqrt{3\mu\log(2/\delta)} + \mu\right] \leq \frac{\beta}{2}. \tag{4.2}$$

Let E_γ be the above event $\sum_{t=1}^{T} \gamma_t \leq \sqrt{3\mu\log(2/\delta)} + \mu$. Then for any a

$$\mathbb{P}\left[S \geq a \mid E_\gamma\right] \leq \mathbb{P}\left[Z \geq a\right]$$

where we define Z to be the sample complexity of $\sqrt{3\mu\log(2/\beta)}$ runs of rejection sampling,

$Z = \sum_{t=1}^{\sqrt{3\mu\log(2/\beta)}+\mu} N'_t$ for $N'_t \overset{iid}{\sim} \mathsf{Geom}(\frac{e^{-\epsilon}}{2})$. Define $\mu' = \mathbb{E}[Z] = 2e^{\epsilon}(\sqrt{3\mu\log(2/\beta)} + \mu)$. Then by a tail bound for the sum of geometric random variables (see e.g. Theorem 2.1 from Janson [45]), for any $b \geq 1$

$$\mathbb{P}\left[Z \geq b\mu'\right] \leq e^{-e^{-\varepsilon}\mu'(b-1-\log(b))/2}.$$

Setting $b = 2\left(\frac{2e^{\varepsilon}\log(2/\beta)}{\mu'} + 1\right)$ gives $\mathbb{P}\left[Z > 4\log(2/\beta)e^{\varepsilon} + \mu'\right] \leq \beta/2$, so

$$\mathbb{P}\left[S \geq 2(\log(2/\beta)2e^{\varepsilon} + \mu') \mid E_\gamma\right] \leq \frac{\beta}{2}. \tag{4.3}$$

Finally, we combine Inequalities 4.2 and 4.3 to get

$$\begin{aligned}\mathbb{P}\left[S \geq 2(\log(2/\beta)2e^{\epsilon} + \mu')\right] &\leq \mathbb{P}\left[S \geq 2(\log(2/\beta)2e^{\epsilon} + \mu') \mid E_\gamma\right]\mathbb{P}\left[E_\gamma\right] + (1 - \mathbb{P}\left[E_\gamma\right]) \\ &\leq \frac{\beta}{2} + \frac{\beta}{2} = \beta.\end{aligned}$$

We finish by substituting in

$$\begin{aligned}\mu' &= 2e^{\varepsilon}(\sqrt{3\mu\log(2/\beta)} + \mu) \\ &= O\left(e^{\varepsilon}\sqrt{\frac{\varepsilon}{1-e^{-\varepsilon}}} \cdot \sqrt{nk\log(1/\beta)} + \frac{e^{\varepsilon}\varepsilon}{1-e^{-\varepsilon}} \cdot nk\right) \\ &= \mathcal{O}\left(e^{\varepsilon}[\sqrt{nk\log(1/\beta)} + nk]\right).\end{aligned}$$

□

This completes the overall proof. □

Theorem 2 thus establishes compositionality as a relevant parameter distinguishing sequential and full interactivity. In Chapter 6, we show that this dependence is tight up to logarithmic factors.

in

Chapter 5

Polynomial Separation: Central vs. Local Privacy

We now turn from connections to separations. Henceforth, our results will be oriented toward separating central, pan-, and local privacy (and subdivisions thereof). In this chapter, we focus on separating central and local privacy. The first section shows that *simple hypothesis testing* separates central and local privacy [47]. This is the first separation between central and local privacy for a problem other than summation. In the second section, we build on this result to separate centrally and locally private one-dimensional Gaussian estimation [48].

5.1. Simple Hypothesis Testing

We start by defining simple hypothesis testing.

Definition 11. *Given known distributions P_0 and P_1 and access to i.i.d. samples from unknown distribution $P_j \in \{P_0, P_1\}$, we say algorithm $\mathcal{A}$ is a simple hypothesis tester with sample complexity n if, given $\geq n$ samples from P_j and $||P_0 - P_1||_{TV} \geq \alpha$, with probability at least $2/3$ $\mathcal{A}$ correctly identifies P_j.*

The Neyman-Pearson lemma [55] establishes that the log-likelihood ratio test is optimal for this problem absent privacy, and recent work by Canonne, Kamath, McMillan, Smith, and Ullman [23] extends this idea to give an optimal centrally private simple hypothesis test[1]. Both tests compute a variant of the log-likelihood ratio

$$\sum_{\text{samples } x} \log\left(\frac{\mathbb{P}_{P_0}[x]}{\mathbb{P}_{P_1}[x]}\right)$$

and compare it to some threshold. Intuitively, samples from P_0 should produce a positive sum and samples from P_1 should produce a negative sum. In the centrally private case,

[1] This result extends to pan-privacy as well. See Chapter 9 for details.

the algorithm receives raw samples and must satisfy differential privacy when producing its output decision. In the locally private case, the protocol must coordinate randomizer outputs from users holding samples and use those outputs to make its decision.

5.1.1. Upper Bound

We now consider a simple (folklore) noninteractive ε-locally private version of the log-likelihood test. Recall that for bit x, randomized response $\mathsf{RR}(x, \varepsilon)$ outputs x with probability $\frac{e^\varepsilon}{e^\varepsilon+1}$ and outputs $1-x$ with probability $\frac{1}{e^\varepsilon+1}$. Our tester will have each user randomized respond on which of the two distributions has higher likelihood to generate the user's datum: each user i with input x_i outputs $\mathsf{RR}\left(\arg\max_{j\in\{0,1\}} P_j(x_i), \varepsilon\right)$. The protocol processes these responses by taking the majority response as its answer.

Algorithm 5 ε-Locally Private Simple Hypothesis Tester $\mathcal{A}$

procedure NONINTERACTIVE PROTOCOL($\{x_i\}_{i=1}^n$)
  for $i = 1 \dots n$ **do**
    User i publishes $y_i \leftarrow \mathsf{RR}\left(\arg\max_{j\in\{0,1\}} P_j(x_i), \varepsilon\right)$
  Protocol computes $\hat{N}_0 \leftarrow |\{y_i \mid y_i = 0\}|$
  if $\hat{N}_0 \geq n/2$ **then**
    Protocol outputs P_0
  else
    Protocol outputs P_1

It is immediate that $\mathcal{A}$ is noninteractive and, since it relies on randomized response, satisfies ε-local privacy. We can also bound its sample complexity by simple concentration arguments.

Theorem 3. *$\mathcal{A}$ is noninteractive ε-locally private and, with probability $\geq 2/3$, distinguishes between P_0 and P_1 given $n' \geq n = O\left(\frac{1}{\alpha^2\varepsilon^2}\right)$ samples.*

Proof. Let $j^* \in \{0,1\}$ index the true distribution. Since $||P_0 - P_1||_{TV} \geq \alpha$, over the randomness of both the sample x and the randomizer,

$$\mathbb{P}\left[\mathsf{RR}\left(\arg\max_{j\in\{0,1\}} P_j(x), \varepsilon\right) = j^*\right] \geq \frac{1}{2} + \Omega(\alpha\varepsilon).$$

This is equivalent to distinguishing Bernoulli distributions with parameters separated by $\Omega(\alpha\varepsilon)$, so $n' = \Omega\left(\frac{1}{\alpha^2\varepsilon^2}\right)$ samples suffice. □

In fact, we can show that the simple tester given above is optimal even among the much larger class of fully interactive (ϵ, δ)-locally private tests.

5.1.2. Lower Bound

We prove a general lower bound that applies to both pure and approximate locally private protocols. We start by revisiting the distinction between these two settings. In central privacy, approximate privacy sometimes offers large utility improvements over pure privacy [44]. No such separation is known for local privacy. In fact, we can show that the two models are essentially equivalent for sequentially interactive protocols.

First, combining (slightly modified versions of) Theorem 6.1 from Bun, Nelson, and Stemmer [19] and Theorem A.1 from Cheu, Smith, Ullman, Zeber, and Zhilyaev [27], we get the following result[2]

Lemma 7. *Given* $\delta < \min\left(\frac{\epsilon\beta}{48n\ln(2n/\beta)}, \frac{\beta}{64n\ln(n/\beta)e^{7\varepsilon}}\right)$ *and sequentially interactive* (ε, δ)*-locally private protocol* $\mathcal{A}$*, there exists a sequentially interactive* $(10\varepsilon, 0)$*-locally private protocol* $\mathcal{A}'$ *such that for any dataset* U*,* $||\mathcal{A}(U) - \mathcal{A}'(U)||_{TV} \leq \beta$*.*

Lemma 7 enables us to apply existing lower bound tools for ε-locally private protocols to sequentially interactive (ε, δ)-locally private protocols. With this result in hand, we now sketch the proof of our lower bound for locally private simple hypothesis testing.

Our proof relies on controlling the squared Hellinger distance between transcript distributions induced by an (ε, δ)-locally private protocol when samples are generated by P_0 and

[2] Bun et al. [19] and Cheu et al. [27] prove their results for noninteractive protocols. However, their constructions both rely on replacing a single (ε, δ)-local randomizer call for each user with an $(O(\varepsilon), 0)$-local randomizer call and proving that these randomizers induce similar output distributions. Since each user still makes a single randomizer call in sequential interactive protocols, essentially the same argument applies. For fully interactive protocols, a naive modification of the same result forces a stronger restriction on δ in terms of the number of the maximum number of randomizer calls to any one user T, roughly $\delta = \tilde{o}\left(\frac{\varepsilon\beta}{\max(n,T)}\right)$.

P_1. Once we show that these distributions are "close together" in a way that depends on ε, δ, and n, we can show that distinguishing the two settings forces a lower bound on n.

More specifically, we borrow a simulation technique used by Braverman, Garg, Ma, Nguyen, and Woodruff [18] for a similar (non-private) problem and find that we can control this squared Hellinger distance by bounding the KL divergence between a simpler, *noninteractive* pair of transcript distributions. This transformation is important because it lets us employ existing lower bound tools for noninteractive protocols [32].

Since our proofs rely on the squared Hellinger distance and KL divergence, we first recall their definitions.

Definition 12. *Let f and g be distributions over $\mathcal{X}$. The* squared Hellinger distance *between f and g is $H^2(f,g) = 1 - \int_{\mathcal{X}} \sqrt{f(x)g(x)}dx$.*

Definition 13. *Let f and g be distributions over $\mathcal{X}$. The* KL divergence *between f and g is $D_{KL}(f||g) = \int_{\mathcal{X}} f(x) \log\left(\frac{f(x)}{g(x)}\right) dx$.*

Next, some facts relating the two notions and total variation distance will be useful.

Fact 4. *For any distributions f, g, and h,*

1. $H^2(f,g) \leq 2(H^2(f,h) + H^2(h,g))$.
2. $H^2(f,g) \leq ||f-g||_{TV}$.
3. $H^2(f,g) \leq \frac{1}{2} D_{KL}(f||g)$.
4. $||f-g||_{TV}^2 \leq 2H^2(f,g)$.

We also recall Pinsker's inequality.

Fact 5. *For distributions f and g,*

$$||f-g||_{TV} \leq \sqrt{\frac{D_{KL}(f||g)}{2}}.$$

We now have most of the tools necessary for our locally private simple hypothesis testing lower bound.

Theorem 4. *Let* $||P_0 - P_1||_{TV} = \alpha$ *and let* $\mathcal{A}$ *be an* (ε, δ)*-locally private simple hypothesis testing protocol distinguishing between* P_0 *and* P_1 *with probability* $\geq 2/3$ *using* n *samples where* $\varepsilon = O(1)$ *and* $\delta < \min\left(\frac{\varepsilon^3\alpha^2}{48n\ln(2n/\beta)}, \frac{\varepsilon^2\alpha^2}{64n\ln(n/\beta)e^{7\varepsilon}}\right)$. *Then* $n = \Omega\left(\frac{1}{\alpha^2\varepsilon^2}\right)$.

Proof. Let $\Pi_{\vec{0}}, \Pi_{\vec{1}}$, and $\Pi_{\vec{e}_i}$ respectively denote the distributions over transcripts induced by protocol $\mathcal{A}$ when samples are drawn from P_0, P_1, and $x_i \sim P_1$ but the remaining $x_{i'} \sim P_0$. Our goal is to upper bound $H^2(\Pi_{\vec{0}}, \Pi_{\vec{1}})$. We begin with Lemma 8, originally proven as Lemma 2 by Braverman et al. [18].

Lemma 8. $H^2(\Pi_{\bar{0}}, \Pi_{\bar{1}}) = O\left(\sum_{i=1}^n H^2(\Pi_{\bar{0}}, \Pi_{\vec{e}_i})\right)$.

By Lemma 8, to bound $H^2(\Pi_{\vec{0}}, \Pi_{\vec{1}})$ it now suffices to bound each $H^2(\Pi_{\bar{0}}, \Pi_{\vec{e}_i})$.

Fix one such term i. Consider the following protocol: user i simulates $\mathcal{A}$ using draws from P_0 for the inputs of other users and their input x_i for input i. Since $\mathcal{A}$ is an (ε, δ)-locally private protocol, this simulation can be viewed as a single (ε, δ)-local randomizer applied to x_i. We can therefore use the approximate-to-pure transformation of Lemma 7 to get a $(10\varepsilon, 0)$-local randomizer $\mathcal{A}'$ inducing distributions $\Pi'_{\vec{0}}$ and $\Pi'_{\vec{e}_i}$ such that $||\Pi'_{\vec{0}} - \Pi_{\vec{0}}||_{TV} \leq \alpha^2\varepsilon^2$ and $||\Pi'_{\vec{e}_i} - \Pi_{\vec{e}_i}||_{TV} \leq \alpha^2\varepsilon^2$. We now upper bound $H^2(\Pi_{\vec{0}}, \Pi_{\vec{e}_i})$ in terms of α, ε, and $H^2(\Pi'_{\vec{0}}, \Pi'_{\vec{e}_i})$:

$$
\begin{aligned}
H^2(\Pi_{\vec{0}}, \Pi_{\vec{e}_i}) &\leq 2(H^2(\Pi_{\vec{0}}, \Pi'_{\vec{e}_i}) + H^2(\Pi'_{\vec{e}_i}, \Pi_{\vec{e}_i})) && \text{(Fact 4, item 1)} \\
&\leq 4(H^2(\Pi_{\vec{0}}, \Pi'_{\vec{0}}) + H^2(\Pi'_{\vec{0}}, \Pi'_{\vec{e}_i})) + 2H^2(\Pi'_{\vec{e}_i}, \Pi_{\vec{e}_i}) && \text{(Fact 4, item 1)} \\
&\leq 4(||\Pi_{\vec{0}} - \Pi'_{\vec{0}}||_{TV} + H^2(\Pi'_{\vec{0}}, \Pi'_{\vec{e}_i})) + 2||\Pi'_{\vec{e}_i} - \Pi_{\vec{e}_i}||_{TV} && \text{(Fact 4, item 2)} \\
&\leq 6\alpha^2\varepsilon^2 + 4H^2(\Pi'_{\vec{0}}, \Pi'_{\vec{e}_i})
\end{aligned}
$$

where the last inequality uses our upper bounds on $||\Pi'_{\vec{0}} - \Pi_{\vec{0}}||_{TV}$ and $||\Pi'_{\vec{e}_i} - \Pi_{\vec{e}_i}||_{TV}$.

It now remains to bound $H^2(\Pi'_{\vec{0}}, \Pi'_{\vec{e}_i})$. By Fact 4, item 3, $4H^2(\Pi'_{\vec{0}}, \Pi'_{\vec{e}_i}) \leq 2D_{KL}\left(\Pi'_{\vec{0}} || \Pi'_{\vec{e}_i}\right)$. As noted previously, the transcript distributions $\Pi'_{\vec{0}}$ and $\Pi'_{\vec{e}_i}$ can be simulated by noninteractive $(10\varepsilon, 0)$-local randomizers. We can therefore apply Theorem 1 from Duchi et al. [32], restated for our setting as Lemma 9.

Lemma 9. *Let Q be an ε-randomizer and let P_0 and P_1 be distributions on $\mathcal{X}$. Let $x_0 \sim P_0$ and $x_1 \sim P_1$. Then*

$$D_{KL}\left(Q(x_0)||Q(x_1)\right) + D_{KL}\left(Q(x_1)||Q(x_0)\right) \leq \min(4, e^{2\varepsilon}) \cdot (e^{\varepsilon} - 1)^2 ||P_0 - P_1||_{TV}^2.$$

Thus

$$D_{KL}\left(\Pi'_{\vec{0}} || \Pi'_{\vec{e}_i}\right) + D_{KL}\left(\Pi'_{\vec{e}_i} || \Pi'_{\vec{0}}\right) = O(\varepsilon^2 \cdot ||P_0 - P_1||_{TV}^2) = O\left(\varepsilon^2 \alpha^2\right).$$

It follows that $H^2(\Pi'_{\vec{0}}, \Pi'_{\vec{e}_i}) = O(\alpha^2 \varepsilon^2)$. Moreover, since our original choice of i was arbitrary, tracing back to Lemma 8 yields $H^2(\Pi_{\vec{0}}, \Pi_{\vec{1}}) = O(\alpha^2 \varepsilon^2 n)$. By Fact 4 item 4, $H^2(\Pi_{\vec{0}}, \Pi_{\vec{1}}) \geq \frac{1}{2}||\Pi_{\vec{0}} - \Pi_{\vec{1}}||_{TV}^2 = \Omega(1)$ since $\mathcal{A}$ distinguishes between P_0 and P_1 with $\Omega(1)$ probability. Thus $\alpha^2 \varepsilon^2 n = \Omega(1)$, so $n = \Omega\left(\frac{1}{\alpha^2 \varepsilon^2}\right)$. □

For comparison, as observed by Canonne et al. [23], it is easy to obtain a superior centrally private simple hypothesis tester using the subsample-and-aggregate framework [56]. By folklore, $m = \Theta\left(\frac{1}{H^2(P_0, P_1)}\right)$ samples are necessary and sufficient to test absent privacy. We can therefore repeat this non-private test $O\left(\frac{1}{\varepsilon}\right)$ times on disjoint data and output the result of a random test. This ensures ε-central privacy because, while each tester is not necessarily private, each user has a limited chance of contributing data to the final result. This requires $m = O\left(\frac{1}{\varepsilon H^2(P_0, P_1)}\right)$ samples and obtains the same accuracy as the non-private tester.

In contrast, by Fact 4 item 4, Theorem 4 implies that any such approximate locally private hypothesis tester requires $m = \Omega\left(\frac{1}{\varepsilon^2 H^2(P_0, P_1)}\right)$ samples. This separates central and local privacy[3].

[3]The true separation is actually even larger, because the subsample-and-aggregate approach is suboptimal [23]. However, the optimal central guarantee is more involved, so we omit it.

Moreover, we can also extend the reasoning above to (a restricted form of) *compound* hypothesis testing. Here P_0 and P_1 are replaced by (disjoint) collections of discrete hypotheses H_0 and H_1 such that

$$\inf_{(P,Q)\in H_0\times H_1} ||P-Q||_{TV} \geq \alpha.$$

The goal is to determine whether samples are generated by a distribution in H_0 or one in H_1.

Theorem 5. *Let H_0 and H_1 be convex and compact sets of distributions over ground set X such that $\inf_{(P,Q)\in H_0\times H_1} ||P-Q||_{TV} \geq \alpha$. Then there exists noninteractive ε-locally private protocol $\mathcal{A}$ that with probability at least $2/3$ distinguishes between H_0 and H_1 given $n = \Omega\left(\frac{1}{\alpha^2\varepsilon^2}\right)$ samples.*

Proof. Let X be the ground set for distributions in H_0 and H_1, and consider the two-player zero-sum game

$$\sup_{S\in\Delta(2^X)} \inf_{(P,Q)\in H_0\times H_1} \mathbb{E}_{E\sim S}\left[P(E)-Q(E)\right].$$

Here, the sup player chooses a distribution over events, and the inf player chooses distributions in H_0 and H_1. We will use (a simplified version of) Sion's minimax theorem [60].

Lemma 10 (Sion's minimax theorem)**.** *For $f : A \times B \to \mathbb{R}$, if*

1. *for all $a \in A$ $f(a,\cdot)$ is continuous and concave on B,*
2. *for all $b \in B$ $f(\cdot,b)$ is continuous and convex on A, and*
3. *A and B are convex and A is compact,*

then

$$\sup_{b\in B}\inf_{a\in A} f(a,b) = \inf_{a\in A}\sup_{b\in B} f(a,b).$$

We first verify that the three conditions of Lemma 10 hold. Let

$$f(S,(P,Q)) = \mathbb{E}_{E\sim S}\left[P(E) - Q(E)\right].$$

Linearity of expectation implies that $f(\cdot,(P,Q))$ is linear in $\Delta(2^X)$ and $f(S,\cdot)$ is linear in $H_0 \times H_1$. Therefore conditions 1 and 2 hold. Moreover, since $\Delta(2^X)$ is convex and we assumed H_0 and H_1 to be convex and compact — properties which are both closed under Cartesian product — condition 3 holds as well. As a result,

$$\sup_{S\in\Delta(2^X)} \inf_{(P,Q)\in H_0\times H_1} \mathbb{E}_{E\sim S}\left[P(E) - Q(E)\right] = \inf_{(P,Q)\in H_0\times H_1} \sup_{S\in\Delta(2^X)} \mathbb{E}_{E\sim S}\left[P(E) - Q(E)\right] \geq \alpha$$

and there exists fixed distribution S over events such that for all $(P,Q) \in H_0 \times H_1$,

$$\mathbb{E}_{E\sim S}\left[P(E) - Q(E)\right] \geq \alpha.$$

This leads to the following hypothesis testing protocol $\mathcal{A}$: for each $i \in [n]$, user i computes $y_i = \mathbb{E}_{E\sim S}\left[\mathbb{1}\left[x_i \in E\right]\right]$ and publishes $y_i + \mathsf{Lap}\left(\frac{1}{\varepsilon}\right)$. This protocol is immediately noninteractive, and since $y_i \in [0,1]$, this protocol is $(\varepsilon,0)$-locally private over $\{x_i\}_{i=1}^n$. Finally, by the same analysis used to prove Theorem 3 (replacing concentration of randomized responses with concentration of $\mathsf{Lap}\left(1\right)$ noise [25]) it distinguishes between H_0 and H_1 with probability at least $2/3$ using $n = \Omega\left(\frac{1}{\varepsilon^2\alpha^2}\right)$ samples. □

Since Theorem 4 still applies, this establishes that the above noninteractive protocol is also optimal.

5.2. One-Dimensional Gaussian Estimation

We can also use Theorem 4 to separate centrally and locally private one-dimensional Gaussian estimation. Here, the goal is to use i.i.d. samples $x_1, \ldots x_n \sim N(\mu, \sigma^2)$ to estimate μ and σ^2 while guaranteeing local privacy.

5.2.1. Lower Bound

For central privacy, Karwa and Vadhan [49] gave an algorithm that with probability $1-\beta$ obtains accuracy

$$O\left(\sigma\sqrt{\frac{\log(1/\beta)}{n}}+\frac{\operatorname{poly}\log(1/\beta)}{\varepsilon n}\right).$$

Note that the first term is necessary even without privacy, while the second privacy term has an $O(1/n)$ dependence. In contrast, we can use Theorem 4 to show that locally private learners must suffer an $\Omega(1/\sqrt{n})$ dependence.

Theorem 6. *For a given σ and δ as in Theorem 4, there does not exist an (ε,δ)-locally private protocol $\mathcal{A}$ such that for every $\mu = O\left(\frac{\sigma}{\varepsilon}\sqrt{\frac{1}{n}}\right)$, given $x_1,\ldots,x_n \sim N(\mu,\sigma^2)$, $\mathcal{A}$ outputs estimate $\hat{\mu}$ satisfying $|\hat{\mu}-\mu| = o\left(\frac{\sigma}{\varepsilon}\sqrt{\frac{1}{n}}\right)$ with probability $\geq 15/16$.*

Proof. Let $P_0 = N(0,\sigma^2)$ and $P_1 = N(M,\sigma^2)$. Then since

$$D_{KL}(N(\mu_1,\sigma^2)||N(\mu_2,\sigma^2)) \leq \left(\tfrac{\mu_1-\mu_2}{\sigma}\right)^2$$

we get $D_{KL}(P_0||P_1) = O\left(\frac{M^2}{\sigma^2}\right)$. Pinsker's inequality (Fact 5) then implies $||P_0-P_1||_{TV}^2 = O\left(\frac{M^2}{\sigma^2}\right)$. Substituting this into Theorem 4, we get that distinguishing P_0 and P_1 with constant probability and n samples requires $n = \Omega\left(\frac{\sigma^2}{\varepsilon^2 M^2}\right)$, so $M = \Omega\left(\frac{\sigma}{\varepsilon\sqrt{n}}\right)$. □

In simultaneous independent work, Gaboardi, Rogers, and Sheffet [39] gave a stronger lower bound for the same problem that incorporates the $\log(1/\beta)$ failure probability dependence. However, their result does not extend to fully interactive protocols.

5.2.2. Upper Bound

We now spend the remainder of this section showing that the lower bound of Theorem 6 is tight up to logarithmic factors. For brevity, we only consider the case where μ is unknown, σ is known, and the protocol must is sequentially interactive. Gaboardi et al. [39] gave similar upper bounds, albeit with much higher round complexity. A concise comparison of

	Gaboardi et al. [39]	This Work
Setting	Accuracy α, Round Complexity T	Accuracy α, Round Complexity T
Known σ, adaptive	$\alpha = O\left(\frac{\sigma}{\varepsilon}\sqrt{\frac{\log\left(\frac{1}{\beta}\right)\log\left(\frac{n}{\beta}\right)\log\left(\frac{1}{\delta}\right)}{n}}\right)$ $T = 2$	$\alpha = O\left(\frac{\sigma}{\varepsilon}\sqrt{\frac{\log\left(\frac{1}{\beta}\right)}{n}}\right)$ $T = 2$
Known σ, nonadaptive	–	$\alpha = O\left(\frac{\sigma}{\varepsilon}\sqrt{\frac{\log\left(\frac{1}{\beta}\right)\sqrt{\log(n)}}{n}}\right)$ $T = 1$
Unknown σ, adaptive	$\alpha = O\left(\frac{\sigma}{\varepsilon}\sqrt{\frac{\log\left(\frac{1}{\beta}\right)\log\left(\frac{n}{\beta}\right)\log\left(\frac{1}{\delta}\right)}{n}}\right)$ $T = \Omega\left(\log\left(\frac{R}{\sigma_{\min}}\right)\right)$	$\alpha = O\left(\frac{\sigma}{\varepsilon}\sqrt{\frac{\log\left(\frac{1}{\beta}\right)\log(n)}{n}}\right)$ $T = 2$
Unknown σ, non-adaptive	–	$\alpha = O\left(\frac{\sigma}{\varepsilon}\sqrt{\frac{\log\left(\frac{\sigma_{\max}}{\sigma_{\min}}+1\right)\log\left(\frac{1}{\beta}\right)\log^{3/2}(n)}{n}}\right)$ $T = 1$

Table 1: A comparison of upper bounds in Gaboardi et al. [39] and here. In all cases, Gaboardi et al. [39] use (ε, δ)-locally private algorithms and we use $(\varepsilon, 0)$. Here, R denotes an upper bound on both μ and σ. In our setting, the upper bound on μ is $O(2^{n\varepsilon^2/\log(n/\beta)})$, leading the unknown variance protocol of Gaboardi et al. [39] to round complexity potentially as large as $\tilde{\Omega}(n\varepsilon^2/\log(1/\beta))$.

our results to those of Gaboardi et al. [39] appears in Table 1. Details for the other cases displayed appear in the full paper [46].

We start by describing our protocol KVGausstimate. Throughout, we use phrases like "the protocol computes". Technically, this should be viewed as the protocol processing the transcript to assign more randomizers or halt as specified in Definition 6. However, we treat "the protocol" as an entity coordinating users for simplicity.

First, KVGausstimate splits users into halves U_1 and U_2. In round one, the protocol queries users in U_1 to obtain an $O(\sigma)$-accurate estimate $\hat{\mu}_1$ of μ. In round two, the protocol

passes $\hat{\mu}_1$ to users in U_2, who respond based on $\hat{\mu}_1$ and their own data. The protocol then aggregates this second set of responses into a better final estimate of μ.

procedure KVGausstimate($\varepsilon, k, \mathcal{L}, n, \sigma, U_1, U_2$)
  for $j \in \mathcal{L}$ **do**
    for user $i \in U_1^j$ **do**
      User i outputs $\tilde{y}_i \leftarrow \text{RR1}(\varepsilon, i, j)$
  Protocol computes $\hat{H}_1 \leftarrow \text{KVAgg1}(\varepsilon, k, \mathcal{L}, U_1)$
  Protocol computes $\hat{\mu}_1 \leftarrow \text{EstMean}(\beta, \varepsilon, \hat{H}_1, k, \mathcal{L})$
  for user $i \in U_2$ **do**
    User i outputs $\tilde{y}_i \leftarrow \text{KVRR2}(\varepsilon, i, \hat{\mu}_1, \sigma)$
  Protocol computes $\hat{H}_2 \leftarrow \text{KVAgg2}(\varepsilon, n/2, U_2)$
  Protocol computes $\hat{T} \leftarrow \sqrt{2} \cdot \text{erf}^{-1}\left(\frac{2(-\hat{H}_2(-1)+\hat{H}_2(1))}{n}\right)$
  Protocol outputs $\hat{\mu}_2 \leftarrow \sigma\hat{T} + \hat{\mu}_1$

First round of KVGausstimate

For neatness, let $L = \lfloor n/(2k) \rfloor$, $L_{\min} = \lfloor \log(\sigma) \rfloor$, $L_{\max} = L_{\min} - 1 + L$, and $\mathcal{L} = \{L_{\min}, L_{\min} + 1, \ldots, L_{\max}\}$. KVGausstimate splits U_1 into L subgroups indexed by $\mathcal{L}$ where each subgroup has size $k = \Omega\left(\frac{\log(n/\beta)}{\varepsilon^2}\right)$. The protocol begins by iterating through each subgroup $j \in \mathcal{L}$. Each user $i \in U_1^j$ releases a privatized version of $\lfloor x_i/2^j \rfloor \bmod 4$ using a simple variant of randomized response (RR1): with probability $e^\varepsilon/(e^\varepsilon + 3)$, user i outputs $\lfloor x_i/2^j \rfloor \bmod 4$, and otherwise outputs one of the remaining elements of $\{0, 1, 2, 3\}$ uniformly at random.

KVGausstimate uses responses from group U_1^j to estimate the j^{th} least significant bit of μ (rounded to an integer). It then uses "Known Variance Aggregation" (KVAgg1) to aggregate and debias responses to account for added randomness.

procedure KVAgg1($\varepsilon, k, \mathcal{L}, U$)
  for $j \in \mathcal{L}$ **do**
    for $a \in \{0, 1\}$ **do**
      $C^j(a) \leftarrow |\{\tilde{y}_i \mid i \in U^j, \tilde{y}_i = a\}|$
      $\hat{H}^j(a) \leftarrow \frac{e^\varepsilon+3}{e^\varepsilon-1} \cdot \left(C^j(a) - \frac{k}{e^\varepsilon+3}\right)$
  Output $\hat{H}$

The result is a collection of histograms $\hat{H}_1$. KVGausstimate uses $\hat{H}_1$ to binary search for μ (EstMean). Intuitively, for each subgroup U_1^j, if all multiples of 2^j are far from μ then Gaussian concentration implies that almost all users $i \in U_1^j$ compute the same value of $\lfloor x/2^j \rfloor \bmod 4$. This produces a histogram $\hat{H}_1^j$ where most elements concentrate in a single bin. The protocol in turn narrows its search range for μ. For example, if $\hat{H}_1^{L_{\max}}$ concentrates in 0, then the range narrows to $\mu \in [0, 2^{L_{\max}})$; if $\hat{H}_1^{L_{\max}-1}$ concentrates in 1, then the range narrows to $\mu \in [2^{L_{\max}-1}, 2^{L_{\max}})$, and so on.

If instead some multiple of 2^j is near μ, the elements of $\hat{H}_1^j$ will spread over multiple (adjacent) bins. This is also useful: a point from the "middle" of this block of bins is $O(\sigma)$-close to μ. The protocol thus takes such a point as $\hat{\mu}_1$ and ends its search. Our analysis will also rely on having a bin with a noticeably low count that is not adjacent to the bin containing μ. This motivates using 4 as a modulus.

---

procedure EstMean($\beta, \varepsilon, \hat{H}_1, k, \mathcal{L}$)
  $\psi \leftarrow \left(\frac{\varepsilon+4}{\varepsilon\sqrt{2}}\right) \cdot \sqrt{k \ln(8L/\beta)}$
  $j \leftarrow L_{\max}$
  $I_j \leftarrow [0, 2^{L_{\max}}]$
  while $j \geq L_{\min}$ and $\max_{a \in \{0,1,2,3\}} \hat{H}_1^j(a) \geq 0.52k + \psi$ **do**
    Protocol computes integer c such that $c2^j \in I_j$ and $c \equiv M_1(j) \bmod 4$
    Protocol computes $I_{j-1} \leftarrow [c2^j, (c+1)2^j]$
    $j \leftarrow j - 1$
  $j \leftarrow \max(j, L_{\min})$
  Protocol computes $M_1(j) \leftarrow \arg\max_{a \in \{0,1,2,3\}} \hat{H}_1^j(a)$
  Protocol computes $M_2(j) \leftarrow \arg\max_{a \in \{0,1,2,3\} - \{M_1(j)\}} \hat{H}_1^j(a)$
  Protocol computes $c^* \leftarrow$ maximum integer such that $c^*2^j \in I_j$ and $c^* \equiv M_1(j)$ or $M_2(j) \bmod 4$
  protocol outputs $\hat{\mu}_1 \leftarrow c^*2^j$

KVGausstimate therefore uses EstMean to examine $\hat{H}_1^{L_{\max}}, \hat{H}_1^{L_{\max}-1}, \ldots$ in sequence, estimating μ from most to least significant bit. Crucially, the modulus structure of user responses enables the protocol to carry out this binary search with *one* round of interaction. Thus the first round of the protocol concludes with an $O(\sigma)$-accurate estimate $\hat{\mu}_1$ of μ.

Second round of KVGausstimate

In the second round, the KVGausstimate passes $\hat{\mu}_1$ to users in U_2. Users respond through 'Known Variance Randomized Response" (KVRR2), a variant of randomized response based on an algorithm from the distributed statistical estimation literature [18]. In KVRR2, each user i centers their point x_i with $\hat{\mu}_1$, standardizes it using σ, and randomized responds by outputting $\mathsf{RR}\left(\mathsf{sgn}\left((x_i - \hat{\mu}_1)/\sigma\right), \varepsilon\right)$. The protocol aggregates these responses by a debiasing process KVAgg2 akin to KVAgg1.

1: **procedure** KVAgg2(ε, k, U)
2: **for** $a \in \{-1, 1\}$ **do**
3: $C(a) \leftarrow |\{\tilde{y}_i \mid i \in U, \tilde{y}_i = a\}|$
4: $\hat{H}(a) \leftarrow \frac{e^\varepsilon + 1}{e^\varepsilon - 1} \cdot \left(C(a) - \frac{k}{e^\varepsilon + 1}\right)$
5: Protocol outputs $\hat{H}$

From this aggregation $\hat{H}_2$, the protocol obtains a good estimate of the bias of the initial estimate $\hat{\mu}_1$. If $\hat{\mu}_1 < \mu$, responses will skew toward 1, and if $\hat{\mu}_1 > \mu$ responses will skew toward -1. By comparing this skew to the true standard CDF using the error function erf, the protocol recovers a better final estimate $\hat{\mu}_2$ of μ (Lines 12-13 of KVGausstimate).

We now state the full guarantee for KVGausstimate.

Theorem 7. KVGausstimate *is sequentially interactive ε-locally private and, given users with data $x_1, \ldots, x_n \sim_{i.i.d.} N(\mu, \sigma^2)$ where σ is known and $\frac{n}{\log(n)} = \Omega\left(\frac{\log(\mu)\log(1/\beta)}{\varepsilon^2}\right)$, with probability $1 - \beta$ outputs $\hat{\mu}$ such that $|\hat{\mu} - \mu| = O\left(\frac{\sigma}{\varepsilon}\sqrt{\frac{\log(1/\beta)}{n}}\right)$.*

Proof. Privacy: Since each user produces exactly one output, using either RR1 or KVRR2, it suffices to note that both RR1 and KVRR2 are ε-randomizers.

Utility: First, recall that $\hat{H}_1$ is the aggregation (via KVAgg1) of user responses (via RR1). Let H_1 be the "true" histogram, $H_1^j(a) = |\{y_i \mid i \in U_1^j, y_i = a\}|$ for all $a \in \{0, 1, 2, 3\}$ and $j \in \mathcal{L}$. Since the protocol only has access to $\hat{H}_1$, we need to show that $\hat{H}_1$ and H_1 are similar.

Lemma 11. *With probability at least $1-\beta$, for all $j \in \mathcal{L}$,*

$$||\hat{H}_1^j - H_1^j||_\infty \leq \left(\tfrac{\varepsilon+4}{\varepsilon\sqrt{2}}\right) \cdot \sqrt{k \ln(8L/\beta)}.$$

Proof. Choose $a \in \{0,1,2,3\}$ and $j \in \mathcal{L}$. $\mathbb{E}\left[C^j(a)\right] = \frac{H_1^j(a)e^\varepsilon}{e^\varepsilon+3} + \frac{k-H_1^j(a)}{e^\varepsilon+3} = \frac{H_1^j(a)(e^\varepsilon-1)+k}{e^\varepsilon+3}$, so by a pair of Chernoff bounds on the k users in U_1^j, with probability at least $1-\beta/4L$,

$$|C^j(a) - \tfrac{H_1^j(a)(e^\varepsilon-1)+k}{e^\varepsilon+3}| \leq \sqrt{k \ln(8L/\beta)/2}.$$

Then since $\hat{H}_1^j(a) = \frac{e^\varepsilon+3}{e^\varepsilon-1} \cdot \left(C^j(a) - \frac{k}{e^\varepsilon+3}\right)$, this implies

$$|\hat{H}_1^j(a) - H_1^j(a)| \leq \frac{e^\varepsilon+3}{e^\varepsilon-1} \cdot \sqrt{k \ln(8L/\beta)/2} < \left(\tfrac{\varepsilon+4}{\varepsilon\sqrt{2}}\right) \cdot \sqrt{k \ln(8L/\beta)}$$

where the last step uses $\frac{e^\varepsilon+3}{e^\varepsilon-1} < \frac{\varepsilon+4}{\varepsilon}$. Union bounding over $a \in \{0,1,2,3\}$ and all L groups U_1^j completes the proof. □

Next, we show how the protocol uses $\hat{H}_1$ to estimate μ through EstMean. Intuitively, in subgroup U_1^j when user responses concentrate in a single bin mod 4, this suggests that μ lies in the corresponding bin. In the other direction, when user responses do not concentrate in a single bin, users with points near μ must spread out over multiple bins, suggesting that μ lies near the boundary between bins. We formalize this intuition in EstMean and Lemma 12.

Lemma 12. *Conditioned on the success of the preceding lemmas, with probability at least $1-\beta$, $|\hat{\mu}_1 - \mu| \leq 2\sigma$.*

Proof. Recall the definitions of ψ, $M_1(j)$, and $M_2(j)$ from the pseudocode for EstMean:

$$\psi = \left(\tfrac{\varepsilon+4}{\varepsilon\sqrt{2}}\right) \cdot \sqrt{k \ln(8L/\beta)},$$

$$M_1(j) = \arg\max_{a \in \{0,1,2,3\}} \hat{H}_1^j(a),$$

$$M_2(j) = \arg \max_{a \in \{0,1,2,3\} - \{M_1(j)\}} \hat{H}_1^j(a).$$

We start by proving two useful claims.

Claim 1: With probability at least $1 - \beta/5$, for all $j \in \mathcal{L}$ where $2^j > \sigma$, if $j' = L_{\max}, L_{\max} - 1, \ldots, j+1$ all have $\hat{H}_1^{j'}(M_1(j)) \geq 0.52k + \psi$, then $\mu \in I_j$.

To see why, suppose $2^j > \sigma$ and let $x \sim N(\mu, \sigma^2)$. Recall the Gaussian CDF $F(x) = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right]$. Then for any $a \not\equiv \lfloor \mu/2^j \rfloor \bmod 4$

$$\mathbb{P}\left[\lfloor x/2^j \rfloor \equiv a \bmod 4\right] \leq \mathbb{P}\left[x \notin [\mu, \mu + 3 \cdot 2^j)\right] < \mathbb{P}\left[x \notin [\mu, \mu + 3\sigma)\right] < 0.51$$

where the second inequality uses $2^j > \sigma$. Thus by a binomial Chernoff bound, the assumption $k > 5000 \ln(5L/\beta)$, and Lemma 11, with probability $\geq 1 - \beta/5L$, $\hat{H}_1^j(a) < 0.52k + \psi$. Therefore if for some a we have $\hat{H}_1^j(a) \geq 0.52k + \psi$, $a \equiv \lfloor \mu/2^j \rfloor \bmod 4$. Moreover, if $\mu \in I_j$ then letting c be the (unique) integer such that $c \equiv M_1(j) \bmod 4$ and $c2^j \in I_j$ (since I_j has endpoints $c_1 2^j$ and $(c_1 + 2)2^j$ for integer c_1) we get $\mu \in [c2^j, (c+1)2^j] = I_j$. As $\mu \in I_{L_{\max}}$ by our assumed lower bound on n, the claim follows by induction.

Claim 2: Let j be the maximum $j \in \mathcal{L}$ with $\hat{H}_1^j(M_1(j)) < 0.52k + \psi$, and let c^* be the maximum integer such that $c^* 2^j \in I_j$ and $c^* \equiv M_1(j)$ or $M_2(j) \bmod 4$. If $2^j > \sigma$, then with probability at least $1 - 4\beta/5$, $|c^* 2^j - \mu| \leq 2\sigma$.

To see why, first note that by Claim 1, $\mu \in I_j$. Let $[c2^j, (c+1)2^j)$ be the subinterval of I_j containing μ for integer c. Then as $2^j > \sigma$, for $x \sim N(\mu, \sigma^2)$, by another application of the Gaussian CDF,

$$\mathbb{P}\left[x \in [c2^j, (c+1)2^j)\right] > \mathbb{P}\left[x \in [\mu, \mu + \sigma)\right] \geq 0.34.$$

Thus by the same method as above, using the assumption $k > 5000 \ln(5/\beta)$, with probability

at least $1-\beta/5$, $\hat{H}_1^j(c \bmod 4) \geq 0.33k - \psi$. By similar logic, since

$$\begin{aligned}\mathbb{P}\left[\lfloor x/2^j \rfloor \equiv c+2 \bmod 4\right] &< \max_{\lambda \in [0,2^j]} \mathbb{P}\left[x \notin [\mu - 2^j - \lambda, \mu + 2 \cdot 2^j - \lambda]\right] \\ &< \mathbb{P}\left[x \notin [\mu - \sigma, \mu + 2\sigma)\right] \leq 0.19\end{aligned}$$

with probability at least $1-\beta/5$, $\hat{H}_1^j(c+2 \bmod 4) \leq 0.2k + \psi$. Next, consider $\hat{H}_1^j(c-1 \bmod 4)$. If $\mu \geq (c+0.75)2^j$, then

$$\mathbb{P}\left[x \in [(c-1)2^j, c2^j)\right] \leq \mathbb{P}\left[x \notin [\mu - 3\sigma/4, \mu + 9\sigma/4]\right] \leq 0.24$$

so with probability at least $1-\beta/5$

$$\hat{H}_1^j(c-1 \bmod 4) \leq 0.25k + \psi < 0.33k - \psi \leq \hat{H}_1^j(c \bmod 4)$$

where the middle inequality uses $k > 625\left(\frac{\varepsilon+4}{\varepsilon\sqrt{2}}\right)^2 \ln(4L/\beta)$. Thus $c \equiv M_1(j)$ or $M_2(j) \bmod 4$; the $\mu \leq (c+0.25)2^j)$ case is symmetric. If instead $\mu \in ((c+0.25)2^j, (c+0.75)2^j)$ then by similar logic with probability at least $1-\beta/5$

$$\hat{H}_1^j(c \bmod 4) \geq 0.36k - \psi$$

so by $\psi < 0.08k$ (implied by $k > 40\left(\frac{\varepsilon+4}{\varepsilon\sqrt{2}}\right)^2 \ln(8L/\beta)$) $c \equiv M_1(j)$ or $M_2(j) \bmod 4$. It follows that with probability at least $1-3\beta/5$ in all cases $c \equiv M_1(j)$ or $M_2(j) \bmod 4$. Moreover, by a similar application of the Gaussian CDF, one of $c-1 \bmod 4$ and $c+1 \bmod 4$ lies in $\{M_1(j), M_2(j)\}$ as well.

Recalling that c^* is the maximum integer such that $c^* 2^j \in I_j$ and $c^* \equiv M_1(j)$ or $M_2(j) \bmod 4$, $c^* - 1 \bmod 4 \in \{M_1(j), M_2(j)\}$ as well. Assume $|c^* 2^j - \mu| > 2\sigma$. By above, $\mu \in [c^* 2^j, (c^*+1)2^j)$ or $[(c^*-1)2^j, (c^* 2^j))$. In the first case,

$$\mathbb{P}\left[\lfloor x/2^j \rfloor \equiv c^* - 1 \bmod 4\right] \leq \mathbb{P}\left[x \notin [\mu - 2\sigma, \mu + 2\sigma]\right] \leq 0.05$$

so with probability at least $1-\beta/5$, $\hat{H}_1^j(c^*-1) \leq 0.06k+\psi$, a contradiction of $c^*-1 \bmod 4 \in \{M_1(j), M_2(j)\}$. In the second case,

$$\mathbb{P}\left[\lfloor x/2^j \rfloor \equiv c^* \bmod 4\right] \leq \mathbb{P}\left[x \notin [\mu-2\sigma, \mu+2\sigma]\right] \leq 0.05$$

and with probability at least $1-\beta/5$, $\hat{H}_1^j(c^*) \leq 0.06k+\psi$, contradicting $c^* \bmod 4 \in \{M_1(j), M_2(j)\}$. Thus $|c^*2^j - \mu| \leq 2\sigma$.

We put these facts together in EstMean as follows: let j_1 be the maximum element of $\mathcal{L}$ such that $\hat{H}_1^j(M_1(j)) < 0.52k - \psi$. If $2^{j_1} > \sigma$, then by Fact 2 setting $\hat{\mu}_1 = c^*2^j$ implies $|\hat{\mu}_1 - \mu| \leq 2\sigma$. If instead $2^{j_1} \leq \sigma$, then any setting of $\hat{\mu}_1 \in I_j$ (including $\hat{\mu}_1 = c^*2^j$) guarantees $|\hat{\mu}_1 - \mu| \leq 2^{j_1+1} \leq 2\sigma$. Thus in all cases, with probability at least $1-\beta$, $|\hat{\mu}_1 - \mu| \leq 2\sigma$. □

The results above give the protocol an (initial) estimate $\hat{\mu}_1$ such that $|\hat{\mu}_1 - \mu| \leq 2\sigma$. This concludes our analysis of round one of KVGausstimate. Now, the protocol passes this estimate $\hat{\mu}_1$ to users $i \in U_2$, and each user uses $\hat{\mu}_1$ to center their value x_i and randomized respond on the resulting $(x_i - \hat{\mu}_1)/\sigma$ in KVRR2. The protocol then aggregates these results using KVAgg2. We now prove that this centering process results in a more accurate final estimate $\hat{\mu}_2$ of μ, which concludes the overall proof.

Lemma 13. *Conditioned on the success of the previous lemmas, with probability at least* $1-\beta$ KVGausstimate *outputs* $\hat{\mu}_2$ *such that*

$$|\hat{\mu}_2 - \mu| = O\left(\frac{\sigma}{\varepsilon}\sqrt{\frac{\log(1/\beta)}{n}}\right).$$

Proof. The proof is broadly similar to that of Theorem B.1 in Braverman et al. [18], with some modifications for privacy. First, by Lemma 12 $\mu - \hat{\mu}_1 \in [-2\sigma, 2\sigma]$. Letting $\bar{\mu} =$

$(\mu - \hat{\mu}_1)/\sigma$ we get that $x_i' \sim N(\bar{\mu}, 1)$. Next, since $\mathbb{E}[y_i] = 2\mathbb{P}[x_i' \geq 0] - 1$, and in general

$$\Phi_{\mu,\sigma^2}(x) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right)$$

where Φ_{μ,σ^2} is the CDF of $N(\mu, \sigma^2)$, by $\Phi_{\bar{\mu},1}(0) = \mathbb{P}[x_i' \geq 0]$ we get $\mathbb{E}[y_i] = \operatorname{erf}(\bar{\mu}/\sqrt{2})$. Note that we are analyzing the unprivatized values y_i to start; later, we will use this analysis to prove the analogous result for the privatized values $\tilde{y}_i$.

A Chernoff bound then shows that, with probability at least $1 - \beta/2$, for $y = \frac{2}{n}\sum_{i \in U_2} y_i$ we have

$$|y - \operatorname{erf}(\bar{\mu}/\sqrt{2})| \leq 2\sqrt{\ln(4/\beta)/n}$$

and by $\mathbb{E}[y] = \operatorname{erf}(\bar{\mu}/\sqrt{2})$ we get $|y - \mathbb{E}[y]| \leq 2\sqrt{\ln(4/\beta)/n}$ as well.

Since $\mu - \hat{\mu}_1 \in [-2\sigma, 2\sigma]$, $|\operatorname{erf}(\bar{\mu}/\sqrt{2})| \leq \operatorname{erf}(\sqrt{2})$. Thus $|\mathbb{E}[y]| \leq \operatorname{erf}(\sqrt{2})$, so by $|y - \mathbb{E}[y]| \leq 2\sqrt{\ln(4/\beta)/n}$ we get

$$|y| \leq \operatorname{erf}(\sqrt{2}) + 2\sqrt{\ln(4/\beta)/n}.$$

Using $n > 20000\ln(4/\beta)$ we get $2\sqrt{\ln(4/\beta)/n} < 0.01$ and $\operatorname{erf}(\sqrt{2}) < 0.96$, so $|y| \leq 0.97$ and thus $|y| < \operatorname{erf}(1.6)$. Let M be an upper bound on the Lipschitz constant for erf^{-1} in $[-0.97, 0.97]$,

$$\begin{aligned} M &= \max_{x \in [-0.97, 0.97]} \frac{d\operatorname{erf}^{-1}(x)}{dx} \\ &= \max_{x \in [-0.97, 0.97]} \frac{\sqrt{\pi}}{2} \exp([\operatorname{erf}^{-1}(x)]^2) \\ &\leq \frac{\sqrt{\pi}}{2} \exp([\operatorname{erf}^{-1}(0.97)]^2) < 10. \end{aligned}$$

Then for any $x, y \in [-0.97, 0.97]$ we have $|\operatorname{erf}^{-1}(x) - \operatorname{erf}^{-1}(y)| \leq M|x - y|$, so setting

$T = \sqrt{2}\text{erf}^{-1}(y)$,

$$|T - \bar{\mu}| = |\sqrt{2}(\text{erf}^{-1}(y) - \text{erf}^{-1}(\mathbb{E}[y])| \leq 10\sqrt{2}|y - \mathbb{E}[y]| \\ \leq 20\sqrt{2\ln(4/\beta)/n}$$

using the bound on $|y - \mathbb{E}[y]|$ from above.

It remains to analyze the privatized values $\{\tilde{y}_i\}$ and bound $|T - \hat{T}|$, recalling that we set

$$\hat{T} = \sqrt{2} \cdot \text{erf}^{-1}\left(\frac{2(-\hat{H}_2(-1) + \hat{H}_2(1))}{n}\right)$$

in KVAGG1. By a Chernoff bound analogous to that of Lemma 11, with probability at least $1 - \beta/2$

$$|T - \hat{T}| \leq \sqrt{2}\left|\text{erf}^{-1}(|y|) - \text{erf}^{-1}\left(|y| + \left[\frac{\varepsilon + 2}{\varepsilon}\right]\sqrt{\frac{2\ln(4/\beta)}{n}}\right)\right|.$$

Using $n > 20000\left(\frac{\varepsilon+2}{\varepsilon}\right)^2 \ln(4/\beta)$ (which implies $\left[\frac{\varepsilon+2}{\varepsilon}\right]\sqrt{\frac{2\ln(4/\beta)}{n}} \leq 0.01$) and the same derivative trick as above on $[-0.98, 0.98]$, we get

$$|T - \hat{T}| \leq 14\left[\frac{\varepsilon + 2}{\varepsilon}\right]\sqrt{\frac{2\ln(4/\beta)}{n}}.$$

Therefore by the triangle inequality

$$|\hat{T} - \bar{\mu}| \leq \left(20 + 14\left[\frac{\varepsilon + 2}{\varepsilon}\right]\right)\sqrt{\frac{2\ln(4/\beta)}{n}}$$

and by $\sigma\bar{\mu} = \mu - \hat{\mu}_1$ we get

$$|\sigma\hat{T} - \sigma\bar{\mu}| = |(\sigma\hat{T} + \hat{\mu}_1) - \mu| \leq \sigma\left(20 + 14\left[\frac{\varepsilon + 2}{\varepsilon}\right]\right)\sqrt{\frac{2\ln(4/\beta)}{n}}.$$

Thus by taking $\hat{\mu}_2 = \sigma\hat{T} + \hat{\mu}_1$, we get

$$|\hat{\mu}_2 - \mu| = O\left(\frac{\sigma}{\varepsilon}\sqrt{\frac{\log(1/\beta)}{n}}\right).$$

□

This concludes the overall proof. □

Chapter 6

Exponential Separation: Fully vs. Sequentially Interactive Local Privacy

The previous section focused on polynomial separations between central and local privacy. We now proceed to an exponential separation between fully and sequentially interactive local privacy [48]. The main tool that drives this result is a general connection between the *communication complexity* of two-player problems and the *sample complexity* of sequentially interactive locally private multi-player problems. Informally, we show that the "noise" that must be added to ensure local privacy also makes it possible to convert the protocol into a two-party protocol over a noisy channel, and vice-versa. In combination with past work relating communication complexity over noisy and noiseless channels, this conversion extends to noiseless channels as well (Theorem 8). This connection enables us to translate existing communication lower bounds (here, for the "hidden layers" problem) into sample complexity lower bounds for sequentially interactive locally private protocols (Corollary 1).

We also include a lower bound offering a polynomial separation between fully and sequentially interactive local privacy (Theorem 11). This separation is smaller, based on a hidden layers-style problem, and requires a much longer analysis. However, it does demonstrate that the compositionality dependence of our full-to-sequential conversion (Theorem 2) is tight.

6.1. Additional Preliminaries

Since our results rely on past work on two-party communication, we now define some of the necessary terms.

Definition 14. *In the* two-party communication model*, one player Alice receives input $x \in \mathcal{X}$, and the other player Bob receives input $y \in \mathcal{Y}$. Alice and Bob want to jointly*

compute some output $z \in \mathcal{Z}$ such that (x, y, z) satisfies some relation $\mathcal{R} \subset \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$.

To compute z, Alice and Bob coordinate their actions using a *protocol*.

Definition 15. *Given a two-party communication model, a* protocol $\mathcal{A}$ *specifies a binary output function that each player should apply to their data at each time step, as a function of the time step, the* transcript *of previously released values and any shared randomness.*

Note that, by our inclusion of shared randomness, all protocols we consider are public-coin. Two salient protocol characteristics are the number of bits that must be exchanged and the likelihood of a "good" outcome.

Definition 16. *The* communication complexity *of a protocol $\mathcal{A}$, denoted by* $\mathrm{CC}(\mathcal{A})$*, is the maximum number of bits exchanged over all inputs (x, y) and all random coins. If on all inputs x and y $\mathbb{P}_{\mathcal{A}}[(x, y, z) \notin \mathcal{R}] \leq \gamma$, we say $\mathcal{A}$* computes $\mathcal{R}$ with error at most γ. *The* randomized communication complexity with error γ *of a relation $\mathcal{R}$ is then defined as*

$$\mathrm{CC}_\gamma(\mathcal{R}) = \min_{\mathcal{A}:\mathcal{A} \text{ computes } \mathcal{R} \text{ with error at most } \gamma} \mathrm{CC}(\mathcal{A}).$$

We can also combine this communication model and the notion of protocol defined in Chapter 2 to define a *multi-party communication model*.

Definition 17. *In the* multi-party communication model *there exists an "Alice input" $x \in \mathcal{X}$ and a "Bob input" $y \in \mathcal{Y}$. Each of an unboundedly large number of players receives an independent and uniformly random draw over $\{x, y\}$. The users' goal is to output $z \in \mathcal{Z}$ such that $(x, y, z) \in \mathcal{R}$.*

A *multi-party protocol* is defined as in the local privacy setting (and we will be interested in local privacy as the primary constraint on multi-party protocols). As before, we will quantify the likelihood that a multi-party protocol achieves a "good" outcome. Unlike before, our metric of interest for a multi-party protocol is its *sample complexity*.

Definition 18. *The* sample complexity *of a multi-party protocol $\mathcal{A}$, denoted by* $\mathrm{SC}(\mathcal{A})$*, is the maximum number of users appearing in the transcript over all inputs (x, y) and all*

random coins. If on all inputs x and y $\mathbb{P}_{\mathcal{A}}[(x, y, z) \notin \mathcal{R}] \leq \gamma$, we say $\mathcal{A}$ computes $\mathcal{R}$ with error at most γ. *The* randomized sample complexity with error γ *of a relation* $\mathcal{R}$ *is then defined as*

$$\mathrm{SC}_{\gamma}(\mathcal{R}) = \min_{\mathcal{A}:\mathcal{A} \text{ computes } \mathcal{R} \text{ with error at most } \gamma} \mathrm{SC}(\mathcal{A}).$$

Let $\mathrm{SC}_{\gamma}^{\varepsilon,N}(\mathcal{R}), \mathrm{SC}_{\gamma}^{\varepsilon,S}(\mathcal{R})$, *and* $\mathrm{SC}_{\gamma}^{\varepsilon,F}(\mathcal{R})$ *denote the sample complexities of* ε*-locally private protocols computing* $\mathcal{R}$ *with error* γ *under noninteraction, sequential interaction, and full interaction respectively.*

Our two-party models are defined by an input pair (x, y), and we deliberately constrain our multi-party models to be defined by a pair (x, y) as well, where each party randomly receives either x or y. Each user therefore has an equal chance of being an "Alice" or "Bob" user. In this way, a given two-party problem on a pair of inputs induces a unique multi-party problem on the same inputs, and vice-versa. We note that multi-party problems as we define them are unusual statistical estimation problems: their primary use for us will be in proving lower bounds.

6.2. Reduction and Separation

We now prove an equivalence between a two-party problem's communication complexity and the induced multi-party problem's sample complexity for sequentially interactive locally private protocols. We do so by using *noisy* two-party communication complexity as an intermediary.

First, we give an equivalence between noisy two-party communication and locally private multi-party communication. To convert a two-party protocol over an ε'-noisy channel into an ε-locally private multi-party protocol, at each step of the protocol the next user will imitate the action of Alice or Bob (depending on their own datum) in the two-party protocol, all while using ε-randomized response. Because the original two-party protocol is intended for a noisy channel, the multi-party protocol works like the two-party protocol as long as users employ an appropriately calibrated ε for randomized response (Lemma 14). In

the other direction, it is possible to convert locally private multi-party protocols to noisy two-party protocols by a similar idea (Lemma 15).

The results so far connect noisy two-party communication and locally private multi-party communication. However, we are interested in using noiseless two-party communication lower bounds. It therefore remains to connect noisy and noiseless two-party communication. Fortunately, previous work has already accomplished this step. We recap these earlier contributions in Lemma 18 and Lemma 20.

Finally, sequential interactivity means that each user in the multi-party protocol speaks at most once. The number of users in the multi-party protocol is therefore connected to the number of communications in the two-party protocol. This leads to the overall equivalence between two-party communication and locally private multi-party sample complexity (Theorem 8).

6.2.1. Noisy Two-party Communication

We start by connecting noisy two-party communication and locally private multi-party communication. In the noisy two-party communication model, Alice and Bob may only communicate over a *binary symmetric channel* that flips each transmitted bit with a certain probability.

Definition 19. *For $\varepsilon \in (0, 1/2)$, a* binary symmetric channel with crossover probability *ε, denoted BSC_ε, correctly transmits a bit b with probability $1/2 + \varepsilon$ and transmits $1 - b$ with probability $1/2 - \varepsilon$. We additionally suppose that the binary symmetric channel has* feedback*: the sender always sees the received bit.*

Let $\mathrm{CC}^{\varepsilon}_{\gamma}(\mathcal{R})$ denote the communication complexity of $\mathcal{R}$ with error γ under the additional requirement that communication occurs over BSC_ε. We now show how to transform two-party protocols over a binary symmetric channel into multi-party protocols.

Lemma 14. *Let $\mathcal{R}$ be a relation for some communication problem, $\varepsilon \geq 0$, $\varepsilon' = \frac{e^\varepsilon - 1}{4(e^\varepsilon + 1)}$, and $\gamma \in (0, 1)$. Then $\mathrm{SC}^{\varepsilon, S}_{\gamma}(\mathcal{R}) = O\left(\mathrm{CC}^{\varepsilon'}_{\gamma}(\mathcal{R})\right)$.*

Proof. Let $\mathcal{A}_2$ be any protocol for the two-party problem over $\mathrm{BSC}_{\varepsilon'}$ computing $\mathcal{R}$ with error γ. Consider the first bit sent in $\mathcal{A}_2$. Without loss of generality, Alice sends this first bit $f(x)$, where x is Alice's input. Since communication occurs over $\mathrm{BSC}_{\varepsilon}$, with probability $1/2 + \varepsilon'$ Bob receives $f(x)$, and with probability $1/2 - \varepsilon'$ Bob receives its negation.

We will use $\mathcal{A}_2$ to build a multi-party protocol $\mathcal{A}_m$. To simulate this bit, $\mathcal{A}_m$ selects a new (previously un-selected) agent, and the new agent takes one of two actions. If the agent has an Alice input x, then they send $\mathsf{RR}(x, \varepsilon)$. If instead the agent has a Bob input y, then they send a uniform random bit. Thus the probability that the agent sends $f(x)$ is

$$\begin{aligned}
\mathbb{P}\left[\text{Alice input}\right] \cdot \tfrac{e^{\varepsilon}}{e^{\varepsilon}+1} + \mathbb{P}\left[\text{Bob input}\right] \cdot \tfrac{1}{2} &= \tfrac{e^{\varepsilon}}{2(e^{\varepsilon}+1)} + \tfrac{1}{4} \\
&= \tfrac{2e^{\varepsilon}}{4(e^{\varepsilon}+1)} + \tfrac{e^{\varepsilon}+1}{4(e^{\varepsilon}+1)} \\
&= \tfrac{3e^{\varepsilon}+1}{4(e^{\varepsilon}+1)} \\
&= \tfrac{2(e^{\varepsilon}+1)}{4(e^{\varepsilon}+1)} + \tfrac{e^{\varepsilon}-1}{4(e^{\varepsilon}+1)} \\
&= \tfrac{1}{2} + \varepsilon'.
\end{aligned}$$

It follows that the first bit of $\mathcal{A}_m$ is distributed identically to the first bit of $\mathcal{A}_2$. Repeating this process for each bit sent in $\mathcal{A}_2$, $\mathcal{A}_m$ induces an identical distribution over the bits output, and thus computes $\mathcal{R}$ with error γ. Since each bit sent in $\mathcal{A}_2$ used a new user in $\mathcal{A}_m$, $\mathrm{SC}(\mathcal{A}_m) = O\left(\mathrm{CC}^{\varepsilon'}(\mathcal{A}_2)\right)$. Since randomized response satisfies ϵ-differential privacy, the sequentially interactive mechanism $\mathcal{A}_m$ is ϵ-locally private. □

In the other direction, we now show how to transform locally private multi-party protocols into two-party protocols over a binary symmetric channel.

Lemma 15. *Let $\mathcal{R}$ be a relation for some communication problem, $0 < \varepsilon = O(1)$, $\varepsilon' = \frac{e^{\varepsilon}-1}{2(e^{\varepsilon}+1)}$, and $\gamma, \eta > 0$ such that $\gamma + \eta < 1$. Then $\mathrm{CC}^{\varepsilon'}_{\gamma+\eta}(\mathcal{R}) = O\left(\frac{1}{\eta} \cdot \mathrm{SC}^{\varepsilon,S}_{\gamma}(\mathcal{R})\right)$.*

Proof. Here, we define $\varepsilon' = \frac{e^{\varepsilon}-1}{2(e^{\varepsilon}+1)}$, which differs from our previous ε' by a factor of 2.

Let $\mathcal{A}_m$ be any sequentially interactive ε-locally private protocol for a multi-party problem computing $\mathcal{R}$ with error γ and sample complexity n. By the following result from Bassily and Smith [12], we can transform $\mathcal{A}_m$ into a new, functionally equivalent protocol $\mathcal{A}'_m$ in which each user sends only one bit.

Lemma 16 (Theorem 4.1 in Bassily and Smith [12])**.** *Given an ε-locally private protocol $\mathcal{A}$ with expected number of randomizer calls T, there exists a sequentially interactive ε-locally private protocol $\mathcal{A}'$ with expected number of users $e^{\varepsilon} \cdot T$ where each user sends a single bit (produced by a call to a single ε-local randomizer). Moreover, there exists a deterministic function f on transcripts such that $f(\Pi(\mathcal{A}')) = \Pi(\mathcal{A})$, where $\Pi(\cdot)$ denotes a distribution over transcripts induced by a given protocol with randomness is over the protocol and its samples.*

Note that the deterministic function f is only technical bookkeeping to account for the fact that $\Pi(\mathcal{A}')$ and $\Pi(\mathcal{A})$ do not have the same representation, but either one can still be translated to the other. The cost is twofold. First, $\mathcal{A}'_m$ requires $O(n \log(\log(n)))$ bits of public randomness. Second, $\mathcal{A}'_m$ requires $e^{\varepsilon} n$ users in expectation. By Markov's inequality, the number of users can be bounded by $\frac{e^{\varepsilon} n}{\eta}$ at the cost of an η increase in failure probability.

We now transform $\mathcal{A}'_m$ into a two-player protocol $\mathcal{A}_2$. The idea is that Alice and Bob simulate A'_m by randomly partitioning the users from the multi-party protocol between themselves. Each then simulates the role of their assigned users. To see why this works, recall that for multi-party communication problems, users are randomly assigned "Alice" or "Bob" data points. It follows that this random partition will induce the same distribution on data elements. Thus, $\mathcal{A}_2$ begins with Alice and Bob using their shared public randomness to generate $\frac{e^{\varepsilon} n}{\eta}$ coin flips determining who will simulate which agents.

Without loss of generality, suppose Alice simulates the first agent. Let

$$p_x = \mathbb{P}\left[\text{agent sends } 1 \mid \text{agent has Alice's data } x\right],$$

and let $p_{\min} = \min_{x \in \mathcal{X}} p_x$ and $p_{\max} = \max_{x \in \mathcal{X}} p_x$. Alice and Bob take one of two choices depending on $p_{\min}$ and $p_{\max}$.

Case 1: $p_{\min} + p_{\max} \leq 1$. Then Alice sends 1 with probability $\frac{1}{2} + \frac{p_x}{2\varepsilon'(p_{\min}+p_{\max})} - \frac{1}{4\varepsilon'}$ and sends 0 with the remaining probability. Since

$$\begin{aligned}
\tfrac{1}{2} + \tfrac{p_x}{2\varepsilon'(p_{\min}+p_{\max})} - \tfrac{1}{4\varepsilon'} &= \tfrac{1}{2} + \tfrac{2p_x - p_{\min} - p_{\max}}{4\varepsilon'(p_{\min}+p_{\max})} \\
&\leq \tfrac{1}{2} + \tfrac{p_{\max} - p_{\min}}{4\varepsilon'(p_{\min}+p_{\max})} \\
&= \tfrac{1}{2} + \tfrac{e^{\varepsilon}+1}{2(e^{\varepsilon}-1)} \cdot \tfrac{p_{\max} - p_{\min}}{p_{\max}+p_{\min}} \\
&= \tfrac{1}{2} + \tfrac{e^{\varepsilon}+1}{2(e^{\varepsilon}-1)} \cdot \left[1 - \tfrac{2p_{\min}}{p_{\max}+p_{\min}}\right] \\
&\leq \tfrac{1}{2} + \tfrac{e^{\varepsilon}+1}{2(e^{\varepsilon}-1)} \left[1 - \tfrac{2}{e^{\varepsilon}+1}\right] = 1
\end{aligned}$$

(where both inequalities use the fact that the agent sends output from an ε-local randomizer), and similarly

$$\begin{aligned}
\tfrac{1}{2} + \tfrac{2p_x - p_{\min} - p_{\max}}{4\varepsilon'(p_{\min}+p_{\max})} &\geq \tfrac{1}{2} + \tfrac{e^{\varepsilon}+1}{2(e^{\varepsilon}-1)} \cdot \tfrac{p_{\min} - p_{\max}}{p_{\max}+p_{\min}} \\
&\geq \tfrac{1}{2} + \tfrac{e^{\varepsilon}+1}{2(e^{\varepsilon}-1)} \left[\tfrac{2}{e^{\varepsilon}+1} - 1\right] = 0
\end{aligned}$$

these are valid probabilities. Next, as Alice sends the bit over $\mathrm{BSC}_{\varepsilon'}$, the probability that the received bit is 1 is

$$\left(\tfrac{1}{2} + \varepsilon'\right)\left(\tfrac{1}{2} + \tfrac{p_x}{2\varepsilon'(p_{\min}+p_{\max})} - \tfrac{1}{4\varepsilon'}\right) + \left(\tfrac{1}{2} - \varepsilon'\right)\left(\tfrac{1}{2} - \tfrac{p_x}{2\varepsilon'(p_{\min}+p_{\max})} + \tfrac{1}{4\varepsilon'}\right) = \tfrac{p_x}{p_{\min}+p_{\max}}.$$

With probability $p_{\min} + p_{\max}$, Alice and Bob "use" the received bit: that is, they enter this received bit into their transcript and continue the protocol. With probability $1 - p_{\min} - p_{\max}$ Alice and Bob instead enter the bit 0 into their transcript (and omit the true received bit from the transcript) and continue. Then $\mathbb{P}$ [enter 1 in transcript] (and $\mathbb{P}$ [enter 0 in transcript]) are identical in both the two-party and multi-party protocols.

Case 2: $p_{\min} + p_{\max} > 1$. Then if we define $p'_{\min}$ and $p'_{\max}$ as $1 - p_{\min}$ and $1 - p_{\max}$

respectively, we get $p'_{\min} + p'_{\max} < 1$. Let $p'_x = 1 - p_x$ and have Alice send 0 with probability $\frac{1}{2} + \frac{p'_x}{2\varepsilon'(p'_{\min}+p'_{\max})} - \frac{1}{4\varepsilon'}$, and Alice and Bob "use" the received bit (just as defined in Case 1) with probability $p'_{\min} + p'_{\max}$. Repeating the analysis from Case 1 for p'_x, $p'_{\min}$, and $p'_{\max}$ yields that $\mathbb{P}[\text{use 0}]$ (and $\mathbb{P}[\text{use 1}]$) are identical in both the two-party and multi-party protocols.

Combining Cases 1 and 2, Alice and Bob produce the same distribution over the first bit of the protocol in $\mathcal{A}_2$ and $\mathcal{A}_m$. Since we can repeat this process for subsequent bits, by induction the distribution over transcripts (and thus answers) is identical. Therefore $\mathcal{A}_2$ also computes $\mathcal{R}$ with error γ. Moreover, there is a one-to-one correspondence between users in $\mathcal{A}_m$ and bits in $\mathcal{A}_2$, so by $\varepsilon = O(1)$, $\mathrm{CC}^{\varepsilon'}_{\gamma+\eta}(\mathcal{R}) = O\left(\frac{1}{\eta} \cdot \mathrm{SC}^{\varepsilon,S}_{\gamma}(\mathcal{R})\right)$. □

6.2.2. Relating Noisy and Noiseless Communication

Lemmas 14 and 15 relate the sample complexity of sequentially interactive ε-locally private multi-party protocols and the communication complexity of noisy two-party protocols. The remaining step is to relate the communication complexity of noisy and noiseless two-party protocols.

The noisy-to-noiseless direction follows almost immediately from previous work by Braverman and Mao [16]. Note that, in one sense, it is trivial to simulate noisy communication over a noiseless channel: the sender can simply simulate the binary symmetric channel by adding noise to their own messages. The resulting protocol inherits the same functionality and communication complexity as the noisy original. However, the Lemma 17 does better. In particular, it takes advantage of the information loss of the noisy channel to *reduce* the communication complexity when given a noiseless channel.

Lemma 17 (Theorem 3.1 in Braverman and Mao [16])**.** *For every protocol $\mathcal{A}$ over BSC_ε with feedback, there exists a protocol $\mathcal{A}'$ over a noiseless channel that simulates $\mathcal{A}$ with $\overline{\mathrm{CC}}(\mathcal{A}') = O\left(\varepsilon^2 \mathrm{CC}^{\varepsilon}(\mathcal{A})\right)$. Here, $\overline{\mathrm{CC}}(\mathcal{A}')$ is the maximum over all inputs (x, y) of the expected number of bits exchanged over the randomness of $\mathcal{A}'$.*

By Markov's inequality, we get a high-probability version of their result for our setting.

Lemma 18. *Let $\mathcal{R}$ be a relation for a two-party communication problem. Then for $\gamma, \eta > 0$ where $\gamma + \eta < 1$, $\mathrm{CC}_{\gamma+\eta}(\mathcal{R}) = O\left(\frac{\varepsilon^2}{\eta} \cdot \mathrm{CC}^{\varepsilon}_{\gamma}(\mathcal{R})\right)$.*

It remains to upper bound noisy communication complexity using noiseless communication complexity. Schulman [59] first studied the problem of noisy interactive two-party communication. He showed how to simulate a noiseless channel using a binary symmetric channel with small ($< 1/240$) crossover probability and a constant blowup in communication complexity. Braverman and Rao [17] then improved this result to binary symmetric channels with crossover probability bounded away from $1/8$ by a constant.

Lemma 19 (Simplified Version of Theorem 2 in Braverman and Rao [17])**.** *Let $\mathcal{R}$ be a relation for a two-party communication problem, $\gamma \in (0, 1)$, and $0 < p \leq 1/8 - c$ where $c = \Omega(1)$. Then $\mathrm{CC}^{p}_{\gamma}(\mathcal{R}) = O\left(\mathrm{CC}_{\gamma}(\mathcal{R})\right)$.*

One obstacle remains: Lemma 19 requires a channel C with crossover probability bounded away from $1/8$, while our channel C' may have crossover probability ε-close to $1/2$. We therefore use a standard amplification argument, replacing each bit over C with $\Theta(1/\varepsilon^2)$ bits over C' and taking the majority as the transmitted bit. This yields Lemma 20.

Lemma 20. *Let $\mathcal{R}$ be a relation for a two-party communication problem. Then $\mathrm{CC}^{\varepsilon}_{\gamma}(\mathcal{R}) = O\left(\frac{1}{\varepsilon^2} \cdot \mathrm{CC}_{\gamma}(\mathcal{R})\right)$.*

6.2.3. Equivalence

The results above yield the following theorem. The proof is simply chaining together the appropriate results in each direction.

Theorem 8. *Let $\mathcal{R}$ be a relation for some communication problem. Then for any $\varepsilon = O(1)$ and $0 < \gamma, \eta = \Omega(1)$ such that $\gamma + \eta < 1$, $\mathrm{SC}^{\varepsilon,S}_{\gamma}(\mathcal{R}) = \Theta\left(\frac{1}{\varepsilon^2} \cdot \mathrm{CC}_{\gamma+\eta}(\mathcal{R})\right)$.*

Proof. We first show $\mathrm{SC}^{\varepsilon,S}_{\gamma}(\mathcal{R}) = O\left(\frac{1}{\varepsilon^2} \cdot \mathrm{CC}_{\gamma}(\mathcal{R})\right)$. By Lemma 14, $\mathrm{SC}^{\varepsilon,S}_{\gamma}(\mathcal{R}) = O\left(\mathrm{CC}^{\varepsilon_1}_{\gamma}(\mathcal{R})\right)$ where $\varepsilon_1 = \frac{e^{\varepsilon}-1}{4(e^{\varepsilon}+1)}$. Then, by Lemma 20, $\mathrm{CC}^{\varepsilon_1}_{\gamma}(\mathcal{R}) = O\left(\frac{1}{\varepsilon_1^2} \cdot \mathrm{CC}_{\gamma}(\mathcal{R})\right)$. Since $\varepsilon = O(1)$,

$\varepsilon_1 = \Omega(\varepsilon)$, and tracing back yields the claim.

Next, we show $\mathrm{CC}_{\gamma+\eta}(\mathcal{R}) = O\left(\varepsilon^2 \cdot \mathrm{SC}_{\gamma}^{\varepsilon,S}(\mathcal{R})\right)$. Since $\gamma, \eta = \Omega(1)$, by Lemma 18 $\mathrm{CC}_{\gamma+\eta}(\mathcal{R}) = O\left(\varepsilon_1^2 \cdot \mathrm{CC}_{\gamma+\eta/2}^{\varepsilon_1}(\mathcal{R})\right)$ where $\varepsilon_1 = \frac{e^{\varepsilon}-1}{2(e^{\varepsilon}+1)}$. By Lemma 15, $\mathrm{CC}_{\gamma+\eta/2}^{\varepsilon_1}(\mathcal{R}) = O\left(\mathrm{SC}_{\gamma}^{\varepsilon,S}(\mathcal{R})\right)$. Tracing back and using $\varepsilon_1 = O(\varepsilon)$ implies the claim. □

6.3. Separating Sequential and Full Interactivity

The previous section showed that any two-party communication lower bound implies a sequentially interactive locally private multi-party sample complexity lower bound. In this section, we plug in a two-party communication lower bound for the *hidden layers* problem to show that it is hard for sequentially interactive locally private protocols (Corollary 1). In contrast, the same problem is much easier for fully interactive protocols (Theorem 9).

6.3.1. Hidden Layers Problem $\mathcal{HL}$

We first formally recap the hidden layers problem that drives our results. While Braverman [15] first proposed this problem, we imitate the presentation of Ganor, Kol, and Raz [40].

The hidden layers problem is essentially a search problem on a large tree. Alice and Bob each have information about one "hidden layer" in the tree, and any correct solution must agree with that information on both layers. Other layers do not affect the correctness of the solution, so there are many correct answers. The problem is that the tree is so large that neither Alice nor Bob can simply communicate their information to the other in $o(2^k)$ bits.

More formally, the hidden layers problem is parameterized by $k \in \mathbb{N}$ and denoted $\mathcal{HL}(k)$. It features a 2^{4k}-ary tree $\mathcal{T}$ with directed edges from root to leaves and $2^{rs}+1$ layers where $r = 2^{2^{8k}}$ and $s = 2^{8k}$. $\mathcal{T}$ thus has a number of layers triply exponential in k and a number of leaves quadruply exponential in k. Two players, Alice and Bob, each receive a small amount of information about $\mathcal{T}$. Alice receives (a, f) where $a \in \{0, 2, \ldots, 2^{rs}-2\}$ indexes

an even-numbered layer, and f labels each vertex in layer a of $\mathcal{T}$ with a single outgoing edge. Similarly, Bob receives (b, g) where $b \in \{1, 3, \ldots, 2^{rs} - 1\}$ indexes an odd-numbered layer, and g labels each vertex in layer b with a single outgoing edge. Thus, Alice and Bob each have information about one "hidden layer" of $\mathcal{T}$. Letting v be a leaf of $\mathcal{T}$, we say v is *consistent* with (a, f) (or (b, g)) if the path from the root to v goes through an edge identified by f (or g, respectively).

If at the end of protocol $\mathcal{A}$ Alice and Bob output different leaves, or at least one of them outputs a leaf not consistent with at least one of (a, f) and (b, g), we say $\mathcal{A}$ *errs*. A simplified illustration of the hidden layers problem appears in Figure 2.

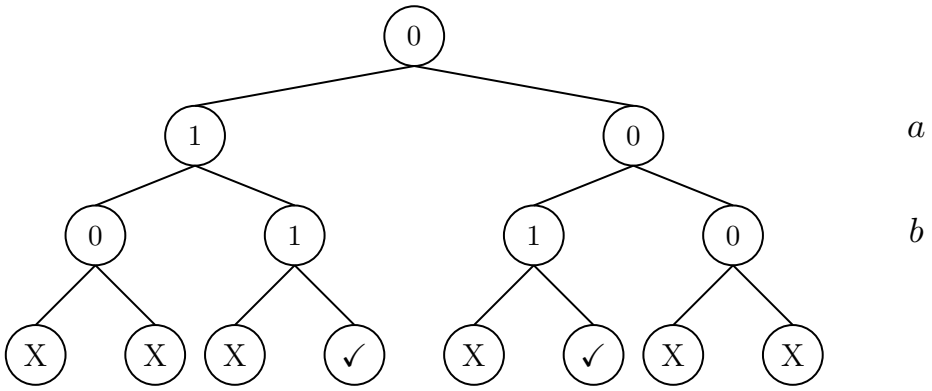

Figure 2: A simplified instance of the hidden layers problem. Each node is labeled 0 (left) or 1 (right). For layers a and b, these labels correspond to the correct child node. Leaves 4 and 6 are thus the only two leaves consistent with the hidden layers a and b. Note that a true instance of the hidden layers problem is much larger.

Ganor et al. [40] proved that the hidden layers problem has high communication complexity. To do so, they constructed a specific distribution P over user inputs for their result. When we want to specify the distribution P over user inputs, we write $\mathcal{HL}(k, P)$.

Lemma 21 (Theorem 1 in Ganor et al. [40])**.** *There exists constant k' and input distribution P for $((a, f), (b, g))$ such that, for every $k \geq k'$ and protocol $\mathcal{A}$ with $\mathrm{CC}(\mathcal{A}) \leq 2^k$, $\mathbb{P}_P[\mathcal{A}$ errs on $\mathcal{HL}(k, P)] \geq 1 - 2^{-k}$.*

In particular, Lemma 21 implies that $\Omega(2^k)$ communication is necessary to achieve constant

success probability for $\mathcal{HL}(k, P)$. Combining Theorem 8 and Lemma 21 gives the following corollary.

Corollary 1. *For $\varepsilon = O(1)$ and $\gamma = \Omega(1)$, $\mathrm{SC}_{\gamma}^{\varepsilon,S}(\mathcal{HL}(k, P)) = \Omega\left(\frac{2^k}{\varepsilon^2}\right)$.*

6.3.2. Fully Interactive Upper Bound for $\mathcal{HL}$

To finish our separation, we now provide a fully interactive protocol HLSolver that solves $\mathcal{HL}(k)$ with poly(k) sample complexity. HLSolver works by greedily following a path from the root to a leaf, querying users to guide its path as it descends the tree.

Concretely, the protocol starts at the root. Then at each vertex v encountered, for all 2^{4k} children v_j of v, the analyst "asks" all n users if (v, v_j) is the labelled edge in that level. To "answer", each user x_i compares the level ℓ of vertex v and edge (v, v_j) to their own data and replies using randomized response.

In the first case, $x_{i,1} = \ell$ and $(v, v_j) \in x_{i,2}$, i.e. user i's hidden layer is ℓ, and v is labelled with the (v, v_j) edge. Then the user "votes yes" and transmits a draw from $\mathsf{RR}(1, \varepsilon)$, i.e. outputs 1 with probability $\frac{e^{\varepsilon}}{e^{\varepsilon}+1}$ and a 0 otherwise. In the second case, the user transmits a draw from $\mathsf{Ber}(1/2)$. Based on the responses, the protocol then chooses an edge out of v to follow, to obtain the next vertex in the path at level $\ell + 1$. When the protocol reaches a leaf, it proposes this leaf as the solution.

Note that any selection made by the protocol at a non-hidden layer is irrelevant: it can follow any outgoing edge and still be on track to correctly solve the problem instance. To argue correctness, all that is important is that for the two (unknown) levels that correspond to hidden layers, the protocol correctly identifies the correct labeled edge. At each of those levels, the bias induced by randomized response will be enough to identify the correct edge with high probability.

While this protocol is run, each user answers a very large (triply exponentially many in k) number of queries. But crucially, only one of those queries is their response sampled

Algorithm 6 HLSolver

procedure HLSolver($\varepsilon, n, \mathcal{T}$)
  Initialize current node $v \leftarrow$ root node
  Initialize level $\ell \leftarrow 0$
  Set $\varepsilon' \leftarrow \varepsilon/2$
  while $\ell <= 2^{rs} - 1$ **do**
    Initialize NextNodeFound $\leftarrow 0$
    Initialize child index $j \leftarrow 0$
    while not NextNodeFound and $j <= 2^{4k} - 1$ **do**
      for users $i = 1, 2, \ldots, n$ **do**
        Initialize $b_i \leftarrow 0$
        if $x_{i,1} = \ell$ and $(v, v_j) \in x_{i,2}$ **then**
          User i outputs $y_i \sim \mathsf{RR}(1, \varepsilon)$
        else
          User i outputs $y_i \sim \mathsf{Ber}(1/2)$
      $\bar{y} \leftarrow \frac{1}{n} \sum_{i=1}^{n} y_i$
      if $\bar{y} \geq 0.6$ or $v_j = v_{2^{4k}-1}$ **then**
        Set new current node $v \leftarrow v_j$
        NextNodeFound $\leftarrow 1$
    $\ell \leftarrow \ell + 1$
  Output v

from $\mathsf{RR}(\varepsilon, 1)$. For all other queries, their response is sampled from $\mathsf{Ber}(1/2)$. Hence the privacy loss over the whole protocol is constant and does not accumulate with the number of queries. In contrast, a similar sequentially interactive protocol would require new users for each of this large number of queries. In the language of Chapter 4, our fully interactive solution is extremely compositional.

Note also that, by Theorem 2, any fully interactive protocol that uses r local randomizer calls per user can be converted into a sequentially interactive protocol with an $O(r)$ factor blowup in sample complexity. Consequently, it is necessary that any protocol witnessing an exponential separation between the sample complexities of fully and sequentially interactive protocols must make at least exponentially many queries per user.

Theorem 9. HLSolver *is ε-locally private and has constant success probability on $\mathcal{HL}(k)$ given $n = \Omega\left(\frac{k}{\varepsilon^2}\right)$ samples.*

Proof. Privacy: Recall that each user draws one of two samples, (a, f) or (b, g). Accordingly,

each user has only a single point in the transcript where they output a sample from $\mathsf{RR}(\varepsilon, 1)$ and the remaining outputs come from $\mathsf{Ber}(1/2)$. Thus, if we compare the transcript distributions (restricted to a single user with the specified data) of $\pi(a, f)$ and $\pi(b, g)$, there are at most two points in the transcript where their output distributions are not identical. Therefore for any single-user transcript output z,

$$\frac{\mathbb{P}[\pi(a, f) = z]}{\mathbb{P}[\pi(b, g) = z]} \leq \frac{\frac{e^\varepsilon}{2(e^\varepsilon+1)}}{\frac{1}{2(e^\varepsilon+1)}} \leq e^\varepsilon.$$

Accuracy: It suffices to show that whenever $\ell \in \{a, b\}$, HLSolver chooses the correct child node. Without loss of generality, consider level a. There are two conditions to verify. First, HLSolver should not choose a child node in a before visiting the correct child. At any incorrect child all users publish output from $\mathsf{Ber}(1/2)$. Thus by a Chernoff bound and union bound, at all $k-1$ incorrect children in level a, $\bar{y} < \frac{1}{2} + \sqrt{\frac{4k+\log(1/\beta)}{2n}}$. Thus for $n \geq 200k + 50\log(1/\beta)$, $\bar{y} < 0.6$ and so HLSolver chooses an incorrect child in a with probability $\leq \beta$.

Next, HLSolver should select the correct child when users vote on it. Since each user has a $1/2$ probability to know the correct child in level a, a Chernoff bound implies that at least $0.4n$ users know the correct child. By the same concentration argument, with probability at least $1-\beta$, for $n = \Omega\left(\frac{k+\log(1/\beta)}{\varepsilon^2}\right)$, at the correct child $\bar{y} \geq 0.6$. $\beta = O(1)$ gives the final result □

Combining Corollary 1 and Theorem 9 yields an exponential (in k) separation between sequentially and fully interactive protocols achieving constant success probability on $\mathcal{HL}$.

We conclude this section by noting that our communication complexity-sample complexity equivalence also recovers an exponential separation between 1) noninteractive and interactive local privacy and 2) k-round and $(k+1)$-round interactive local privacy. Both results use pointer-chasing problems [48]. Past work gave similar separations for the problems of

masked parity [50] and learning decision lists [28] using the polynomial local privacy-SQ learning equivalence of Kasiviswanathan et al. [50]. However, that equivalence extends to fully interactive local privacy only in a restricted way: it shows a relationship between the number of randomizer calls in the locally private protocol and the query complexity of the SQ learner. This is crucial because the number of randomizer calls may greatly exceed the number of users in (and only in) the fully interactive model. In particular, our separation between sequential and full interaction shows that this polynomial equivalence cannot extend to full interaction.

Chapter 7

Polynomial Separation: Fully vs. Sequentially Interactive Local Privacy

Having just given an exponential separation between fully and sequentially interactive local privacy, we now give a polynomial separation. At first glance, this is redundant. However, the goal of this separation is not a dramatic sample complexity difference. Instead, the goal is to show that our full-to-sequential conversion from Chapter 4 is tight up to logarithmic factors. Specifically, we show that any transformation from a fully interactive k-compositional protocol to a sequentially interactive protocol must have a sample complexity blowup of $\tilde{\Omega}(k)$ (Theorem 11). We place this result after the previous section because, viewed simply as a separation, its proof is both much longer and less intuitive than that of Corollary 1, and the sample complexity separation is weaker. However, only Theorem 11 shows that Theorem 2 is tight in terms of compositionality.

7.1. Additional Preliminaries

We prove our lower bound using information theory. We first recap some basic tools.

Definition 20. *The* entropy *of a random variable X, denoted by $H(X)$, is defined as $H(X) = \sum_x \mathbb{P}[X = x] \ln\left(\frac{1}{\mathbb{P}[X=x]}\right)$, and the* conditional entropy *of random variable X conditioned on random variable Y is defined as $H(X|Y) = \mathbb{E}_y[H(X|Y = y)]$.*

Next, we use entropy to define the mutual information between two random variables.

Definition 21. *The* mutual information *between two random variables X and Y is defined as $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$, and the* conditional mutual information *between X and Y given Z is defined as $I(X;Y|Z) = H(X|Z) - H(X|YZ) = H(Y|Z) - H(Y|XZ)$.*

A few facts about mutual information will be useful.

Fact 6. *Let X_1, X_2, Y, Z be random variables, we have $I(X_1X_2; Y|Z) = I(X_1; Y|Z) + I(X_2; Y|X_1Z)$.*

We also recall the definition of KL divergence, originally given as Definition 13 but given again here. Note that, like the other definitions in this section, we use the natural logarithm. This usage is specific to this chapter.

Definition 22. *The* KL divergence *between two random variables X and Y is defined as* $D_{KL}(X||Y) = \sum_x \mathbb{P}[X = x] \ln\left(\frac{\mathbb{P}[X=x]}{\mathbb{P}[Y=x]}\right)$.

The following fact connects KL divergence and mutual information.

Fact 7. *For random variables X, Y, Z,*

$$I(X; Y|Z) = \mathbb{E}_{x,z}\left[D_{KL}\left((Y|X = x, Z = z)||(Y|Z = z)\right)\right].$$

7.2. Multi-Party Pointer Jumping

We prove our lower bound by defining a family of *multi-party pointer jumping* ($\mathcal{MPJ}$) problems such that for every k, there is a fully interactive k-compositional protocol that can solve the problem with sample complexity $n = n(k)$, but such that *any* sequentially interactive protocol solving the problem must have sample complexity $\tilde{\Omega}(k \cdot n)$.

An *instance* of $\mathcal{MPJ}(d)$ is given by a complete tree of depth d. Every vertex of the tree is labelled by one of its children. By following the labels down the tree, starting at the root, an instance defines a unique root-to-leaf path.

It is instructive to distinguish $\mathcal{MPJ}$ from the hidden layers problem. First, there are no more "hidden" layers. Instead, every layer has a correct child, and there is only one solution. We define the user data distribution such that every layer is known by some users, and each user knows about at most one layer.

We first show that there is a fully interactive protocol that can solve this problem with sample complexity $n = \tilde{O}(d^2/\varepsilon^2)$ (Theorem 10). The protocol is k-compositional for $k = \Theta(d)$. The protocol (Algorithm 7) is similar to the fully interactive solution to the Hidden Layers problem (Algorithm 6). The main difference is that the protocol now "asks one question per level" to each of several groups, with each user only responsible for responding about a single group-specific bit of the index for the correct child. The reason for this is that we now need to be careful about identifying the correct child in *every* layer of the problem, whereas in the Hidden Layers problem only two layers were relevant.

We will define a user data distribution with roughly $\tilde{\Theta}(\sqrt{n}/\varepsilon^2)$ users with relevant data in each level, out of n users total. It is therefore (just) possible to identify the child in question subject to local differential privacy. Although every user applies an ε-local randomizer d times in sequence, because each user's data corresponds to only a single level in the tree, the protocol is still ε-locally private. The difficulties faced by a sequentially interactive protocol are similar to those for the Hidden Layers problem: every question must now be posed to a new collection of users.

Our lower bound (Theorem 11) formalizes this intuition using induction on the depth of the tree to bound the success probability of any protocol as a function of its sample complexity. Additionally, the precise definition of $\mathcal{MPJ}(d)$ is somewhat more complicated: half of the weight on the underlying distribution is assigned to "level 0" dummy agents whose purpose is to break correlations between levels of the tree in the argument.

We now formally define multi-party pointer jumping

Definition 23. *Given integer parameter $d > 1$, an instance of* multi-party pointer jumping *$\mathcal{MPJ}(d)$ is defined by a vector $Z = Z_1 \circ \cdots \circ Z_d$, a concatenation of d vectors of increasing length. Letting $s = d^4$, for each $i \in [d]$ Z_i is a vector of s^{i-1} integers in $\{0, 1, \ldots, s-1\}$. For each Z_i, $Z_{i,j}$ is its j^{th} coordinate.*

Viewed as a tree, Z is a complete s-ary tree of depth d where each $Z_{i,j}$ marks a child of

the j-th vertex at depth i. $P = P(Z)$ then denotes the vector of d integers representing the unique root to leaf path *down this tree through the children marked by Z. Formally, P is defined in a recursive way: $P_1 = Z_{1,1}$, ...,$P_i = Z_{i,P_1 \cdot s^{i-1} + P_2 \cdot s^{i-2} + \cdots + P_{i-1} + 1}$,...,$P_d = Z_{d,P_1 \cdot s^{d-1} + P_2 \cdot s^{d-2} + \cdots + P_{d-1} + 1}$.*

Finally, an instance $\mathcal{MPJ}(d)$ defines a data distribution $\mathcal{D}$. For each $x \sim \mathcal{D}$, with probability $1/2$, $x = (0, \emptyset)$ is a "dummy datapoint", and with the remaining probability $x = (\ell, Z_\ell)$ where ℓ is a level drawn uniformly at random from $[d]$. A protocol solves $\mathcal{MPJ}(d)$ if it recovers P using samples from $\mathcal{D}$.

An illustration of $\mathcal{MPJ}(d)$ where $s = 2$ appears in Figure 3.

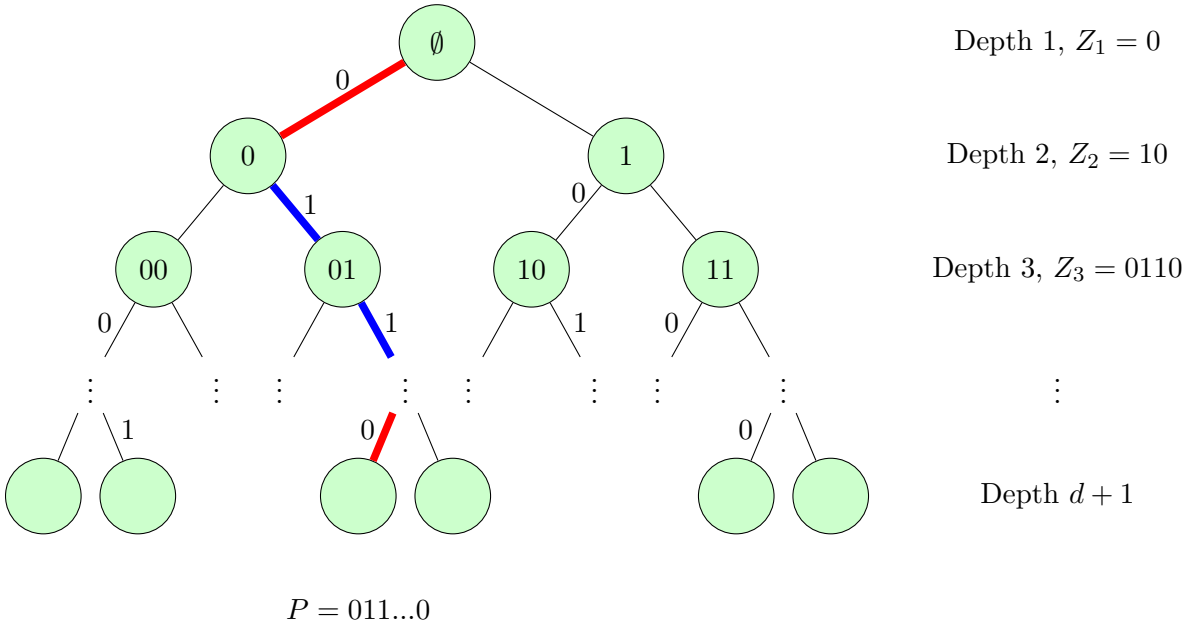

Figure 3: Multi-party pointer jumping

7.3. Separating Sequential and Full Interactivity (Again)

Our separation consists of a fully interactive solution (Theorem 10) and a sequentially interactive lower bound (Theorem 11). We first give pseudocode for our fully interactive solution in Algorithm 7.

Algorithm 7 A fully interactive ε-locally private protocol for $\mathcal{MPJ}(d)$

1: Divide users into $u = \lceil \ln(s)/\ln(2) \rceil$ groups each of $m = 512d^2 \ln(d) \cdot \frac{(e^\varepsilon+1)^2}{(e^\varepsilon-1)^2}$ users.
2: Initialize $Q \leftarrow 0$
3: **for** $r = 1, 2, \ldots, d$ **do**
4: $\quad Q_r \leftarrow 0$
5: $\quad$ **for** each group $g = 1, 2, \ldots, u$ **do**
6: $\quad\quad$ **for** each user $i = 1, 2, \ldots, m$ **do**
7: $\quad\quad\quad \ell_i \leftarrow$ level of user x_i
8: $\quad\quad\quad$ **if** $\ell_i = r$ **then**
9: $\quad\quad\quad\quad b_{i,r} \leftarrow g$-th bit of binary representation of $Z_{r,Q+1}$
10: $\quad\quad\quad\quad$ User i publishes randomized response $y_i \sim \mathsf{RR}(b_{i,r}, \varepsilon)$
11: $\quad\quad\quad$ **else**
12: $\quad\quad\quad\quad$ User i publishes $y_i \sim \mathsf{Ber}(0.5)$
13: $\quad\quad$ g-th bit of $Q_r \leftarrow$ majority bit of $\{y_i\}_{i=1}^m$
14: $\quad Q \leftarrow s \cdot Q + Q_r$
15: Output $Q_1 \circ \cdots \circ Q_d$

Theorem 10. *Algorithm 7 is ε-locally private and, on any instance Z of $\mathcal{MPJ}(d)$, correctly identifies $P(Z)$ with probability at least $1 - 1/d$ with sample complexity*

$$n = O\left(d^2 \log^2(d) \left[\frac{e^\varepsilon + 1}{e^\varepsilon - 1}\right]^2\right).$$

Proof. Privacy: Each user publishes at most one output using randomized response, and the remaining outputs come from $\mathsf{Ber}(0.5)$. The privacy analysis is therefore exactly the same as for Theorem 9.

Sample Complexity: The sample complexity is defined in line 1 of Algorithm 7: $u = O(\log(d))$ groups of $O\left(d^2 \log(d) \left[\frac{e^\varepsilon+1}{e^\varepsilon-1}\right]^2\right)$ users gives the claim.

Accuracy: We first show that each group contains enough users from each level. For each group $g \in [u]$, define binary random variable $X_{i,g,r}$ to be 1 if user i in group g has level r and 0 otherwise. By definition, for any level $r \in [d]$, $\mathbb{P}[X_{i,g,r} = 1] = 1/(2d)$. We therefore have $\mathbb{E}\left[\sum_{i=1}^m X_{i,g,r}\right] = \frac{m}{2d}$, and by a Chernoff bound

$$\mathbb{P}\left[\sum_{i=1}^m X_{i,g,r} < \frac{m}{4d}\right] \leq \exp\left(-\frac{m}{16d}\right) \leq \frac{1}{d^4}.$$

Define W to be the event that for every level $r \in [d]$ and group $g \in [u]$, there are $\geq \frac{m}{4d}$ users in group g with level r. By a union bound, we know

$$\mathbb{P}[W] \geq 1 - \frac{ud}{d^4} \geq 1 - \frac{1}{d^2}$$

since $u = O(\log(d))$. Thus with high probability we have enough users in each level in each group.

We now analyze the quantities Q_r. For each $r \in [d]$, we want to show

$$\mathbb{P}[Q_r = P_r | Q_1 = P_1, ..., Q_{r-1} = P_{r-1}, W] \geq 1 - \frac{1}{d^3},$$

i.e. that the output Q actually matches P bit-by-bit. Conditioning on $Q_1 = P_1, \ldots, Q_{r-1} = P_{r-1}$ and W, $Z_{r,Q+1} = P_r$. Define binary random variable $Y_{i,g,r}$ to be 1 if the bit sent by user i in group g is equal to the j-th bit of P_r and 0 otherwise. If the user i has level r then they send their bit using randomized response. Thus $\mathbb{P}[Y_{i,g,r} = 1] = \frac{e^\varepsilon}{e^\varepsilon + 1}$. If the user i's level is not r, then they send a uniform random bit $\mathbb{P}[Y_{i,g,r} = 1] = 1/2$. Since we conditioned on W, there are $\geq \frac{m}{4d}$ users in group g with level r. Thus

$$\mathbb{E}\left[\sum_{i=1}^{m} Y_{i,g,r}\right] \geq \frac{m}{4d} \cdot \frac{e^\varepsilon}{e^\varepsilon + 1} + \left(m - \frac{m}{4d}\right) \cdot \frac{1}{2}. \tag{7.1}$$

Then we have

$$
\begin{aligned}
&\mathbb{P}\left[Q_r, P_r \text{ have the same } g\text{-th bit} | Q_1 = P_1, ..., Q_{r-1} = P_{r-1}, W\right] \\
&= \mathbb{P}\left[\sum_{i=1}^{m} Y_{i,g,r} > \frac{m}{2}\right] \\
&\geq \mathbb{P}\left[\sum_{i=1}^{m} Y_{i,g,r} > \mathbb{E}\left[\sum_{i=1}^{m} Y_{i,g,r}\right] + \frac{m}{2} - \frac{m}{4d} \cdot \frac{e^{\varepsilon}}{e^{\varepsilon}+1} - \left(m - \frac{m}{4d}\right) \cdot \frac{1}{2}\right] && \text{(Equation 7.1)} \\
&\geq \mathbb{P}\left[\sum_{i=1}^{m} Y_{i,g,r} > \mathbb{E}\left[\sum_{i=1}^{m} Y_{i,g,r}\right] - \frac{m}{8d} \cdot \frac{e^{\varepsilon}-1}{e^{\varepsilon}+1}\right] \\
&= 1 - \exp\left(-\frac{1}{2m} \cdot \left(\frac{m}{8d} \cdot \frac{e^{\varepsilon}-1}{e^{\varepsilon}+1}\right)^2\right) && \text{(Chernoff Bound)} \\
&= 1 - \exp\left(-m \cdot \frac{1}{128d^2} \cdot \frac{(e^{\varepsilon}-1)^2}{(e^{\varepsilon}+1)^2}\right) \\
&\geq 1 - \exp(-4\ln(d)) = 1 - 1/d^4
\end{aligned}
$$

where the last inequality substitutes in the definition of m in Algorithm 7. Union bounding over all u groups yields

$$\mathbb{P}\left[Q_r = P_r | Q_1 = P_1, ..., Q_{r-1} = P_{r-1}, W\right] \geq 1 - \frac{u}{d^4} \geq 1 - \frac{1}{d^3}.$$

Putting this all together, Algorithm 7 outputs $P(Z)$ with probability at least

$$
\begin{aligned}
\mathbb{P}\left[Q_1 = P_1, ..., Q_d = P_d\right] &\geq \mathbb{P}\left[W\right] \cdot \mathbb{P}\left[Q_1 = P_1, ..., Q_d = P_d | W\right] \\
&\geq \mathbb{P}\left[W\right] \cdot \prod_{r=1}^{d} \mathbb{P}\left[Q_r = P_r | Q_1 = P_1, ..., Q_{r-1} = P_{r-1}, W\right] \\
&\geq \left(1 - \frac{1}{d^2}\right) \cdot \left(1 - \frac{1}{d^3}\right)^d \\
&\geq \left(1 - \frac{1}{d^2}\right) \cdot \left(1 - \frac{d}{d^3}\right) \\
&> 1 - \frac{1}{d}.
\end{aligned}
$$

□

Note that Algorithm 7 is k-compositional only for $k \geq d$. The lower bound in the next section shows that any sequentially interactive protocol for the same problem must have a larger sample complexity by a factor of $\tilde{\Omega}(d) = \tilde{\Omega}(k)$. This shows that in general the sample-complexity dependence on k of our original full-to-sequential transformation (Theorem 2) cannot be improved.

We now prove our lower bound for sequentially interactive ε-locally private protocols; the extension to approximate local privacy follows the outline given in Lemma 7.

Theorem 11. *Let $\mathcal{A}$ be a sequentially interactive ε-locally private protocol that, for every instance Z of $\mathcal{MPJ}(d)$, correctly identifies $P(Z)$ with probability $\geq 2/3$. Then $\mathcal{A}$ must have sample complexity $n \geq \frac{d^3}{216(e^\varepsilon - 1)^2 \ln(d)}$.*

Proof. We will prove that any sequentially interactive ε-locally private protocol with $n = \frac{d^3}{216(e^\varepsilon - 1)^2 \ln(d)}$ samples fails to solve $\mathcal{MPJ}(d)$ correctly with probability $> 1/3$ when Z, the collection of correct child labels for each node, is sampled uniformly at random. This is a distributional lower bound which is only stronger than the worst-case lower bound claimed. For notational simplicity, we assume in this argument that all local randomizers have discrete message spaces. However, this assumption is without loss of generality and can be removed (e.g. using Lemma 16).

We will prove our lower bound even for protocols to which we "reveal" some information about the hidden instance Z and users' inputs to the protocol and users. This only makes our lower bound stronger, as the mechanism can ignore this information if desired. Before the protocol starts, each user i publishes a quantity R_i. If $\ell_i \neq 0$ — i.e., if user i is not a "dummy" user — then $R_i = \ell_i$, the user's level. Otherwise R_i is set to be $\lfloor \frac{d(i-1)}{n} \rfloor + 1$. At a high level, we reveal these $\{R_i\}_{i=1}^n$ to break the dependence between Z_i's during the execution of the protocol (see Claim 5 for a formalization of this intuition). Throughout the proof and its claims, we fix realizations $R_1 = r_1, R_2 = r_2, \ldots, R_n = r_n$. We will show that even given such $r_1, ..., r_n$, any sequentially interactive ε-locally private protocol with n

users fails with probability more than 1/3.

For each $i \in [n]$, denote by Π_i the message sent by user i via their local randomizer. Note that there is at most one such message since the protocol is sequentially interactive. We begin with a result about how conditioning on messages and revealed values affects the distribution of Z.

Claim 5. *Suppose $Z_1, ..., Z_d$, the correct child node labels for each level, are sampled from a product distribution. Conditioned on the messages $\Pi_1, ..., \Pi_i$ of the first i users and the revealed values $R_1, \ldots, R_n$, $Z_1, ..., Z_d$ are still distributed according to a product distribution.*

Proof. We proceed via induction on the number of messages i. The base case $i = 0$ is immediate. Now suppose the claim is true for $i - 1$. Use $\mathcal{D}_1 \times \mathcal{D}_2 \times \cdots \mathcal{D}_d$ to denote the product distribution of $Z_1, ..., Z_d$ conditioned on $\Pi_1, ..., \Pi_{i-1}$ and $R_1, \ldots, R_n$ (all quantities that follow are conditioned on $R_1, \ldots, R_n$, and so for notational simplicity we elide the explicit conditioning).

Since the protocol is sequentially interactive, conditioned on $\Pi_1, ..., \Pi_{i-1}$, Π_i depends only on the correct child node labels at level r_i (Z_{r_i}), user i's internal randomness, and their level ℓ_i (recall that when $r_i = \lfloor \frac{i-1}{n/d} \rfloor + 1$, it may be that $\ell_i = 0$ or $\ell_i = r_i$). Therefore, conditioned on $\Pi_1, ..., \Pi_i$, $Z_1, ..., Z_d$ distribute as

$$\mathcal{D}_1 \times \mathcal{D}_2 \times \cdots \times (\mathcal{D}_{r_i} | \Pi_i) \times \cdots \times \mathcal{D}_d,$$

a product distribution. □

We also use induction over levels $\ell \in [d]$ to prove the overall theorem. For each such ℓ, let Δ_ℓ be the following set of distributions on Z.

Definition 24. *For each $\ell \in [d]$, the set Δ_ℓ is composed of distributions $\mathcal{D}$ on Z such that*

1. *$\mathcal{D}$ is a product distribution on $Z_1, ..., Z_d$,*

2. *for each* $i = 1, ..., d - \ell$, Z_i *is deterministically fixed to be* z_i, *and*

3. *since* $Z_1, ..., Z_{d-\ell}$ *are fixed, by the definition of* $\mathcal{MPJ}$, $P_1, ..., P_{d-\ell}$ *are also fixed to some* $p_1, ..., p_{d-\ell}$. *The marginal distribution on* $Z_{|p_1,...,p_{d-\ell}}$ *is the uniform distribution.*

In the induction step, we consider sequentially interactive locally private protocols with fewer users. The idea is that for any sequentially interactive ε-locally private protocol on n users, if we fix the messages of the first i users, then what remains is a sequentially interactive ε-locally private protocol on $n - i$ users. Accordingly, we want to lower bound the failure probability of this remaining protocol. More concretely:

Inductive statement: Any sequentially interactive ε-locally private protocol with $n \cdot \frac{\ell}{d}$ users (the $\left(n \cdot \frac{d-\ell}{d} + 1\right)$-th user to the n-th user) fails to solve $\mathcal{MPJ}(d)$ correctly with probability $> \frac{2}{3} - \frac{\ell}{3d}$ when the collection of correct child node labels Z is sampled from a distribution in Δ_ℓ.

It will be easier to establish the inductive case first and then treat the base case afterward.

Induction step ($\ell > 1$): Assume the above statement is true for $\ell - 1$.

In this induction, let $\mathcal{A}$ be a sequentially interactive ε-locally private protocol with $n \cdot \frac{\ell}{d}$ users and let $\mathcal{D}$ be the distribution generating Z before $\mathcal{A}$ starts. Let Π be the messages sent by the first n/d users of $\mathcal{A}$ (the $\left(n \cdot \frac{d-\ell}{d} + 1\right)$-th user to the $\left(n \cdot \frac{d-\ell+1}{d}\right)$-th user) and let $\mathcal{A}_\pi$ be the sequentially interactive ε-locally private protocol with $n \cdot \frac{\ell-1}{d}$ users conditioned on $\Pi = \pi$. For notational convenience, define $n_\ell = n \cdot \frac{d-\ell}{d}$, $\Pi_{<i} = \Pi_{n_\ell+1}, ..., \Pi_{i-1}$ and $\Pi_{\leq i} = \Pi_{n_\ell+1}, ..., \Pi_i$.

For each prefix of messages, π, let $\mathcal{D}'(\pi)$ be some mixture of distributions in $\Delta_{\ell-1}$ (to be specified later). By the induction hypothesis on $\ell - 1$,

$$\mathbb{P}_{Z \sim \mathcal{D}'(\pi)} \left[\mathcal{A}_\pi \text{ outputs } P(Z)\right] < \frac{1}{3} + \frac{\ell - 1}{3d}.$$

Thus $\mathbb{P}_{Z\sim\mathcal{D}}[\mathcal{A} \text{ outputs } P(Z)]$

$$
\begin{aligned}
&= \sum_{\pi} \mathbb{P}[\Pi = \pi] \cdot \mathbb{P}_{Z\sim(\mathcal{D}|\Pi=\pi)}[\mathcal{A}_\pi \text{ outputs } P(Z)] \\
&\leq \sum_{\pi} \mathbb{P}[\Pi = \pi] \cdot \left(\mathbb{P}_{Z\sim\mathcal{D}'(\pi)}[\mathcal{A}_\pi \text{ outputs } P(Z)] + \|(\mathcal{D} \mid \Pi = \pi) - \mathcal{D}'(\pi)\|_1\right) \\
&< \frac{1}{3} + \frac{\ell - 1}{3d} + \sum_{\pi} \mathbb{P}[\Pi = \pi] \cdot \|(\mathcal{D} \mid (\Pi = \pi)) - \mathcal{D}'(\pi)\|_1 \qquad (7.2)
\end{aligned}
$$

Recall that we want to show $\mathbb{P}_{Z\sim\mathcal{D}}[\mathcal{A} \text{ outputs } P(Z)] \leq \frac{1}{3} + \frac{\ell}{3d}$. It therefore suffices to bound the sum in Equation 7.2 as

$$\sum_{\pi} \mathbb{P}[\Pi = \pi] \cdot \|(\mathcal{D} \mid (\Pi = \pi)) - \mathcal{D}'(\pi)\|_1 \leq \frac{1}{3d}.$$

We show this via Claims 6, 7, and 8. We finally define $\mathcal{D}'(\pi)$ in Claim 8.

First we define some notation for the path we need to reason about. Since $\mathcal{D} \in \Delta_\ell$, by the definition of Δ_ℓ we know that for $Z \sim \mathcal{D}$, the first $d - \ell$ levels of the tree $Z_1, ..., Z_{d-\ell}$ deterministically take fixed values $z_1, ..., z_{d-\ell}$. Thus, the first $d - \ell$ nodes in the path $P_1, ..., P_{d-\ell}$ marked by Z are also fixed to take particular values $p_1, ..., p_{d-\ell}$. For the induction step, we write $P = P_1, ..., P_{d-\ell+1}$ to denote the first $d - \ell + 1$ vertices of the path. Since $P_{d-\ell+1}$ is the only value that is not fixed from Δ_ℓ, and the path is through an s-ary tree, P can take on at most s different possible values and is determined by $Z_{d-\ell+1}$.

In the first claim, we show that after observing the messages sent by n/d agents, uncertainty still remains about P.

Claim 6. *For $i \in \{n_\ell + 1, ..., n_\ell + n/d\}$,*

$$\sum_{\pi_{\leq i}} \mathbb{P}[\Pi_{\leq i} = \pi_{\leq i}] \cdot \left(\max_p \mathbb{P}[P = p \mid \Pi_{\leq i} = \pi_{\leq i}]\right) \leq \frac{3}{d^4}.$$

Proof. Denoting by $\mathbb{1}[E]$ the indicator function for event E,

$$\begin{aligned}
&\sum_{\pi_{\le i}} \mathbb{P}[\Pi_{\le i} = \pi_{\le i}] \cdot \left(\max_p \mathbb{P}[P = p \mid \Pi_{\le i} = \pi_{\le i}] \right) \\
\le &\sum_{\pi_{\le i}} \mathbb{P}[\Pi_{\le i} = \pi_{\le i}] \cdot \left(\mathbb{1}\left[\max_p \mathbb{P}[P = p \mid \Pi_{\le i} = \pi_{\le i}] > \frac{2}{s} \right] \right. \\
&\left. + \ \mathbb{1}\left[\max_p \mathbb{P}[P = p \mid \Pi_{\le i} = \pi_{\le i}] \le \frac{2}{s} \right] \cdot \frac{2}{s} \right) \\
\le &\frac{2}{s} + \sum_{\pi_{\le i}} \mathbb{P}[\Pi_{\le i} = \pi_{\le i}] \cdot \left(\mathbb{1}\left[\max_p \mathbb{P}[P = p \mid \Pi_{\le i} = \pi_{\le i}] > \frac{2}{s} \right] \right) \\
\le &\frac{2}{s} + \sum_p \sum_{\pi_{\le i}} \mathbb{P}[\Pi_{\le i} = \pi_{\le i}] \cdot \left(\mathbb{1}\left[\mathbb{P}[P = p \mid \Pi_{\le i} = \pi_{\le i}] > \frac{2}{s} \right] \right). \qquad (7.3)
\end{aligned}$$

Now consider some specific path p. We know that

$$\begin{aligned}
\mathbb{P}[P = p \mid \Pi_{\le i} = \pi_{\le i}] &= \frac{\mathbb{P}[P = p, \Pi_{\le i} = \pi_{\le i}]}{\mathbb{P}[\Pi_{\le i} = \pi_{\le i}]} \\
&= \mathbb{P}[P = p] \cdot \frac{\mathbb{P}[\Pi_{\le i} = \pi_{\le i} \mid P = p]}{\mathbb{P}[\Pi_{\le i} = \pi_{\le i}]} && \text{(Bayes' rule)} \\
&= \frac{1}{s} \cdot \frac{\mathbb{P}[\Pi_{\le i} = \pi_{\le i} \mid P = p]}{\mathbb{P}[\Pi_{\le i} = \pi_{\le i}]}. && \text{(Uniformity of } P\text{)}
\end{aligned}$$

For $j = n_\ell + 1, ..., i$, define random variable

$$X_j = \ln\left(\frac{\mathbb{P}[\Pi_j \mid \Pi_{<j}, P = p]}{\mathbb{P}[\Pi_j \mid \Pi_{<j}]} \right).$$

We want to upper bound the quantity in Equation 7.3. We do so using these X_j. Recall that r_j is user j's level if that level is non-zero, i.e. user j is not a "dummy" user. Otherwise r_j is $d - \ell + 1$ for $j = n_\ell + 1, ..., n_\ell + n/d$. If $r_j \neq d - \ell + 1$, by Claim 5, we know that conditioned on $\Pi_{<j}$, Π_j is independent of P. Therefore when $r_j \neq d - \ell + 1$, $X_j = \ln(1) = 0$.

If instead $r_j = d - \ell + 1$, we know the level ℓ_j of the user j is 0 with probability $\frac{d}{d+1}$ and $d - \ell + 1$ with probability $\frac{1}{d+1}$. If $\ell_j = 0$, then the user is a "dummy", has no private data about P, and Π_j is independent of P conditioned on $\Pi_{<j}$. Call the input distribution of the j-th user q_j. Here, we give (a slightly modified version of) Lemmas 3 and 4 from Duchi

et al. [32].

Lemma 22. *Let m_1 and m_2 be the output distributions of an ε-randomizer in a sequentially interactive protocol given, respectively, input distributions $q_j \mid \Pi_{<j}, P = p$ and $q_j \mid \Pi_{<j}$. Then*

$$\left|\ln\left(\frac{m_1(z)}{m_2(z)}\right)\right| \leq \min(2, e^{\varepsilon})(e^{\varepsilon} - 1) \cdot ||(q_j \mid \Pi_{<j}, P = p) - (q_j \mid \Pi_{<j})||_{TV}.$$

We know that $||(q_j|\Pi_{<j} = \pi_{<j}, P = p) - (q_j|\Pi_{<j} = \pi_{<j})||_{TV} \leq \frac{1}{d+1}$, as the difference stems from the event $\ell_j = d - \ell + 1$. Thus, by Lemma 22

$$|X_j| \leq \frac{2(e^{\varepsilon} - 1)}{d+1} < \frac{2(e^{\varepsilon} - 1)}{d}.$$

Next, we bound the conditional expectation of X_j:

$$\begin{aligned}
\mathbb{E}\left[X_j \mid \Pi_{<j} = \pi_{<j}\right] &= \sum_{\pi_j} \mathbb{P}\left[\Pi_j = \pi_j | \Pi_{<j} = \pi_{<j}\right] \cdot \ln\left(\frac{\mathbb{P}\left[\Pi_j = \pi_j \mid \Pi_{<j} = \pi_{<j}, P = p\right]}{\mathbb{P}\left[\Pi_j = \pi_j | \Pi_{<j} = \pi_{<j}\right]}\right) \\
&= -D_{KL}\left((\Pi_j \mid \Pi_{<j} = \pi_{<j}, P = p)||(\Pi_j \mid \Pi_{<j} = \pi_{<j})\right) \\
&\leq 0.
\end{aligned}$$

Therefore $X_{n_\ell+1}, X_{n_\ell+1} + X_{n_\ell+2}, ..., X_{n_\ell+1} + \cdots + X_i$ form a supermartingale. Next, we use the above bounds on these X_j to control their sum using the Azuma-Hoeffding inequality:

$$\begin{aligned}
\mathbb{P}\left[X_{n_\ell+1} + \cdots + X_i > \ln(2)\right] &\leq \exp\left(-\frac{\ln^2(2)}{2(2(e^{\varepsilon} - 1)/d)^2(i - n_\ell)}\right) \\
&\leq \exp\left(-\frac{\ln^2(2)}{2(2(e^{\varepsilon} - 1)/d)^2(n/d)}\right) \\
&\leq \frac{1}{d^8} = \frac{1}{sd^4}
\end{aligned}$$

since Z is an s-ary tree with $s = d^4$. Next,

$$\begin{aligned} X_{n_\ell+1} + \cdots + X_i &= \sum_{j=n_\ell+1}^{i} \ln\left(\frac{\mathbb{P}\left[\Pi_j \mid \Pi_{<j}, P = p\right]}{\mathbb{P}\left[\Pi_j \mid \Pi_{<j}\right]}\right) \\ &= \ln\left(\prod_{j=n_\ell+1}^{i} \frac{\mathbb{P}\left[\Pi_j \mid \Pi_{<j}, P = p\right]}{\mathbb{P}\left[\Pi_j \mid \Pi_{<j}\right]}\right) \\ &= \ln\left(\frac{\mathbb{P}\left[\Pi_{\le i} \mid P = p\right]}{\mathbb{P}\left[\Pi_{\le i}\right]}\right) && \text{(Chain rule)} \\ &= \ln\left(s \cdot \mathbb{P}\left[P = p \mid \Pi_{\le i}\right]\right) \end{aligned}$$

where the last step uses Bayes' rule and the uniformity of P over the s possible values for the last step in the path P. Therefore

$$\begin{aligned} &\sum_{\pi_{\le i}} \mathbb{P}\left[\Pi_{\le i} = \pi_{\le i}\right] \cdot \left(\mathbb{1}\left[\mathbb{P}\left[P = p \mid \Pi_{\le i} = \pi_{\le i}\right] > \frac{2}{s}\right]\right) \\ &= \sum_{\pi_{\le i}} \mathbb{P}\left[\Pi_{\le i} = \pi_{\le i}\right] \cdot \left(\mathbb{1}\left[s \cdot \mathbb{P}\left[P = p \mid \Pi_{\le i} = \pi_{\le i}\right] > 2\right]\right) \\ &= \mathbb{P}\left[X_{n_\ell+1} + \cdots + X_i > \ln(2)\right] \\ &\le \frac{1}{sd^4}. \end{aligned}$$

Tracing the above inequality back through Equation 7.3, we have

$$\begin{aligned} \sum_{\pi_{\le i}} \mathbb{P}\left[\Pi_{\le i} = \pi_{\le i}\right] \cdot \left(\max_p \mathbb{P}\left[P = p \mid \Pi_{\le i} = \pi_{\le i}\right]\right) &\le (7.3) \\ &\le \frac{2}{s} + s \cdot \frac{1}{sd^4} \\ &= \frac{3}{d^4}. \end{aligned}$$

□

We now proceed to Claim 7. Here, we bound the information Π contains about $Z_{|P}$. Intuitively, by Claim 6 users have little information about P, and as a result they cannot

know which potential subtree $Z_{|p}$ to focus their privacy budget on.

Claim 7.

$$\sum_p \mathbb{P}[P = p] \cdot I(\Pi; Z_{|p} \mid P = p) \leq \frac{1}{18d^2}.$$

Proof. By the inductive hypothesis, Z is sampled from $\mathcal{D} \in \Delta_\ell$. Define $Z_{|<p}$ to be $Z_{|p_1,\ldots,p_{d-\ell},0}, \ldots, Z_{|p_1,\ldots,p_{d-\ell},p_{d-\ell+1}-1}$. By the definition of Δ_ℓ, we know $Z_{|<p}$ and $Z_{|p}$ are independent given P, so $I(Z_{|<p}; Z_{|p} \mid P = p) = 0$. Therefore by the chain rule for mutual information, we get

$$\begin{aligned} I(\Pi; Z_{|p} \mid P = p) &\leq I(\Pi, Z_{|<p}; Z_{|p} \mid P = p) \\ &= I(Z_{|<p}; Z_{|p} \mid P = p) + I(\Pi; Z_{|p} \mid P = p, Z_{|<p}) \\ &= I(\Pi; Z_{|p} \mid P = p, Z_{|<p}). \end{aligned}$$

The main step of the proof is to compare $I(\Pi_i; Z_{|p} \mid P = p, \Pi_{<i} = \pi_{<i}, Z_{|<p})$ and $I(\Pi_i; Z_{|p} \mid \Pi_{<i} = \pi_{<i}, Z_{|<p})$, i.e. quantify the effect of the event $P = p$ on the mutual information between Π and $Z_{|p}$. First, by Claim 5, conditioning on $\Pi_{<i} = \pi_{<i}$ induces a product distribution for $Z_1, \ldots, Z_d$. We also know that (as mentioned in the proof of Claim 5) conditioned on $\Pi_{<i} = \pi_{<i}$, Π_i only depends on Z_{r_i}, the internal randomness of the user i, and their level ℓ_i. By item 3 in the definition of Δ_ℓ, P only depends on $Z_{d-\ell+1}$. We prove

$$I(\Pi_i; Z_{|p} \mid P = p, \Pi_{<i} = \pi_{<i}, Z_{|<p}) = I(\Pi_i; Z_{|p} \mid \Pi_{<i} = \pi_{<i}, Z_{|<p}). \tag{7.4}$$

There are two cases depending on r_i.

- When $r_i \leq d - \ell + 1$, user i either has $\ell_i \leq d - \ell + 1$ or is a "dummy" user. Therefore, whether or not we condition on $P = p$, user i does not have any information about $Z_{|p}$ or $Z_{|<p}$. Thus Π_i is independent of $Z_{|p}, Z_{|<p}$, so

$$I(\Pi_i; Z_{|p} \mid P = p, \Pi_{<i} = \pi_{<i}, Z_{|<p}) = 0 = I(\Pi_i; Z_{|p} \mid \Pi_{<i} = \pi_{<i}, Z_{|<p}).$$

- When $r_i > d - \ell + 1$, once we've conditioned on $\Pi_{<i} = \pi_{<i}$, additionally conditioning on $P = p$ does not change the joint distribution of $Z_{d-\ell+2}, ..., Z_d$. This is because $P = P_1, \ldots, P_{d-\ell+1}$ and by above conditioning on $\Pi_{<i} = \pi_{<i}$ induces a product distribution on $Z_1, \ldots, Z_d$ (and in particular on $Z_{d-\ell+2}, ..., Z_d$). It follows that conditioning on $P = p$ does not change the joint distribution of $Z_{|p}, Z_{|<p}, \Pi_i$. Thus

$$I(\Pi_i; Z_{|p} | P = p, \Pi_{<i} = \pi_{<i}, Z_{|<p}) = I(\Pi_i; Z_{|p} | \Pi_{<i} = \pi_{<i}, Z_{|<p}).$$

Putting things together, we have

$$\begin{aligned}
&\sum_p \mathbb{P}[P = p] \cdot I(\Pi; Z_{|p} \mid P = p) \\
\leq &\sum_p \mathbb{P}[P = p] \cdot I(\Pi; Z_{|p} \mid P = p, Z_{|<p}) \\
= &\sum_p \sum_{i=n_\ell+1}^{n_\ell+n/d} \mathbb{P}[P = p] \cdot I(\Pi_i; Z_{|p} \mid P = p, \Pi_{<i}, Z_{|<p}) \\
= &\sum_{i=n_\ell+1}^{n_\ell+n/d} \sum_{\pi_{<i}} \sum_p \mathbb{P}[P = p] \cdot \mathbb{P}[\Pi_{<i} = \pi_{<i} \mid P = p] \cdot I(\Pi_i; Z_{|p} \mid P = p, \Pi_{<i} = \pi_{<i}, Z_{|<p}) \\
= &\sum_{i=n_\ell+1}^{n_\ell+n/d} \sum_{\pi_{<i}} \sum_p \mathbb{P}[\Pi_{<i} = \pi_{<i}] \cdot \mathbb{P}[P = p \mid \Pi_{<i} = \pi_{<i}] \cdot I(\Pi_i; Z_{|p} \mid P = p, \Pi_{<i} = \pi_{<i}, Z_{|<p}) \\
= &\sum_{i=n_\ell+1}^{n_\ell+n/d} \sum_{\pi_{<i}} \sum_p \mathbb{P}[\Pi_{<i} = \pi_{<i}] \cdot \mathbb{P}[P = p \mid \Pi_{<i} = \pi_{<i}] \cdot I(\Pi_i; Z_{|p}| \mid \Pi_{<i} = \pi_{<i}, Z_{|<p}) \\
\leq &\sum_{i=n_\ell+1}^{n_\ell+n/d} \sum_{\pi_{<i}} \left(\sum_p \mathbb{P}[\Pi_{<i} = \pi_{<i}] \cdot I(\Pi_i; Z_{|p} \mid \Pi_{<i} = \pi_{<i}, Z_{|<p}) \right) \cdot \left(\max_p \mathbb{P}[P = p \mid \Pi_{<i} = \pi_{<i}] \right) \\
\leq &\sum_{i=n_\ell+1}^{n_\ell+n/d} \sum_{\pi_{<i}} \mathbb{P}[\Pi_{<i} = \pi_{<i}] \cdot I(\Pi_i; Z \mid \Pi_{<i} = \pi_{<i}) \cdot \left(\max_p \mathbb{P}[P = p \mid \Pi_{<i} = \pi_{<i}] \right). \qquad (7.5)
\end{aligned}$$

where the third equality uses Bayes' rule and the fourth equality uses Equation 7.4.

We now bound $I(\Pi_i; Z | \Pi_{<i} = \pi_{<i})$ using Theorem 1 from Duchi et al. [32], simplified here as Lemma 23, itself a version of Lemma 9.

Lemma 23. *Let Π be the distribution over randomizer outputs for an ε-local randomizer with inputs drawn from a distribution family parametrized by $\mathcal{V}$. Then $I(\Pi; \mathcal{V}) \leq 4(e^{\varepsilon} - 1)^2$.*

In particular, the proof of Lemma 23 implies that $I(\Pi_i; Z | \Pi_{<i} = \pi_{<i}) \leq 4(e^{\varepsilon} - 1)^2$. We continue our chain of inequalities:

$$\begin{aligned}
(7.5) &\leq \sum_{i=n_\ell+1}^{n_\ell+n/d} \sum_{\pi_{<i}} \mathbb{P}\left[\Pi_{<i} = \pi_{<i}\right] \cdot 4(e^{\varepsilon} - 1)^2 \cdot \left(\max_p \mathbb{P}\left[P = p \mid \Pi_{<i} = \pi_{<i}\right]\right) \\
&\leq \frac{n}{d} \cdot (e^{\varepsilon} - 1)^2 \cdot \frac{12}{d^4} && \text{(Claim 6)} \\
&\leq \frac{1}{18d^2}
\end{aligned}$$

since the overall (theorem-level) proof uses $n = \frac{d^3}{216(e^{\varepsilon}-1)^2 \ln(d)}$ □

In our last claim, we convert the bound on mutual information from Claim 7 into a bound on the L_1 distance between distributions.

Claim 8. *There exists a distribution $\mathcal{D}'(\pi)$ which is a mixture of distributions in $\Delta_{\ell-1}$ for each π such that*

$$\sum_{\pi} \mathbb{P}\left[\Pi = \pi\right] \cdot \|(\mathcal{D} \mid (\Pi = \pi)) - \mathcal{D}'(\pi)\|_1 \leq \frac{1}{3d}.$$

Proof. By the definition of mutual information in terms of KL-divergence (Fact 7),

$$I(\Pi; Z_{|p} \mid P = p) = D_{KL}\left(\mathbb{P}\left[\Pi, Z_{|p} \mid P = p\right] || \mathbb{P}\left[\Pi \mid P = p\right] \cdot \mathbb{P}\left[Z_{|p} \mid P = p\right]\right).$$

Next, by Pinsker's inequality (Fact 5),

$$\begin{aligned}
&\sum_{\pi, z_{|p}} \left|\mathbb{P}\left[\Pi = \pi, Z_{|p} = z_{|p} \mid P = p\right] - \mathbb{P}\left[\Pi = \pi \mid P = p\right] \cdot \mathbb{P}\left[Z_{|p} = z_{|p} \mid P = p\right]\right| \\
\leq &\sqrt{2 D_{KL}\left(\mathbb{P}\left[\Pi, Z_{|p} \mid P = p\right] || \mathbb{P}\left[\Pi \mid P = p\right] \cdot \mathbb{P}\left[Z_{|p} \mid P = p\right]\right)}.
\end{aligned}$$

Define

$$U = \sum_p \mathbb{P}\left[P = p\right] \sum_{\pi, z_{|p}} \left| \mathbb{P}\left[\Pi = \pi, Z_{|p} = z_{|p}\right| \mid P = p\right] - \mathbb{P}\left[\Pi = \pi \mid P = p\right] \mathbb{P}\left[Z_{|p} = z_{|p} \mid P = p\right] \Big| .$$

Then, using our previous application of Pinsker's inequality, we upper bound U by

$$\begin{aligned}
U &\leq \sum_p \mathbb{P}\left[P = p\right] \sqrt{2 D_{KL}\left(\mathbb{P}\left[\Pi, Z_{|p} \mid P = p\right] || \mathbb{P}\left[\Pi \mid P = p\right] \mathbb{P}\left[Z_{|p} \mid P = p\right]\right)} \\
&= \sum_p \mathbb{P}\left[P = p\right] \sqrt{2 I(\Pi; Z_{|p} \mid P = p)} && \text{(definition of mutual information)} \\
&\leq \sqrt{2 \sum_p \mathbb{P}\left[P = p\right] \cdot 2 I(\Pi; Z_{|p} \mid P = p)} && \text{(Jensen's inequality and concavity of } \sqrt{\cdot}) \\
&\leq \frac{1}{3d}. && \text{(Claim 7)}
\end{aligned}$$

Define $A = \sum_p \mathbb{P}\left[P = p\right]$ and $B = \sum_\pi \mathbb{P}\left[\Pi = \pi\right]$. Then we can also lower bound U by

$$\begin{aligned}
& A \sum_{\pi, z_{|p}} \left| \mathbb{P}\left[\Pi = \pi, Z_{|p} = z_{|p} \mid P = p\right] - \mathbb{P}\left[\Pi = \pi \mid P = p\right] \mathbb{P}\left[Z_{|p} = z_{|p} \mid P = p\right] \right| \\
&= A \sum_\pi \mathbb{P}\left[\Pi = \pi \mid P = p\right] \sum_{z_{|p}} \left| \mathbb{P}\left[Z_{|p} = z_{|p} \mid \Pi = \pi, P = p\right] - \mathbb{P}\left[Z_{|p} = z_{|p} \mid P = p\right] \right| \\
&= B \sum_p \mathbb{P}\left[P = p \mid \Pi = \pi\right] \sum_{z_{|p}} \left| \mathbb{P}\left[Z_{|p} = z_{|p} \mid \Pi = \pi, P = p\right] - \mathbb{P}\left[Z_{|p} = z_{|p} \mid P = p\right] \right| \quad (7.6)
\end{aligned}$$

since $\mathbb{P}\left[P = p\right] \cdot \mathbb{P}\left[\Pi = \pi \mid P = p\right] = \mathbb{P}\left[\Pi = \pi\right] \cdot \mathbb{P}\left[P = p \mid \Pi = \pi\right]$. We continue the chain by multiplying each innermost term by

$$1 = \sum_z \mathbb{P}\left[Z = z \mid \Pi = \pi, P = p, Z_{|p} = z_{|p}\right]$$

and then use the triangle inequality to rearrange the order of summation and get

$$(7.6) \geq B \cdot \sum_p \mathbb{P}\left[P = p \mid \Pi = \pi\right] \cdot C \quad (7.7)$$

where

$$C = \sum_z \left| \mathbb{P}\left[Z = z \mid \Pi = \pi, P = p\right] - \mathbb{P}\left[Z_{|p} = z_{|p} \mid P = p\right] \mathbb{P}\left[Z = z \mid \Pi = \pi, P = p, Z_{|p} = z_{|p}\right] \right|.$$

We then repeat our application of the triangle inequality and summation rearrangement to get

$$(7.7) \geq B \cdot \sum_z \left| \mathbb{P}\left[Z = z \mid \Pi = \pi\right] - D \right| \tag{7.8}$$

where

$$D = \sum_p \mathbb{P}\left[P = p \mid \Pi = \pi\right] \cdot \mathbb{P}\left[Z_{|p} = z_{|p} \mid P = p\right] \mathbb{P}\left[Z = z \mid \Pi = \pi, P = p, Z_{|p} = z_{|p}\right] \Big|.$$

Now, define $\mathcal{D}'(\pi)$ to be the distribution on Z such that for all z, $\mathbb{P}_{Z \sim \mathcal{D}'(\pi)}\left[Z = z\right] = D$. Equivalently, $Z \sim \mathcal{D}'(\pi)$ is sampled through the following procedure: (1) sample P according to $P \mid \Pi = \pi$, (2) sample $Z_{|p}$ according to $Z_{|p} \mid P = p$, and (3) sample Z according to $Z \mid \Pi = \pi, P = p, Z_{|p} = z_{|p}$.

Note that $\mathbb{P}_{Z \sim \mathcal{D}|(\Pi = \pi)}\left[Z = z\right] = \mathbb{P}\left[Z = z \mid \Pi = \pi\right]$ for all z. Thus, we connect the above chain of inequalities using U, A, B, C, D to $\|(\mathcal{D} \mid (\Pi = \pi)) - \mathcal{D}'(\pi)\|_1$ by

$$\|(\mathcal{D} \mid (\Pi = \pi)) - \mathcal{D}'(\pi)\|_1 = \sum_z \left| \mathbb{P}\left[Z = z \mid \Pi = \pi\right] - D \right|$$

and substitute this into (7.8) to get

$$\sum_\pi \mathbb{P}\left[\Pi = \pi\right] \cdot \|(\mathcal{D} \mid (\Pi = \pi)) - \mathcal{D}'(\pi)\|_1 \leq \frac{1}{3d}.$$

□

It remains to show that $\mathcal{D}'(\pi)$ is a mixture of distributions in $\Delta_{\ell-1}$; doing so will complete our proof of the original inductive step. We will show that for any $z_1, ..., z_{d-\ell+1}$ such that $\mathbb{P}_{Z \sim \mathcal{D}'(\pi)}\left[Z_1, ..., Z_{d-\ell+1} = z_1, ..., z_{d-\ell+1}\right] \neq 0$, $\mathcal{D}'(\pi) \mid (Z_1, ..., Z_{d-\ell+1} = z_1, ..., z_{d-\ell+1})$ is a

distribution in $\Delta_{\ell-1}$. Recalling that membership in $\Delta_{\ell-1}$ requires meeting three conditions (Definition 24) we verify these conditions below.

1. By Claim 5, we know $\mathcal{D} \mid (\Pi = \pi)$ is a product distribution on $Z_1, ..., Z_d$. As $\mathcal{D}'(\pi)$ is sampled according to $\mathcal{D} \mid (\Pi = \pi)$, $\mathcal{D}'(\pi)$ is also a product distribution on $Z_1, ..., Z_d$. After further conditioning, $\mathcal{D}'(\pi) \mid (Z_1, ..., Z_{d-\ell+1} = z_1, ..., z_{d-\ell+1})$ remains a product distribution on $Z_1, ..., Z_d$.

2. Since we draw the final Z conditioned on $Z_{|p} = z_{|p}$, Z_i is deterministically fixed for $i = 1, \ldots, d - \ell$.

3. First, note that the marginal distribution of $\mathcal{D} \mid (P = p)$ on $Z_{|p}$ is uniform since $D \mid (\Pi = \pi)$ induces a product distribution on $Z_1, \ldots, Z_d$, and conditioning on $P = p$ only fixes $Z_{\leq d-\ell+1}$ and leaves $Z_{d-\ell+2} \times \cdots \times Z_d$ as a product distribution. Thus

$$\mathbb{P}_{Z \sim \mathcal{D}'(\pi) \mid (Z_{\leq d-\ell+1} = z_{\leq d-\ell+1})} \left[Z_{|p} = z_{|p} \right] = \mathbb{P} \left[Z_{|p} = z_{|p} \mid P = p \right]$$

   and the marginal distribution of $\mathcal{D}'(\pi) \mid (Z_1, ..., Z_{d-\ell+1} = z_1, ..., z_{d-\ell+1})$ on $Z_{|p}$ is also the uniform distribution.

Therefore $\mathcal{D}'(\pi) \mid (Z_1, ..., Z_{d-\ell+1} = z_1, ..., z_{d-\ell+1})$ is a distribution in $\Delta_{\ell-1}$ and $\mathcal{D}'(\pi)$ is a mixture of distributions in $\Delta_{\ell-1}$.

Base case ($\ell = 1$): We finally discuss the base case of our induction. Define $\mathcal{A}$, Π and P as in the induction step. Since the output of $\mathcal{A}$ is a function of Π,

$$\mathbb{P} \left[\mathcal{A} \text{ outputs } P(Z) \right] \leq \sum_{\pi} \mathbb{P} \left[\Pi = \pi \right] \cdot \max_{p} \mathbb{P} \left[P = p \mid \Pi = \pi \right].$$

Since Claim 6 also applies to the base case, we get

$$\mathbb{P} \left[\mathcal{A} \text{ outputs } P(Z) \right] \leq \frac{3}{d^4} < \frac{1}{3} < \frac{1}{3} + \frac{1}{3d}.$$

This completes our induction, and the overall proof. □

Chapter 8

Polynomial Separation: Central, Pan-, and Local Privacy

We now turn to our final result, which separates all three of central, pan-, and (sequentially interactive) local privacy for the problem of *uniformity testing*.

8.1. Additional Preliminaries and Related Work

In uniformity testing, a tester receives i.i.d. sample access to an unknown discrete distribution p over $[k]$ and must determine with nontrivial constant probability whether p is uniform or α-far from uniform in total variation distance. Below, let U_k denote the uniform distribution over $[k]$.

Definition 25 (Uniformity testing)**.** *An algorithm $\mathcal{A}$ is a* uniformity tester on m samples *if, given m i.i.d. samples from p,*

1. *when $p = U_k$, with probability $\geq 2/3$ $\mathcal{A}$ outputs "uniform", and*
2. *when $||p - U_k||_{TV} \geq \alpha$, with probability $\geq 2/3$ $\mathcal{A}$ outputs "non-uniform".*

The specific choice of $2/3$, while common in uniformity testing, is arbitrary. The important point is that there is a constant separation between output probabilities,

$$\mathbb{P}\left[\text{output uniform} \mid p = U_k\right] \geq 2/3 \text{ and } \mathbb{P}\left[\text{output uniform} \mid ||p - U_k||_{TV} \geq \alpha\right] \leq 1/3.$$

As long as we achieve constant separation, i.e. have $\mathbb{P}\left[\text{output uniform} \mid p = U_k\right] \geq c_1$ and $\mathbb{P}\left[\text{output uniform} \mid ||p - U_k||_{TV} \geq \alpha\right] \leq c_2$ for positive $c_1 - c_2 = \Omega(1)$, we can amplify it to a $1/3$ separation by repetition. After sufficiently many repetitions, if $p = U_k$ then the proportion of "uniform" answers will concentrate at or above c_1, and if $||p - U_k||_{TV} \geq \alpha$ it

will concentrate at or below c_2. By a Chernoff bound, $r = \Omega\left(\frac{1}{(c_1-c_2)^2}\right)$ repetitions suffice to distinguish between these cases. Since this is still a constant number of repetitions, our algorithms will focus on achieving any constant separation.

Furthermore, like many uniformity testers, ours will employ *Poissonization*. A Poissonized uniformity tester draws $m' \sim \mathsf{Poisson}(m)$ samples instead of just m. The result is that, over the randomness of m', the counts of samples of each element in $[k]$ are independent. This independence will be useful when analyzing the utility of our testers. However, since $\mathsf{Poisson}(m)$ concentrates around m [22], we can guarantee $m' = O(m)$ at the cost of a constant decrease in success probability. Since we focus on constant success probability separation as outlined in the previous paragraph, we elide the distinction between "draws m samples" and "draws $\mathsf{Poisson}(m)$ samples" in our sample complexity results.

A line of work [42, 57, 61] has established that uniformity testing (without privacy) has sample complexity $\Theta\left(\frac{\sqrt{k}}{\alpha^2}\right)$ where k is the domain size and α is the total variation distance parameter (for more information on testing, see the survey by Canonne [21]). Acharya, Sun, and Zhang [3] showed that ε-centrally private uniformity testing has sample complexity $\Theta\left(\frac{\sqrt{k}}{\alpha^2} + \frac{\sqrt{k}}{\alpha\sqrt{\varepsilon}} + \frac{k^{1/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{1}{\alpha\varepsilon}\right)$. Acharya, Canonne, Freitag, and Tyagi [4] showed that noninteractive ε-locally private uniformity testing has sample complexity $\Theta\left(\frac{k}{\alpha^2\varepsilon^2}\right)$. A comparison of our results [8] to this previous work appears in Figure 2.

8.2. Pan-Private Upper Bound

We now present our pan-private uniformity testers. In the first subsection, we give a suboptimal uniformity tester SimplePanTest. SimplePanTest is a warmup and eventual building block for a better algorithm, PanTest, in the subsequent section.

8.2.1. Warmup: SimplePanTest

Like many uniformity testers, SimplePanTest computes a statistic on the data and compares it to a threshold. The statistic is designed to be small when p is uniform and large if

Setting	Previous Work	This Work
Non-private	$\Theta\left(\frac{\sqrt{k}}{\alpha^2}\right)$ ([42, 57, 61])	–
ε-central privacy	$\Theta\left(\frac{\sqrt{k}}{\alpha^2} + \frac{\sqrt{k}}{\alpha\sqrt{\varepsilon}} + \frac{k^{1/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{1}{\alpha\varepsilon}\right)$ ([3])	–
ε-pan-privacy	– –	$O\left(\frac{k^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{\sqrt{k}}{\alpha^2} + \frac{\sqrt{k}}{\alpha\varepsilon}\right)$ $\Omega\left(\frac{k^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{\sqrt{k}}{\alpha^2} + \frac{1}{\alpha\varepsilon}\right)$
SI ε-local privacy	$O\left(\frac{k}{\alpha^2\varepsilon^2}\right)$ ([4])	$\Omega\left(\frac{k}{\alpha^2\varepsilon^2}\right)$
NI ε-local privacy	$\Theta\left(\frac{k}{\alpha^2\varepsilon^2}\right)$ ([4])	–

Table 2: A comparison of the uniformity testing sample complexity bounds given in this and previous work. "SI" is sequentially interactive and "NI" is noninteractive.

p is α-far from uniform. For SimplePanTest, our statistic is

$$Z' = \sum_{i=1}^{k} \frac{(H_i - m/k)^2 - H_i}{m/k}$$

where m is the number of samples and H is a noisy histogram over $[k]$ where bin i counts the number of occurrences of element i in the stream. H contains Laplace noise added to each bin both before and after the stream. The first addition of noise ensures the privacy of the internal states during the stream, while the second addition of noise is for the privacy of the final output. Pseudocode for SimplePanTest appears below; values for m and T_U are determined in the proof of Lemma 24.

Inspired by similar statistics in non-private testing [1, 2, 24], Cai et al. [20] originally studied Z' for centrally private identity testing. However, they lower bounded its variance and argued that high variance makes it a suboptimal centrally private tester. We instead upper bound its variance and show that Z' yields a nontrivial pan-private uniformity tester.

Our argument is simple. First, we upper bound the variance of Z'. We then apply Chebyshev's inequality to upper bound Z' when p is uniform and lower bound Z' when p is α-far

Algorithm 8 Pan-private uniformity tester SimplePanTest

Require: privacy parameter ε, domain $[k]$
  Set sample size $m' \sim \mathsf{Poisson}(m)$ and threshold T_U
  Initialize private histogram $H \leftarrow \mathsf{Lap}\left(\frac{1}{\varepsilon}\right)^k \in \mathbb{R}^k$
  for stream elements $s_t = s_1, \ldots, s_{m'}$ **do**
    $H_{s_t} \leftarrow H_{s_t} + 1$
  $H \leftarrow H + \mathsf{Lap}\left(\frac{1}{\varepsilon}\right)^k \in \mathbb{R}^k$
  $Z' \leftarrow \sum_{i=1}^{k} \frac{(H_i - m/k)^2 - H_i}{m/k}$
  if $Z' > T_U$ **then**
    Output "non-uniform"
  else
    Output "uniform"

from uniform. These bounds drive our choice of the threshold T_U. We then compute the number of samples m required to separate these quantities on either side of T_U.

Finally, note that we actually draw $m' \sim \mathsf{Poisson}(m)$ samples, not m. This is the "Poissonization" trick mentioned in the preceding section.

Lemma 24. *For $m = \Omega\left(\frac{k^{3/4}}{\alpha\varepsilon} + \frac{\sqrt{k}}{\alpha^2}\right)$, SimplePanTest is an ε-pan-private uniformity tester on m samples.*

Proof. Privacy: Let t be a time in the stream, let i be a possible internal state for SimplePanTest, and let o be a possible output. Let $p_{\mathcal{I},s,t}$ be the probability density function for the internal state of SimplePanTest after the first t elements of stream s, and let $p_{\mathcal{O},s,t|i}$ be the probability density function for the output given stream s such that the internal state at time t was i. Finally, fix neighboring streams s and s'. Then to prove that SimplePanTest is ε-pan-private, it suffices to show that $\frac{p_{\mathcal{I},s,t}(i) \cdot p_{\mathcal{O},s,t|i}(o)}{p_{\mathcal{I},s',t}(i) \cdot p_{\mathcal{O},s',t|i}(o)} \leq e^{\varepsilon}$.

The final output of SimplePanTest is a deterministic function of its final internal state (after the second addition of Laplace noise). The final internal state is after m samples, so it is enough to choose arbitrary internal states i_1 and i_2 and show

$$\frac{p_{\mathcal{I},s,t}(i_1) \cdot p_{\mathcal{I},s,m,t|i_1}(i_2)}{p_{\mathcal{I},s',t}(i_1) \cdot p_{\mathcal{I},s',m,t|i_1}(i_2)} \leq e^{\varepsilon}. \tag{8.1}$$

We first recall a basic fact about differential privacy: if f is a real-valued function with sensitivity Δf, i.e. a function whose output changes by at most Δ between neighboring databases, then adding $\mathsf{Lap}\left(\frac{\Delta f}{\varepsilon}\right)$ noise to the output of f is ε-differentially private (see e.g. Theorem 3.4 in the survey of Dwork and Roth [37]). Here, each bin of H is a 1-sensitive function and each sample alters a single bin. Thus by the first application of $\mathsf{Lap}\left(\frac{1}{\varepsilon}\right)$ noise to each bin we get $\frac{p_{\mathcal{I},s,t}(i_1)}{p_{\mathcal{I},s',t}(i_1)} \leq e^{\varepsilon}$. Similarly, the second application of $\mathsf{Lap}\left(\frac{1}{\varepsilon}\right)$ noise to each bin implies $\frac{p_{\mathcal{I},s,m,t|i_1}(i_2)}{p_{\mathcal{I},s',m,t|i_1}(i_2)} \leq e^{\varepsilon}$. To get the overall claim, we split into two cases. If $s_{\leq t} = s'_{\leq t}$, then $\frac{p_{\mathcal{I},s,t}(i_1)}{p_{\mathcal{I},s',t}(i_1)} = 1$. If instead $s_{\leq t} \neq s'_{\leq t}$, then $s_{>t} = s'_{>t}$, so $\frac{p_{\mathcal{I},s,m,t|i_1}(i_2)}{p_{\mathcal{I},s',m,t|i_1}(i_2)} = 1$. Thus Equation 8.1 holds.

Sample complexity: To better analyze Z', we decompose it as the sum of a non-private χ^2-statistic Z and a noise term Y,

$$Z = \sum_{i=1}^{k} \frac{(N_i - m/k)^2 - N_i}{m/k} \text{ and } Y = \sum_{i=1}^{k} \frac{[Y_i + Y'_i]^2 + 2[Y_i + Y'_i](N_i - m/k) - [Y_i + Y'_i]}{m/k}.$$

where N_i is the true stream count of item i and $Y_i, Y'_i \sim \mathsf{Lap}\left(\frac{1}{\varepsilon}\right)$ are the first and second addition of Laplace noise. This lets us rewrite $Z' = Z + Y$. In the uniform case, we will give a high-probability upper bound for $Z + Y$, and in the non-uniform case we will give a high-probability lower bound. Fortunately, Acharya et al. [2] prove several results about Z. We summarize these results in Lemma 25.

Lemma 25 (Lemmas 2 and 3 from Acharya et al. [2])**.** *If $p = U_k$ and $m = \Omega\left(\frac{\sqrt{k}}{\alpha^2}\right)$, then $\mathbb{E}[Z] \leq \frac{\alpha^2 m}{500}$ and $Var[Z] \leq \frac{\alpha^4 m^2}{500000}$. If $||p - U_k||_{TV} \geq \alpha$, then $\mathbb{E}[Z] \geq \frac{\alpha^2 m}{5}$ and $Var[Z] \leq \frac{\mathbb{E}[Z]^2}{100}$.*

We split into cases depending on p. For each case, Lemma 25 will control Z, and our task will be to control Y.

Case 1: $p = U_k$. By Lemma 25, $\mathbb{E}[Z] \leq \frac{\alpha^2 m}{500}$ and $\mathrm{Var}[Z] \leq \frac{\alpha^4 m^2}{500000}$. By Chebyshev's inequality, $\mathbb{P}\left[Z > \left(\frac{1}{500} + \frac{c}{500\sqrt{2}}\right)\alpha^2 m\right] \leq \frac{1}{c^2}$.

Turning our attention to Y, define

$$A = \sum_{i=1}^{k} \frac{[Y_i + Y_i']^2}{m/k},\ B = \sum_{i=1}^{k} \frac{2[Y_i + Y_i'](N_i - m/k)}{m/k}, \text{ and } C = \sum_{i=1}^{k} \frac{Y_i + Y_i'}{m/k}.$$

Then we can rewrite $Y = A + B - C$. We control each of A, B, and C in turn. First, by the independence of all draws of noise, $\mathbb{E}[A] = \frac{k^2 \mathbb{E}\left[[Y_i + Y_i']^2\right]}{m} = \frac{2k^2 \text{Var}[Y_i]}{m} = \frac{4k^2}{\varepsilon^2 m}$ because $\text{Var}\left[\mathsf{Lap}\left(\frac{1}{\varepsilon}\right)\right] = \frac{2}{\varepsilon^2}$. Next,

$$\begin{aligned}
\text{Var}[A] =& \frac{k^3}{m^2} \text{Var}\left[Y_i^2 + 2Y_i Y_i' + Y_i'^2\right] \\
=& \frac{k^3}{m^2}\left(\mathbb{E}\left[(Y_i^2 + 2Y_i Y_i' + Y_i'^2)^2\right] - \mathbb{E}\left[Y_i^2 + 2Y_i Y_i' + Y_i'^2\right]^2\right) \\
=& \frac{k^3}{m^2}\left(\left[2\mathbb{E}\left[Y_i^4\right] + 6\mathbb{E}\left[Y_i^2\right]^2\right] - 4\mathbb{E}\left[Y_i^2\right]^2\right) \\
=& \frac{2k^3}{m^2}\left(\mathbb{E}\left[Y_i^4\right] + \mathbb{E}\left[Y_i^2\right]^2\right) \\
=& \frac{2k^3}{m^2}\left(\frac{12}{\varepsilon^4} + \frac{4}{\varepsilon^4}\right) \\
=& \frac{32k^3}{\varepsilon^4 m^2}
\end{aligned}$$

where we use $\mathbb{E}\left[Y_i^4\right] = \frac{\varepsilon}{2}\int_0^\infty x^4 e^{-\varepsilon x} dx = \frac{12}{\varepsilon^4}$ by repeated integration by parts. With Chebyshev's inequality, $\mathbb{P}\left[A > \frac{4k^2}{\varepsilon^2 m} + 6c\frac{k^{3/2}}{\varepsilon^2 m}\right] < \frac{1}{c^2}$.

To bound B, we use $\mathbb{E}[B] = 0$ and

$$\begin{aligned}
\mathrm{Var}[B] &= \frac{4k^2}{m^2} \cdot \mathrm{Var}\left[\sum_{i=1}^{k} [Y_i + Y_i'] \left(N_i - \frac{m}{k}\right)\right] \\
&= \frac{4k^2}{m^2} \cdot \mathbb{E}\left[\left(\sum_{i=1}^{k} [Y_i + Y_i'] \left[N_i - \frac{m}{k}\right]\right)^2\right] \\
&= \frac{4k^2}{m^2} \sum_{i_1, i_2 \in [k]} \mathbb{E}\left[(Y_{i_1} + Y_{i_1}')(Y_{i_2} + Y_{i_2}')\right] \cdot \mathbb{E}\left[\left(N_{i_1} - \frac{m}{k}\right)\left(N_{i_2} - \frac{m}{k}\right)\right] \\
&= \frac{4k^2}{m^2} \sum_{i=1}^{k} \mathbb{E}\left[(Y_i + Y_i')^2\right] \cdot \mathbb{E}\left[\left(N_i - \frac{m}{k}\right)^2\right] \\
&= \frac{16k^3}{\varepsilon^2 m^2}\left(\mathbb{E}\left[N_1^2\right] - \frac{2m\mathbb{E}[N_1]}{k} + \frac{m^2}{k^2}\right) \\
&= \frac{16k^3}{\varepsilon^2 m^2}\left(\mathrm{Var}[N_1] + \mathbb{E}[N_1]^2 - \frac{2m^2}{k^2} + \frac{m^2}{k^2}\right) \\
&= \frac{16k^2}{\varepsilon^2 m}
\end{aligned}$$

where the last two equalities use $N_i \sim \mathsf{Poisson}\left(\frac{m}{k}\right)$ and $\mathrm{Var}\left[\mathsf{Poisson}\left(\frac{m}{k}\right)\right] = \frac{m}{k}$. Again applying Chebyshev's inequality gives $\mathbb{P}\left[B > 4c\frac{k}{\varepsilon\sqrt{m}}\right] < \frac{1}{c^2}$.

Similarly, $\mathbb{E}[C] = 0$, and with $\mathrm{Var}[C] = \frac{k^3}{m^2} \cdot \mathrm{Var}[Y_i + Y_i'] = \frac{4k^3}{\varepsilon^2 m^2}$, $\mathbb{P}\left[C < -2c\frac{k^{3/2}}{\varepsilon m}\right] \leq \frac{1}{c^2}$.

Combining the above bounds on Z, A, B, and C, with probability at least $1 - \frac{4}{c^2}$,

$$Z' \leq \left(\tfrac{1}{500} + \tfrac{c}{500\sqrt{2}}\right)\alpha^2 m + \frac{4k^2}{\varepsilon^2 m} + 6c\frac{k^{3/2}}{\varepsilon^2 m} + 4c\frac{k}{\varepsilon\sqrt{m}} + 2c\frac{k^{3/2}}{\varepsilon m}.$$

Taking $c = 4\sqrt{2}$ and

$$T_U = \tfrac{1}{100}\alpha^2 m + 4\tfrac{k^2}{\varepsilon^2 m} + 24\sqrt{2}\tfrac{k^{3/2}}{\varepsilon^2 m} + 16\sqrt{2}\tfrac{k}{\varepsilon\sqrt{m}} + 8\sqrt{2}\tfrac{k^{3/2}}{\varepsilon m},$$

$\mathbb{P}[Z' \leq T_U] \geq 7/8$.

Case 2: $||p - U_k||_{TV} \geq \alpha$. By Lemma 25, $\mathbb{E}[Z] \geq \frac{\alpha^2 m}{5}$ and $\mathrm{Var}[Z] \leq \frac{\mathbb{E}[Z]^2}{100}$. Chebyshev's

inequality now gives

$$1 - \frac{1}{c^2} \leq \mathbb{P}\left[Z \geq \mathbb{E}[Z] - c\sqrt{\mathrm{Var}[Z]}\right] \leq \mathbb{P}\left[Z \geq \left(1 - \frac{c}{10}\right)\mathbb{E}[Z]\right] \leq \mathbb{P}\left[Z \geq \left(1 - \frac{c}{10}\right)\frac{\alpha^2 m}{5}\right]$$

where the last inequality requires $c \leq 10$. Returning to the decomposition of Y used in Case 1, A and C are unchanged and we can use our previous expressions for them (with appropriate sign changes for lower bounds). Our last task is to lower bound $B = \frac{2k}{m}\sum_{i=1}^{k}[Y_i + Y_i'](N_i - m/k)$. For any term i, Y_i and Y_i' are symmetric, so $Y_i + Y_i'](N_i - m/k)$ is symmetric as well, and in particular $\mathbb{P}[B \geq 0] \geq 1/2$.

Summing up, with probability at least $\frac{1}{2} - \frac{3}{c'^2}$,

$$Z' \geq \left(\frac{1}{5} - \frac{c'}{50}\right)\alpha^2 m + 4\frac{k^2}{\varepsilon^2 m} - 6c'\frac{k^{3/2}}{\varepsilon^2 m} - 2c'\frac{k^{3/2}}{\varepsilon m}.$$

Taking $c' = 2\sqrt{3}$ and $T_\alpha = \frac{\alpha^2 m}{10} + 4\frac{k^2}{\varepsilon^2 m} - 12\sqrt{3}\frac{k^{3/2}}{\varepsilon^2 m} - 4\sqrt{3}\frac{k^{3/2}}{\varepsilon m}$, $\mathbb{P}[Z' \geq T_\alpha] \geq \frac{1}{4}$.

For $T_\alpha > T_U$, it is enough that $T_\alpha - T_U > 0$.

$$T_\alpha - T_U = \frac{9}{100}\alpha^2 m - \left(12\sqrt{3} + 24\sqrt{2}\right)\frac{k^{3/2}}{\varepsilon^2 m} - 16\sqrt{2}\frac{k}{\varepsilon\sqrt{m}} - \left(4\sqrt{3} + 8\sqrt{2}\right)\frac{k^{3/2}}{\varepsilon m}.$$

Dropping constants, we need $\alpha^2 m = \Omega\left(\frac{k^{3/2}}{\varepsilon^2 m} + \frac{k}{\varepsilon\sqrt{m}} + \frac{k^{3/2}}{\varepsilon m}\right)$. We can drop the lower-order term $\frac{k^{3/2}}{\varepsilon m}$ and get $\alpha^2 m = \Omega\left(\frac{k^{3/2}}{\varepsilon^2 m} + \frac{k}{\varepsilon\sqrt{m}}\right)$, i.e. $m = \Omega\left(\frac{k^{3/4}}{\alpha\varepsilon} + \frac{k^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}}\right)$.

Putting it all together and recalling the assumption from Lemma 25, there exists constant c such that if $m > c\left(\frac{k^{3/4}}{\alpha\varepsilon} + \frac{k^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{\sqrt{k}}{\alpha^2}\right)$ then

$$\mathbb{P}[\text{output "uniform"} \mid ||p - U_k||_{TV} \geq \alpha] \leq 3/4 \text{ and } \mathbb{P}[\text{output "uniform"} \mid p = U_k] \geq 7/8.$$

Thus we get a constant 1/8 separation. By the amplification argument outlined after

Definition 25, SimplePanTest is a uniformity tester. Finally,

$$\frac{k^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}} = \left(\frac{k^{3/4}}{\alpha\varepsilon}\right)^{2/3} \cdot \left(\frac{\sqrt{k}}{\alpha^2}\right)^{1/3} \leq \frac{2}{3}\left(\frac{k^{3/4}}{\alpha\varepsilon}\right) + \frac{1}{3}\left(\frac{\sqrt{k}}{\alpha^2}\right)$$

by the AM-GM inequality, and our statement simplifies to $m = \Omega\left(\frac{k^{3/4}}{\alpha\varepsilon} + \frac{\sqrt{k}}{\alpha^2}\right)$. □

8.2.2. Optimal pan-private tester: PanTest

We now use SimplePanTest as a building block for a more complex tester PanTest. At a high level, PanTest splits the difference between local and central uniformity testers. We briefly recap these approaches for context.

Centrally private uniformity testers compute a fine-grained statistic depending on the empirical counts of each element $i \in [k]$. Specific methods include χ^2-style statistics [20], collision-counting [7], and empirical total variation distance from U_k [3], but all of these methods depend on accurate counts for each $i \in [k]$. Cai et al. [20] observed that adding Laplace noise to each such count before analyzing the statistic is centrally private. The cost is a large decrease in accuracy. This is unfortunate in our pan-private setting, as pan-privacy appears to force the same kind of per-count noise. Intuitively, a pan-private tester might benefit by maintaining a coarser statistic that is easier to maintain privately.

The best known locally private uniformity tester, due to Acharya et al. [4], uses an extreme version of this coarser strategy. Their approach randomly halves the domain $[k]$ into sets U and U^c and compares the number of samples falling into each. They prove that if p is sufficiently non-uniform to start, then $p(U)$ and $p(U^c)$ will also be non-uniform — albeit to a much smaller degree — with constant probability. This reduces uniformity testing to a simpler binary testing problem that, because of its much smaller domain, is more amenable to local privacy. However, it does so at the cost of a large reduction in testing distance, which makes the core distinguishing problem harder. Thus both locally private and pan-private versions of this approach have sample complexity $\Omega(k)$. Intuitively, because

pan-privacy does not force as much noise as local privacy, a pan-private algorithm might benefit by maintaining a finer statistic.

PanTest capitalizes on both of these ideas. First, it randomly partitions $[k]$ into n groups $G_1, \ldots, G_n$ of size $\Theta(k/n)$. It then runs SimplePanTest to test uniformity of the induced distribution over $[n]$, treating samples falling in each G_j as samples of $j \in [n]$.

PanTest thus intermediates between the central and local approaches. It chooses $n = n(\alpha, \varepsilon, k)$ according to $\frac{k^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}}$. When $\frac{k^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} < 2$, $n(\alpha, \varepsilon, k) = 2$ and PanTest uses the half-partition approach from local privacy. When $\frac{k^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} > k$, then $n(\alpha, \varepsilon, k) = k$ and PanTest uses the unpartitioned approach from central privacy. Finally, when $\frac{k^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} \in [2, k]$, $n(\alpha, \varepsilon, k) = \lfloor \frac{k^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} \rfloor$ and PanTest takes a middle ground. These choices enable PanTest to calibrate the noise contributed by privately maintaining different counts with the testing distance α Making this tradeoff work relies crucially on the $O\left(\frac{1}{\alpha}\right)$ dependence on distance achieved by SimplePanTest in its $k^{3/4}$ term. In contrast, the $\Omega\left(\frac{k}{\alpha^2}\right)$ dependence of the best known locally private uniformity tester yields no improvement with this approach. Pseudocode for PanTest appears below.

Algorithm 9 Improved pan-private uniformity tester PanTest

Require: privacy parameter ε, domain $[k]$
  if $\frac{k^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} < 2$ **then**
    $n \leftarrow 2$
  else if $\frac{k^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} > k$ **then**
    $n \leftarrow k$
  else
    $n \leftarrow \lfloor \frac{k^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} \rfloor$
  Randomly partition $[k]$ into n groups $G_1, \ldots, G_n$ of size $\Theta(k/n)$
  Run SimplePanTest$(\varepsilon, [n])$, treating each element $s_t \in G_j$ as $j \in [n]$

For this reduction to work, the aforementioned decrease in testing distance between $[k]$ and $[n]$ must not be too large. We show this in Lemma 26. This generalization is not new (see Theorem 3.2 from Acharya et al. [5]), but we include our proof here for completeness.

Lemma 26. *Let p be a distribution over $[k]$ such that $||p - U_k||_{TV} = \alpha$ and let $G_1, \ldots, G_n$*

be a uniformly random partition of $[k]$ *into* $n > 1$ *subsets of size* $\Theta(k/n)$*. Define induced distribution* p_n *over* $[n]$ *by* $p_n(j) = \sum_{i \in G_j} p(i)$ *for each* $j \in [n]$*. Then, with probability* $\geq \frac{1}{954}$ *over the selection of* $G_1, \ldots, G_n$*,*

$$||p_n - U_n||_{TV} = \Omega\left(\alpha\sqrt{\tfrac{n}{k}}\right).$$

Proof. It is equivalent to sample $G_1, \ldots, G_n$ as follows: randomly partition $[k]$ into $n/2$ same-size subsets $G'_1, \ldots, G'_{n/2}$ (for neatness, we assume n is even), and then randomly halve each of those to produce G_1 and G_2 (from G'_1), G_3 and G_4 (from G'_2), and so on. We use the following lemma from Acharya et al. [4] to connect the distances induced by $\{G'_a\}_{a=1}^{n/2}$ and $\{G_b\}_{b=1}^{n}$. Here, for a set S we let $p(S)$ denote the total probability mass of set S, $p(S) = \sum_{s \in S} p(s)$.

Lemma 27 (Corollary 15 in Acharya et al. [4])**.** *Let* p *be a distribution over* $[k]$ *with* $||p - U_k||_{TV} \geq \alpha$*, and let* U *be a random subset of* $[k]$ *of size* $k/2$*. Then* $\mathbb{P}_U\left[|p(U) - 1/2| \geq \frac{\alpha}{\sqrt{5k}}\right] > \frac{1}{477}$*.*

Slightly more generally, the proof of Lemma 27 shows that for any distribution p over $[k]$ and $S \subset [k]$, if $\frac{1}{2}\sum_{i \in S} |p(i) - \frac{1}{k}| \geq \alpha'$, and we choose a random subset $S' \subset S$ of size $\frac{|S|}{2}$, then $\mathbb{P}_{S'}\left[|p(S') - \frac{p(S)}{2}| \geq \frac{\alpha'}{\sqrt{5|S|}}\right] > \frac{1}{477}$.

Fix the choice of $G'_1, \ldots, G'_{n/2}$. For each $a \in [n/2]$, let $\alpha_a = \frac{1}{2}\sum_{i \in G'_a} |p(i) - \frac{1}{k}|$, the portion of $||p - U_k||_{TV}$ contributed by G'_a. Replacing α' with α_a and $|S|$ with $k/(n/2)$ above, for each $a \in [n/2]$,

$$\mathbb{P}\left[\left|p(G_{2a-1}) - \frac{p(G'_a)}{2}\right| \geq \alpha_a\sqrt{\tfrac{n}{10k}}\right] \geq \frac{1}{477}.$$

$p(G_{2a-1}) + p(G_{2a}) = p(G'_a)$, so

$$\mathbb{P}\left[|p(G_{2a-1}) - p(G_{2a})| \geq 2\alpha_a\sqrt{\tfrac{n}{10k}}\right] \geq \frac{1}{477}.$$

Then by triangle inequality

$$\mathbb{P}\left[\left|p(G_{2a-1})-\frac{1}{n}\right|+\left|p(G_{2a})-\frac{1}{n}\right|\geq 2\alpha_a\sqrt{\tfrac{n}{10k}}\right]\geq\frac{1}{477}$$

and in particular

$$\mathbb{E}\left[\left|p(G_{2a-1})-\frac{1}{n}\right|+\left|p(G_{2a})-\frac{1}{n}\right|\right]\geq\frac{2\alpha_a}{477}\sqrt{\frac{n}{10k}}.$$

For each $b\in[n]$ define $Y_b=\min\left(\left|p(G_b)-\frac{1}{n}\right|,\alpha_{\lceil b/2\rceil}\sqrt{\frac{n}{10k}}\right)$. Let $Y=\sum_{b=1}^{n}Y_b$. First, we can lower bound $\mathbb{E}[Y]$, over the choice of $G'_1,\ldots,G'_{n/2}$ and $G_1,\ldots,G_n$, as

$$\begin{aligned}\mathbb{E}[Y]&=\sum_{b=1}^{n}\mathbb{E}\left[\min\left(\left|p(G_b)-\frac{1}{n}\right|,\alpha_{\lceil b/2\rceil}\sqrt{\frac{n}{10k}}\right)\right]\\&\geq\sum_{b=1}^{n}\tfrac{\alpha_{\lceil b/2\rceil}}{477}\sqrt{\tfrac{n}{10k}}\\&=\tfrac{2\alpha}{477}\sqrt{\tfrac{n}{10k}}\end{aligned}\tag{8.2}$$

where the inequality uses the expectation lower bound above.

Second, by definition of Y_b, $\max(Y)\leq\sum_{b=1}^{n}\alpha_{\lceil b/2\rceil}\sqrt{\frac{n}{10k}}=2\alpha\sqrt{\frac{n}{10k}}$. Now assume for contradiction that $\mathbb{P}\left[Y\geq\frac{\alpha}{477}\sqrt{\frac{n}{10k}}\right]<\frac{1}{954}$. Then

$$\mathbb{E}[Y]<\tfrac{\alpha}{477}\sqrt{\tfrac{n}{10k}}+\tfrac{\max(Y)}{954}\leq\tfrac{2\alpha}{477}\sqrt{\tfrac{n}{10k}}.$$

Thus $\mathbb{E}[Y]<\frac{2\alpha}{477}\sqrt{\frac{n}{10k}}$, which contradicts Equation 8.2. It follows that our assumption is

false, and $\mathbb{P}\left[Y \geq \frac{\alpha}{477}\sqrt{\frac{n}{10k}}\right] \geq \frac{1}{954}$. The final claim follows from

$$\begin{aligned}\frac{Y}{2} &= \frac{1}{2}\sum_{b=1}^{n} \min\left(\left|p(G_b) - \frac{1}{n}\right|, \alpha_{\lceil b/2 \rceil}\sqrt{\frac{n}{10k}}\right) \\ &\leq \frac{1}{2}\sum_{b=1}^{n}\left|p(G_b) - \frac{1}{n}\right| \\ &= ||p_n - U_n||_{TV}.\end{aligned}$$

□

Due to the 1/954 success probability of Lemma 26, we have a smaller (but still constant) separation between output probabilities. We thus use the amplification argument discussed after Definition 25 to get Theorem 12. The guarantee combines Lemma 26 with Lemma 24, substituting n for k and $\alpha\sqrt{\frac{n}{k}}$ for α.

Theorem 12. *For $m = \Omega\left(\frac{k^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{\sqrt{k}}{\alpha^2} + \frac{\sqrt{k}}{\alpha\varepsilon}\right)$,* PanTest *is an ε-pan-private uniformity tester on m samples.*

Proof. Privacy: PanTest only interacts with the data through SimplePanTest, so PanTest inherits SimplePanTest's pan-privacy guarantee.

Sample complexity: Substituting n for k and $\alpha\sqrt{\frac{n}{k}}$ for α in Lemma 24, we require

$$m = \Omega\left(\frac{n^{1/4}\sqrt{k}}{\alpha\varepsilon} + \frac{k}{\alpha^2\sqrt{n}}\right). \tag{8.3}$$

We consider the three cases for $\frac{k^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}}$. Together, these cases exhaust the possible relationships among α, k, and ε, with a different highest-order term in each. This leads to the three terms in our bound.

First, if $\frac{k^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} \in [2, k]$, then $n = \lfloor \frac{k^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} \rfloor$. By Equation 8.3 it is enough for

$$m = \Omega\left(\frac{k^{1/6}\varepsilon^{1/3}\sqrt{k}}{\alpha^{1/3}\alpha\varepsilon} + \frac{k}{\alpha^2 \cdot \frac{k^{1/3}\varepsilon^{2/3}}{\alpha^{2/3}}}\right) = \Omega\left(\frac{k^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}}\right).$$

Next, if $\frac{k^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} > k$, then $n = k$, and Equation 8.3 necessitates $m = \Omega\left(\frac{k^{3/4}}{\alpha\varepsilon} + \frac{\sqrt{k}}{\alpha^2}\right)$. The condition $\frac{k^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} > k$ gives $\frac{\varepsilon^4}{\alpha^4} > k$, so $\frac{\varepsilon}{\alpha} > k^{1/4}$, and then multiplying both sides by $\frac{\sqrt{k}}{\alpha\varepsilon}$ gives $\frac{\sqrt{k}}{\alpha^2} > \frac{k^{3/4}}{\alpha\varepsilon}$.Thus it suffices for $m = \Omega\left(\frac{\sqrt{k}}{\alpha^2}\right)$.

Finally, if $\frac{k^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} < 2$, then $n = 2$ and by Equation 8.3 we require $m = \Omega\left(\frac{\sqrt{k}}{\alpha\varepsilon} + \frac{k}{\alpha^2}\right)$. $\frac{k^{2/3}\varepsilon^{4/3}}{\alpha^{4/3}} < 2$ implies $\varepsilon < \frac{2\alpha}{\sqrt{k}}$, so multiplying both sides by $\frac{k}{\alpha^2\varepsilon}$ yields $\frac{k}{\alpha^2} < \frac{2\sqrt{k}}{\alpha\varepsilon}$ and $\frac{\sqrt{k}}{\alpha\varepsilon} = \Omega\left(\frac{k}{\alpha^2}\right)$. Thus it suffices for $m = \Omega\left(\frac{\sqrt{k}}{\alpha\varepsilon}\right)$. □

8.3. Pan-Private Lower Bound

We now turn to lower bounds. Our first result gives a tight (in k) $\Omega\left(\frac{k^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}}\right)$ lower bound for ε-pan-private testing (Theorem 13). Our second result, in the next section, extends the previous $\Omega\left(\frac{k}{\alpha^2\varepsilon^2}\right)$ lower bound for noninteractive (ε, δ)-locally private uniformity testing [4] to the sequentially interactive case (Theorem 14).

Our lower bounds adapt the approach used by Diakonikolas, Gouleakis, Kane, and Rao [29] to prove testing lower bounds under memory restrictions and communication restrictions. The main difference in our lower bounds is that Diakonikolas et al. restrict their algorithm to use an internal state with b bits of memory. This memory restriction immediately implies that the internal state's entropy (and thus its mutual information with any other random variable) is also bounded by b. In our case, we must use our privacy restrictions to replace this result. Doing so constitutes the bulk of our arguments.

Finally, we note that these results add to lines of work conceptually connecting restricted memory to pan-privacy [36, 53] and connecting restricted communication to local privacy [4, 5, 31, 48, 51].

We start with the pan-private lower bound. While we state our result using $\alpha \leq 1/2$, the choice of $1/2$ is arbitrary: the same argument works for any α bounded below 1 by a constant. A short primer for the information theory used in our argument appears in the Appendix.

Theorem 13. *For $\varepsilon = O(1)$ and $\alpha \leq 1/2$, any ε-pan-private uniformity tester requires $m = \Omega\left(\frac{k^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{\sqrt{k}}{\alpha^2} + \frac{1}{\alpha\varepsilon}\right)$ samples.*

Proof. First, recall the centrally private lower bound ([3]):

$$m = \Omega\left(\frac{\sqrt{k}}{\alpha^2} + \frac{\sqrt{k}}{\alpha\sqrt{\varepsilon}} + \frac{k^{1/3}}{\alpha^{4/3}\varepsilon^{2/3}} + \frac{1}{\alpha\varepsilon}\right).$$

We will prove $m = \Omega\left(\frac{k^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}}\right)$ in the pan-private case. $\frac{k^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}}$ dominates the third term above and also dominates the second term for $\varepsilon = O(1)$, so this produces our final lower bound.

We start with the lower bound construction used by Diakonikolas et al. [29], which itself uses the Paninski lower bound construction [57]. Let X be a uniform random bit determining which of two distributions over $[2k]$ generates the samples. For both $X = 0$ and $X = 1$ we draw $Y_1, \ldots, Y_k \in \{\pm 1\}$ i.i.d. uniformly at random. If $X = 0$, $p = U_{2k}$. If instead $X = 1$, then we pair the bins as $\{1, 2\}, \{3, 4\}, \ldots, \{2k-1, 2k\}$ and define $p(2j-1) = \frac{1+Y_j\alpha}{2k}$ and $p(2j) = \frac{1-Y_j\alpha}{2k}$. Thus if $X = 0$ then p is uniform, and if $X = 1$ each pair i of bins is biased toward one of the bins according to Y_j. Equivalently, we can view each sample $S_t \sim p$ as a pair (J_t, V_t) where $J_t \in [k]$ determines the bin pair chosen and $V_t \in \{0, 1\}$ determines which of the bin pair is chosen. Thus $J_t \sim U_k$, and $V_t \sim \mathsf{Ber}\left(\frac{1}{2}\right)$ if $X = 0$ or $V_t \sim \mathsf{Ber}\left([1 + \alpha Y_{j_t}]/2\right)$ if $X = 1$, where $\mathsf{Ber}(\cdot)$ denotes the Bernoulli distribution.

To avoid confusion with the mutual information $I(\cdot)$, denote by M_t the random variable for the internal state of the algorithm after seeing sample S_t. Our goal is to upper bound the

mutual information between X and the internal state after m samples,

$$\begin{aligned} I(X; M_m) &= \sum_{t=1}^{m} I(X; M_t) - I(X; M_{t-1}) \\ &\leq \sum_{t=1}^{m} I(X; M_{t-1}, S_t) - I(X; M_{t-1}) \\ &= \sum_{t=1}^{m} I(X; S_t \mid M_{t-1}) \\ &= \sum_{t=1}^{m} I(X; V_t \mid M_{t-1}, J_t) \end{aligned} \tag{8.4}$$

where the last equality uses $S_t = (J_t, V_t)$ and the independence of X and J_t

We now have a narrower goal: we choose an arbitrary term in the sum in Equation (8.4) and upper bound it. For neatness, we use the convention that $H_2(p)$ is the entropy of a $\mathsf{Ber}(p)$ random variable. When subscripting we abuse notation and let $a \sim A$ denote a sample a from the distribution for random variable A. The following reproduces (and slightly expands) the first part of the argument given by [29]. It largely reduces to rewriting mutual information in terms of binary entropy and expanding conditional probabilities.

We start by rewriting the chosen term $I(X; V_t \mid M_{t-1}, J_t)$ as

$$\begin{aligned} &= \mathbb{E}_{m^* \sim M_{t-1}} \left[\mathbb{E}_{j \sim J_t} \left[H(V_t \mid M_{t-1} = m^*, J_t = j) \right] \right] \\ &\quad - \mathbb{E}_{m^* \sim M_{t-1}} \left[\mathbb{E}_{j \sim J_t} \left[\mathbb{E}_{x \sim X} \left[H(V_t \mid M_{t-1} = m^*, J_t = j, X = x) \right] \right] \right] \\ &= \mathbb{E}_{m^* \sim M_{t-1}} \left[\mathbb{E}_{j \sim J_t} \left[H_2(\mathbb{P}\left[V_t = 0 \mid M_{t-1} = m^*, J_t = j\right]) \right] \right] \\ &\quad - \mathbb{E}_{m^* \sim M_{t-1}} [\mathbb{E}_{j \sim J_t} [\mathbb{P}\left[X = 1 \mid M_{t-1} = m^*, J_t = j\right]] \\ &\quad \cdot H_2(\mathbb{P}\left[V_t = 0 \mid M_{t-1} = m^*, J_t = j, X = 1\right])] \\ &\quad - \mathbb{E}_{m^* \sim M_{t-1}} [\mathbb{E}_{j \sim J_t} [\mathbb{P}\left[X = 0 \mid M_{t-1} = m^*, J_t = j\right] \\ &\quad \cdot H_2(\mathbb{P}\left[V_t = 0 \mid M_{t-1} = m^*, J_t = j, X = 0\right])] \end{aligned}$$

where the second equality uses $H_2(p) = H_2(1-p)$. Let $\beta_{t-1}^{m^*,j} = \mathbb{P}\left[X = 1 \mid M_{t-1} = m^*, J_t = j\right]$.

Since J_t is a uniform draw from $[k]$ independent of M_{t-1}, we now continue the above chain of equalities as

$$
\begin{aligned}
&= \mathbb{E}_{m^* \sim M_{t-1}} \left[\frac{1}{k} \sum_{j=1}^{k} H_2 \left(\mathbb{P} \left[V_t = 0 \mid M_{t-1} = m^*, J_t = j \right] \right) \right] \\
&\quad - \mathbb{E}_{m^* \sim M_{t-1}} \left[\frac{1}{k} \sum_{j=1}^{k} \beta_{t-1}^{m^*,j} H_2 \left(\mathbb{P} \left[V_t = 0 \mid M_{t-1} = m^*, J_t = j, X = 1 \right] \right) \right] \\
&\quad - \mathbb{E}_{m^* \sim M_{t-1}} \left[\frac{1}{k} \sum_{j=1}^{k} (1 - \beta_{t-1}^{m^*,j}) H_2 \left(\mathbb{P} \left[V_t = 0 \mid M_{t-1} = m^*, J_t = j, X = 0 \right] \right) \right]. \qquad (8.5)
\end{aligned}
$$

Now recall that $V_t \sim \mathsf{Ber}\left(\frac{1}{2}\right)$ when $X = 0$ and $V_t \sim \mathsf{Ber}\left([1 + \alpha Y_{J_t}]/2\right)$ when $X = 1$. Then we can rewrite $\mathbb{P}\left[V_t = 0 \mid M_{t-1} = m^*, J_t = j\right]$ as

$$
\begin{aligned}
&= \beta_{t-1}^{m^*,j} \mathbb{P}\left[V_t = 0 \mid X = 1, M_{t-1} = m^*, J_t = j\right] \\
&\quad + (1 - \beta_{t-1}^{m^*,j}) \mathbb{P}\left[V_t = 0 \mid X = 0, M_{t-1} = m^*, J_t = j\right] \\
&= \beta_{t-1}^{m^*,j} \mathbb{P}\left[V_t = 0 \mid X = 1, M_{t-1} = m^*, J_t = j, Y_j = 1\right] \mathbb{P}\left[Y_j = 1 \mid M_{t-1} = m^*\right] \\
&\quad + \beta_{t-1}^{m^*,j} \mathbb{P}\left[V_t = 0 \mid X = 1, M_{t-1} = m^*, J_t = j, Y_j = -1\right] \mathbb{P}\left[Y_j = -1 \mid M_{t-1} = m^*\right] \\
&\quad + (1 - \beta_{t-1}^{m^*,j}) \mathbb{P}\left[V_t = 0 \mid X = 0\right] \\
&= \beta_{t-1}^{m^*,j} \left(\mathbb{P}\left[Y_j = 1 \mid M_{t-1} = m^*\right] \cdot \frac{1 - \alpha}{2} + \mathbb{P}\left[Y_j = -1 \mid M_{t-1} = m^*\right] \cdot \frac{1 + \alpha}{2} \right) + \frac{1 - \beta_{t-1}^{m^*,j}}{2} \\
&= \beta_{t-1}^{m^*,j} \mathbb{E}\left[\frac{1 - \alpha Y_j}{2} \mid M_{t-1} = m^* \right] + \frac{1 - \beta_{t-1}^{m^*,j}}{2} \\
&= \frac{\beta_{t-1}^{m^*,j} (1 - \alpha \mathbb{E}\left[Y_j \mid M_{t-1} = m^*\right]}{2} + \frac{1 - \beta_{t-1}^{m^*,j}}{2} = \frac{1 - \alpha \beta_{t-1}^{m^*,j} \mathbb{E}\left[Y_j \mid M_{t-1} = m^*\right]}{2}.
\end{aligned}
$$

where the first equality uses the independence of Y_j from X and J_t as well as the independence of V_t from M_{t-1} and J_t conditioned on $X = 0$, and the second equality uses the independence of V_t and M_{t-1} conditioned on $X, J_t = j$, and Y_j. Thus

$$
\mathbb{P}\left[V_t = 0 \mid M_{t-1} = m^*, J_t = j\right] = \frac{1 - \alpha \beta_{t-1}^{m^*,j} \mathbb{E}\left[Y_j \mid M_{t-1} = m^*\right]}{2}.
$$

Using the work above, we can also rewrite

$$\mathbb{P}\left[V_t = 0 \mid M_{t-1} = m^*, J_t = j, X = 1\right] = \frac{1 - \alpha\mathbb{E}\left[Y_j \mid M_{t-1} = m^*\right]}{2}$$

and

$$\mathbb{P}\left[V_t = 0 \mid M_{t-1} = m^*, J_t = j, X = 0\right] = \frac{1}{2}.$$

In the following chain of equalities, for space we let E be the event that $M_{t-1} = m^*$. Now we can return to Equation 8.5 and, since $H_2(\frac{1}{2}) = 1$, get

$$\begin{aligned}
(8.5) &= \mathbb{E}_{m^* \sim M_{t-1}}\left[\frac{1}{k}\sum_{j=1}^{k}\left(H_2\left(\frac{1 - \alpha\beta_{t-1}^{m^*,j}\mathbb{E}\left[Y_j \mid E\right]}{2}\right)\right.\right. \\
&\quad \left.\left. - \beta_{t-1}^{m^*,j} H_2\left(\frac{1 - \alpha\mathbb{E}\left[Y_j \mid E\right]}{2}\right) - (1 - \beta_{t-1}^{m^*,j})\right)\right] \\
&= \mathbb{E}_{m^* \sim M_{t-1}}\left[\frac{1}{k}\sum_{j=1}^{k}\left(\beta_{t-1}^{m^*,j}\left[1 - H_2\left(\frac{1 - \alpha\mathbb{E}\left[Y_j \mid E\right]}{2}\right)\right]\right.\right. \\
&\quad \left.\left. - \left[1 - H_2\left(\frac{1 - \alpha\beta_{t-1}^{m^*,j}\mathbb{E}\left[Y_j \mid E\right]}{2}\right)\right]\right)\right] \\
&\leq \mathbb{E}_{m^* \sim M_{t-1}}\left[\frac{1}{k}\sum_{j=1}^{k}\left[1 - H_2\left(\frac{1 - \alpha\mathbb{E}\left[Y_j \mid E\right]}{2}\right)\right]\right] \\
&= \mathbb{E}_{m^* \sim M_{t-1}}\left[\frac{1}{k}\sum_{j=1}^{k}\left[1 - H_2\left(\frac{1 + \alpha\mathbb{E}\left[Y_j \mid E\right]}{2}\right)\right]\right] \qquad (8.6)
\end{aligned}$$

where the inequality uses $H_2, \beta_{t-1}^{m^*,j} \leq 1$ and the equality uses $H_2\left(\frac{1}{2} - b\right) = H_2\left(\frac{1}{2} + b\right)$. We now control the terms with H_2. The Taylor series for $H_2(p)$ near $1/2$ is $H_2(p) = 1 - \frac{1}{2\ln(2)}\sum_{n=1}^{\infty}\frac{(1-2p)^{2n}}{n(2n-1)}$, so for $a < 1/2$

$$1 - H_2\left(\frac{1}{2} + a\right) < \sum_{n=1}^{\infty}\frac{(2a)^{2n}}{n^2} = 4a^2\sum_{n=1}^{\infty}\frac{(2a)^{2n-2}}{n^2} < 4a^2\sum_{n=1}^{\infty}\frac{1}{n^2} = \frac{2a^2\pi^2}{3}.$$

Substituting $1 - H_2\left(\frac{1}{2} + a\right) < \frac{2\pi^2 a^2}{3}$ into Inequality 8.6 and tracing back to Equation 8.4,

$$I(X; V_t \mid M_{t-1}, J_t) < \frac{\pi^2 \alpha^2}{6k} \mathbb{E}_{m^* \sim M_{t-1}} \left[\sum_{j=1}^{k} \mathbb{E}\left[Y_j \mid M_{t-1} = m^*\right]^2 \right] \tag{8.7}$$

We now depart from the argument of [29]. Our new goal is to upper bound

$$A = \mathbb{E}_{m^* \sim M_{t-1}} \left[\sum_{j=1}^{k} \mathbb{E}\left[Y_j \mid M_{t-1} = m^*\right]^2 \right]$$

$$=$$

$$Em^* \sim M_{t-1} \sum_{j=1}^{k} \left(2\mathbb{P}\left[Y_j = 1 \mid M_{t-1} = m^*\right] - 1\right)^2$$

$$= \mathbb{E}_{m^* \sim M_{t-1}} \left[\sum_{j=1}^{k} \left(\frac{\mathbb{P}\left[M_{t-1} = m^* \mid Y_j = 1\right]}{\mathbb{P}\left[M_{t-1} = m^*\right]} - 1 \right)^2 \right]$$

by Bayes' rule and $\mathbb{P}\left[Y_j = 1\right] = 1/2$. To upper bound this sum, we choose an arbitrary j and show that $\frac{\mathbb{P}[M_{t-1}=m^*|Y_j=1]}{\mathbb{P}[M_{t-1}=m^*]}$ is close to 1. We pause to recap what we've accomplished and what remains. Note that proving $\frac{\mathbb{P}[M_{t-1}=m^*|Y_j=1]}{\mathbb{P}[M_{t-1}=m^*]} \approx 1$ "looks like" a privacy statement: we are claiming that the state distribution M_{t-1} looks similar when its input distribution is slightly different. However, there is still a gap between a difference in input distribution and a difference in input. We close this gap in the following lemma, which relies on pan-privacy.

Lemma 28. $\left| \frac{\mathbb{P}[M_{t-1}=m^*|Y_j=1]}{\mathbb{P}[M_{t-1}=m^*]} - 1 \right| = O\left(\frac{\alpha\varepsilon t}{k}\right).$

Proof. We will prove this claim by showing that both the numerator and denominator of $\frac{\mathbb{P}[M_{t-1}=m^*|Y_j=1]}{\mathbb{P}[M_{t-1}=m^*]}$ fall into a bounded range. This implies that the whole fraction is near 1.

First consider the case $X = 0$. Then the Y_j are irrelevant, so $\left| \frac{\mathbb{P}[M_{t-1}=m^*|Y_j=1]}{\mathbb{P}[M_{t-1}=m^*]} - 1 \right| = 0$.

Next, consider the case $X = 1$. It will be useful to consider an equivalent method of sampling the stream S. At each time step t, we first sample a bin pair $J_t \sim_U [k]$ uniformly at random from the k bin pairs. Having sampled bin pair j, with probability $1 - \alpha$ we take

a uniform random draw from $\{2j-1, 2j\}$. With the remaining probability α, if $Y_j = 1$ then we sample $2j-1$, and if $Y_j = -1$ then we sample $2j$. Note that this method is equivalent because if $Y_j = 1$ then $\mathbb{P}[\text{sample } 2j-1] = \frac{1}{k} \cdot \frac{1-\alpha}{2} + \frac{\alpha}{k} = \frac{1+\alpha}{2k}$ and $\mathbb{P}[\text{sample } 2j] = \frac{1-\alpha}{2k}$, with these equalities swapped for $Y_j = -1$. With this view of sampling, let $E^{\alpha}_{j,t} = 1$ if $J_t = j$ and we sample from the α mixture component and $E^{\alpha}_{j,t} = 0$ otherwise. Finally, let $N^{\alpha}_{j,t} = \sum_{t'=1}^{t} E^{\alpha}_{j,t'}$, the number of samples from the α mixture component of bin pair j through the first t stream elements.

We pause to justify bothering with this alternate view. We use it because the original ratio $\frac{\mathbb{P}[M_{t-1}=m^*|Y_j=1]}{\mathbb{P}[M_{t-1}=m^*]}$ is comparing the views of M_{t-1} depending on Y_j. It is not obvious how to directly use pan-privacy to reason about this comparison because Y_j is a property of the distribution generating the samples (stream elements) rather than the samples themselves. In contrast, pan-privacy is a guarantee formulated in terms of the samples. By defining the $E^{\alpha}_{j,t}$ and $N^{\alpha}_{j,t}$ above we better connect Y_j to the actual samples received. The alternate view therefore makes using pan-privacy easier.

We first analyze the denominator of $\frac{\mathbb{P}[M_{t-1}=m^*|Y_j=1]}{\mathbb{P}[M_{t-1}=m^*]}$. We can rewrite it as

$$\mathbb{P}[M_{t-1} = m^*] = \sum_{q=0}^{t-1} \mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = q\right] \cdot \mathbb{P}\left[N^{\alpha}_{j,t-1} = q\right]. \tag{8.8}$$

Fix some $q \in \{0, 1, \ldots, t-1\}$. Let $S_{j,\le t^*}$ be the random variable for the bin pairs and component of j sampled through time t^*, i.e. $S_{j,\le t^*} = \{(J_t, E^{\alpha}_{j,t})\}_{t=1}^{t^*}$. Note that this means the tuple $(j', 1)$ is possible only when $j' = j$. Define $\mathcal{S}^{\alpha}_{j,q,t}$ to be the set of realizations of $S_{j,\le t}$ with exactly q samples from the α component of bin pair j. Then $\mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = q\right]$

$$\begin{aligned}
&= \sum_{s \in \mathcal{S}^{\alpha}_{j,q,t-1}} \mathbb{P}[M_{t-1} = m^* \mid S_{j,\le t-1} = s] \cdot \mathbb{P}\left[S_{j,\le t-1} = s \mid N^{\alpha}_{j,t-1} = q\right] \\
&= \sum_{s \in \mathcal{S}^{\alpha}_{j,q,t-1}} \frac{1}{\binom{t-1}{q} k^{t-1-q}} \cdot \mathbb{P}[M_{t-1} = m^* \mid S_{j,\le t-1} = s] \qquad (8.9)
\end{aligned}$$

where the second equality uses the fact that, conditioned on $N^{\alpha}_{j,t-1} = q$, there are $\binom{t-1}{q} k^{t-1-q}$

equiprobable realizations of $S_{j,\leq t-1}$. Note that we are now reasoning directly about the stream's effect on the state M_{t-1}. This is much closer to the application of pan-privacy that we set out to achieve.

Consider a length-$(t-1)$ realization $s \in \mathcal{S}^{\alpha}_{j,q,t-1}$. Recall that each index of s takes one of $j+1$ possible values: $(1,0),(2,0),\ldots,(k,0)$, or $(j,1)$. Let $s' \in S^{\alpha}_{j,0,t-1}$ be a realization such that the Hamming distance $d_H(s,s') = q$, i.e. s and s' differ in exactly q indices. Then because M_{t-1} is an ε-differentially private function of the stream, by group privacy (see e.g. Theorem 2.2 [37])

$$\mathbb{P}\left[M_{t-1} = m^* \mid S_{j,\leq t-1} = s\right] \leq e^{q\varepsilon}\mathbb{P}\left[M_{t-1} = m^* \mid S_{j,\leq t-1} = s'\right].$$

Moreover, there are exactly k^q such s' for each such s. Denote this set of s' by $T_{s,q}$. We can now continue

$$\begin{aligned}
(8.9) &= \sum_{s\in\mathcal{S}^{\alpha}_{j,q,t-1}} \frac{1}{k^q} \sum_{s'\in T_{s,q}} \frac{1}{\binom{t-1}{q}k^{t-1-q}} \cdot \mathbb{P}\left[M_{t-1} = m^* \mid S_{j,\leq t-1} = s\right] \\
&\leq \sum_{s\in\mathcal{S}^{\alpha}_{j,q,t-1}} \sum_{s'\in T_{s,q}} \frac{e^{q\varepsilon}}{\binom{t-1}{q}k^{t-1}} \cdot \mathbb{P}\left[M_{t-1} = m^* \mid S_{j,\leq t-1} = s'\right] \\
&= \sum_{s'\in S^{\alpha}_{j,0,t-1}} \frac{e^{q\varepsilon}}{k^{t-1}} \cdot \mathbb{P}\left[M_{t-1} = m^* \mid S_{j,\leq t-1} = s'\right] \\
&= \sum_{s'\in S^{\alpha}_{j,0,t-1}} e^{q\varepsilon} \cdot \mathbb{P}\left[M_{t-1} = m^* \mid S_{j,\leq t-1} = s'\right] \cdot \mathbb{P}\left[S_{j,\leq t-1} = s' \mid N^{\alpha}_{j,t-1} = 0\right] \\
&= e^{q\varepsilon}\mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = 0\right]
\end{aligned}$$

where the first inequality uses the above group privacy guarantee; the second equality uses the fact that, for a given $s' \in T_{s,q}$, there are exactly $\binom{t-1}{q}$ length-$(t-1)$ realizations s with q samples from the α mixture component from bin pair j and $d_H(s,s') = q$; and the last equality uses the fact that M_{t-1} and $N^{\alpha}_{j,t-1}$ are independent conditioned on $S_{j,\leq t-1}$. Note that this expression depending only on the conditioning for $N^{\alpha}_{j,t-1} = 0$ is useful because it will give us a "fixed point" to relate the numerator and denominator analyses. By

expressing both quantities with respect to this condition, we can better compare them (and in particular, obtain a cancellation in the final ratio).

Returning to Equation 8.8

$$\mathbb{P}\left[M_{t-1} = m^*\right] = \sum_{q=0}^{t-1} \mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = q\right] \cdot \mathbb{P}\left[N^{\alpha}_{j,t-1} = q\right]$$

we get

$$\begin{aligned}
\mathbb{P}\left[M_{t-1} = m^*\right] &\leq \sum_{q=0}^{t-1} e^{q\varepsilon} \mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = 0\right] \cdot \mathbb{P}\left[N^{\alpha}_{j,t-1} = q\right] \\
&= \mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = 0\right] \cdot \sum_{q=0}^{t-1} e^{q\varepsilon} \mathbb{P}\left[N^{\alpha}_{j,t-1} = q\right] \\
&= \mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = 0\right] \cdot \mathbb{E}\left[e^{\varepsilon N^{\alpha}_{j,t-1}}\right]. \qquad (8.10)
\end{aligned}$$

To analyze this last quantity, recall that we defined random variable $E^{\alpha}_{j,t}$ as the indicator variable for drawing stream element t from the α mixture component of bin pair j. Then

$$\mathbb{E}\left[e^{\varepsilon N_{j,t-1}}\right] = \mathbb{E}\left[e^{\sum_{i=1}^{t-1} \varepsilon E^{\alpha}_{j,i}}\right] = \prod_{i=1}^{t-1} \mathbb{E}\left[e^{\varepsilon E^{\alpha}_{j,i}}\right] = \left[\left(1 - \frac{\alpha}{k}\right) e^0 + \frac{\alpha}{k} e^{\varepsilon}\right]^{t-1} = \left[1 + \frac{\alpha(e^{\varepsilon} - 1)}{k}\right]^{t-1}.$$

Since $1 + x \leq e^x$, $[1 + \frac{\alpha(e^{\varepsilon}-1)}{k}]^{t-1} \leq e^{\frac{\alpha(e^{\varepsilon}-1)(t-1)}{k}}$. We analyze this quantity in cases.

In the first case, $\frac{\alpha(e^{\varepsilon}-1)(t-1)}{k} \geq 1$. Then $t > \frac{k}{\alpha(e^{\varepsilon}-1)}$, and since $\varepsilon = O(1)$ there exists constant C such that $t > C\frac{k}{\alpha\varepsilon}$. $t \leq m$ so $m > C\frac{k}{\alpha\varepsilon}$. However, by the non-private uniformity testing lower bound, $I(X; M_m) = \Omega(1)$ requires $m = \Omega\left(\frac{\sqrt{k}}{\alpha^2}\right)$. This means we have some constant C' such that

$$m > C' \left(\frac{\sqrt{k}}{\alpha^2}\right)^{1/3} \left(\frac{k}{\alpha\varepsilon}\right)^{2/3} = \Omega\left(\frac{k^{5/6}}{\alpha^{4/3}\varepsilon^{2/3}}\right) \qquad (8.11)$$

which suffices for our overall lower bound.

All that remains is the second case, $\frac{\alpha(e^{\varepsilon}-1)(t-1)}{k} < 1$. Then since $e^x \leq 1 + 2x$ for $x \in [0, 1]$,

$e^{\frac{\alpha(e^{\varepsilon}-1)(t-1)}{k}} \leq 1 + 2\frac{\alpha(e^{\varepsilon}-1)(t-1)}{k}$. Again using $\varepsilon = O(1)$, there exists constant C_1 such that $\left[1 + \frac{\alpha(e^{\varepsilon}-1)}{k}\right]^{t-1} \leq e^{\frac{\alpha(e^{\varepsilon}-1)(t-1)}{k}} \leq 1 + C_1\frac{\alpha\varepsilon(t-1)}{k}$. Thus we return to Equation 8.10 and get

$$\mathbb{P}\left[M_{t-1} = m^*\right] \leq \mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = 0\right] \cdot \left(1 + C_1\frac{\alpha\varepsilon(t-1)}{k}\right).$$

If we repeat this process using the other direction of group privacy, we get

$$\mathbb{P}\left[M_{t-1} = m^*\right] \geq \mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = 0\right] \left[1 + \frac{\alpha(e^{-\varepsilon}-1)}{k}\right]^{t-1}.$$

$k \geq 2$, $\varepsilon > 0$, and $\alpha \leq 1$, so $\frac{\alpha(e^{-\varepsilon}-1)}{k} \in (-1, 0)$. Thus $\left[1 + \frac{\alpha(e^{-\varepsilon}-1)}{k}\right]^{t-1} \geq 1 + \frac{\alpha(e^{-\varepsilon}-1)(t-1)}{k}$. By $\varepsilon = O(1)$, we get a constant C_2 such that $\left[1 + \frac{\alpha(e^{-\varepsilon}-1)}{k}\right]^{t-1} \geq 1 - C_2\frac{\alpha\varepsilon(t-1)}{k}$. Tracing back,

$$\mathbb{P}\left[M_{t-1} = m^*\right] \geq \mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = 0\right] \cdot \left(1 - C_2\frac{\alpha\varepsilon(t-1)}{k}\right).$$

Returning to the beginning of our proof, we can repeat the argument for the numerator of $\frac{\mathbb{P}[M_{t-1}=m^* | Y_j=1]}{\mathbb{P}[M_{t-1}=m^*]}$:

$$\begin{aligned}
\mathbb{P}\left[M_{t-1} = m^* \mid Y_j = 1\right] &= \sum_{q=0}^{t-1} \mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = q, Y_j = 1\right] \cdot \mathbb{P}\left[N^{\alpha}_{j,t-1} = q \mid Y_j = 1\right] \\
&= \sum_{q=0}^{t-1} \mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = q, Y_j = 1\right] \cdot \mathbb{P}\left[N^{\alpha}_{j,t-1} = q\right]
\end{aligned}$$

since $N^{\alpha}_{j,t}$ and Y_j are independent. Fixing a q, we rewrite $\mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = q, Y_j = 1\right]$

$$\begin{aligned}
&= \sum_{s \in \mathcal{S}^{\alpha}_{j,q,t-1}} \mathbb{P}\left[M_{t-1} = m^* \mid S_{j,\leq t-1} = s, Y_j = 1\right] \cdot \mathbb{P}\left[S_{j,\leq t-1} = s \mid N^{\alpha}_{j,t-1} = q\right] \\
&= \sum_{s \in \mathcal{S}^{\alpha}_{j,q,t-1}} \frac{1}{\binom{t-1}{q} k^{t-1-q}} \cdot \mathbb{P}\left[M_{t-1} = m^* \mid S_{j,\leq t-1} = s, Y_j = 1\right] \qquad (8.12)
\end{aligned}$$

where the first equality uses the independence of M_{t-1} and $N^{\alpha}_{j,t-1}$ conditioned on $S_{j,t-1}$ as well as the independence of $S_{j,\leq t-1}$ and Y_j, and the second equality uses the same counting

argument as in the denominator case. Next, ε-pan-privacy gives

$$\mathbb{P}\left[M_{t-1} = m^* \mid S_{j,\leq t-1} = s, Y_j = 1\right] \leq e^{q\varepsilon}\mathbb{P}\left[M_{t-1} = m^* \mid S_{j,\leq t-1} = s', Y_j = 1\right]$$

and so

$$\begin{aligned}
(8.12) &= \sum_{s\in\mathcal{S}^{\alpha}_{j,q,t-1}} \frac{1}{k^q} \sum_{s'\in T_{s,q}} \frac{1}{\binom{t-1}{q}k^{t-1-q}} \cdot \mathbb{P}\left[M_{t-1} = m^* \mid S_{j,t-1} = s, Y_j = 1\right] \\
&\leq \sum_{s\in\mathcal{S}^{\alpha}_{j,q,t-1}} \sum_{s'\in T_{s,q}} \frac{e^{q\varepsilon}}{\binom{t-1}{q}k^{t-1}} \mathbb{P}\left[M_{t-1} = m^* \mid S_{j,t-1} = s', Y_j = 1\right] \\
&= \sum_{s'\in\mathcal{S}^{\alpha}_{j,q,t-1}} \frac{e^{q\varepsilon}}{k^{t-1}} \cdot \mathbb{P}\left[M_{t-1} = m^* \mid S_{j,t,-1} = s', Y_j = 1\right] \\
&= \sum_{s'\in\mathcal{S}^{\alpha}_{j,0,t-1}} \Bigg(e^{q\varepsilon} \cdot \mathbb{P}\left[M_{t-1} = m^* \mid S_{j,\leq t-1} = s', Y_j = 1\right] \\
&\quad \cdot \mathbb{P}\left[S_{j,\leq t-1} = s' \mid N^{\alpha}_{j,t-1} = 0, Y_j = 1\right] \Bigg) \\
&= e^{q\varepsilon}\mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = 0\right]
\end{aligned}$$

where the last equality uses the independence of $S_{j,\leq t-1}$ and Y_j conditioned on $N^{\alpha}_{j,t-1} = 0$ and the independence of M_{t-1} and Y_j and $N^{\alpha}_{j,t-1}$ conditioned on $S_{j,\leq t-1}$. In turn we get

$$\mathbb{P}\left[M_{t-1} = m^* \mid Y_j = 1\right] \leq \mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = 0\right] \sum_{q=0}^{t-1} e^{q\varepsilon}\mathbb{P}\left[N^{\alpha}_{j,t-1} = q\right]$$

which is the same quantity as in Equation 8.10. The same analysis thus gives

$$\mathbb{P}\left[M_{t-1} = m^* \mid Y_j = 1\right] \leq \mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = 0\right] \cdot \left(1 + C_1\frac{\alpha\varepsilon(t-1)}{k}\right)$$

as in the denominator case, and

$$\mathbb{P}\left[M_{t-1} = m^* \mid Y_j = 1\right] \geq \mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = 0\right] \cdot \left(1 - C_2\frac{\alpha\varepsilon(t-1)}{k}\right).$$

Summing up, shorthanding $A = \mathbb{P}\left[M_{t-1} = m^* \mid N^{\alpha}_{j,t-1} = 0\right]$, both $\mathbb{P}\left[M_{t-1} = m^*\right]$ and $\mathbb{P}\left[M_{t-1} = m^* \mid Y_j = 1\right]$ lie in the interval

$$\left[A \cdot \left(1 - C_2 \frac{\alpha\varepsilon(t-1)}{k}\right), A \cdot \left(1 + C_1 \frac{\alpha\varepsilon(t-1)}{k}\right)\right].$$

Thus

$$\begin{aligned}
\frac{\mathbb{P}\left[M_{t-1} = m^* \mid Y_j = 1\right]}{\mathbb{P}\left[M_{t-1} = m^*\right]} &\leq \frac{1 + C_1 \frac{\alpha\varepsilon(t-1)}{k}}{1 - C_2 \frac{\alpha\varepsilon(t-1)}{k}} \\
&= 1 + \frac{C_1 + C_2}{1 - C_2 \frac{\alpha\varepsilon(t-1)}{k}} \cdot \frac{\alpha\varepsilon(t-1)}{k} \\
&= 1 + O\left(\frac{\alpha\varepsilon t}{k}\right)
\end{aligned}$$

where the last equality uses $\frac{\alpha\varepsilon(t-1)}{k} < \frac{1}{2C_2}$ (otherwise, we get $m = \Omega\left(\frac{k}{\alpha\varepsilon}\right)$ and can use the argument given in Equation 8.11). Similarly,

$$\begin{aligned}
\frac{\mathbb{P}\left[M_{t-1} = m^* \mid Y_j = 1\right]}{\mathbb{P}\left[M_{t-1} = m^*\right]} &\geq \frac{1 - C_2 \frac{\alpha\varepsilon(t-1)}{k}}{1 + C_1 \frac{\alpha\varepsilon(t-1)}{k}} \\
&= 1 - \frac{C_1 + C_2}{1 + C_1 \frac{\alpha\varepsilon(t-1)}{k}} \cdot \frac{\alpha\varepsilon(t-1)}{k} \\
&= 1 - O\left(\frac{\alpha\varepsilon t}{k}\right)
\end{aligned}$$

and the claim follows. □

Lemma 28 gives $A \leq \frac{\alpha^2\varepsilon^2 t^2}{k}$, so $\frac{\alpha^2 A}{k} \leq \frac{\alpha^4\varepsilon^2 t^2}{k^2}$. Returning to Equation 8.7 and using $t \leq m$, $I(X; V_t \mid M_{t-1}, J_t) = O\left(\frac{\alpha^4\varepsilon^2 m^2}{k^2}\right)$. Then we trace back to Equation 8.4 and get $I(X; M_m) = O\left(\frac{\alpha^4\varepsilon^2 m^3}{k^2}\right)$. Finally, a uniformity tester requires $I(X; M_m) = \Omega(1)$, so $m = \Omega\left(\frac{k^{2/3}}{\alpha^{4/3}\varepsilon^{2/3}}\right)$.

□

8.4. Locally Private Lower Bound

We now move to the the locally private lower bound. We state our result for pure sequentially interactive local privacy, but this is without loss of generality by the approximate-to-pure result summarized in Lemma 7.

Theorem 14. *For $\varepsilon = O(1)$, any sequentially interactive ε-locally private uniformity tester requires $m = \Omega\left(\frac{k}{\alpha^2\varepsilon^2}\right)$ samples.*

Proof. Let M_t be the random variable for the message sent by user t with sample S_t, and let $M_{1:t}$ be the concatenation of messages sent through time t. We start by distinguishing our approach for this lower bound from its pan-private analogue. Recall that in the pan-private lower bound we expressed the mutual information between the distribution parameter X and the internal state after m samples M_m as $I(X; M_m) = \sum_{t=1}^{m} I(X; S_t \mid M_{t-1})$. Here, we want to control the mutual information between X and the transcript through m samples, $I(X; M_{1:m})$. A key difference in the local setting is that the algorithm does not see any sample S_t. Instead, the algorithm sees a randomizer output based on S_t. We should therefore expect some information loss between the sample and its randomizer output. We formalize this using existing local privacy work (Lemma 29) and get $I(X; M_{1:m}) < \sum_{t=1}^{m} O(\varepsilon^2) \cdot I(X; S_t \mid M_{1:t-1})$. This partially explains the locally private lower bound's different dependence on ε.

More formally, by the chain rule for mutual information, $I(X; M_{1:m}) = \sum_{t=1}^{m} I(X; M_t \mid M_{1:t-1})$. Choose one term $I(X; M_t \mid M_{1:t-1})$ and fix a value m for $M_{1:t-1}$. We can rewrite $I(X; M_t \mid M_{1:t-1} = m)$ as

$$
\begin{aligned}
&= \mathbb{E}_{X|M_{1:t-1}=m}\left[D_{KL}\left(M_t \mid X, M_{1:t-1} = m || M_t \mid M_{1:t-1} = m\right)\right] \\
&= \mathbb{P}\left[X = 0 \mid M_{1:t-1} = m\right] D_{KL}\left(M_t \mid X = 0, M_{1:t-1} = m || M_t \mid M_{1:t-1} = m\right) \\
&\quad + \mathbb{P}\left[X = 1 \mid M_{1:t-1} = m\right] D_{KL}\left(M_t \mid X = 1, M_{1:t-1} = m || M_t \mid M_{1:t-1} = m\right). \qquad (8.13)
\end{aligned}
$$

$M_{1:m}$ is generated by a sequentially interactive ε-locally private protocol. We can therefore use the following result from Duchi et al. [32]. We originally used this result as Lemma 9, but we restate it here for ease of exposition.

Lemma 29. *Let Q be an ε-randomizer and let P_0 and P_1 be distributions on $\mathcal{X}$. Let $x_0 \sim P_0$ and $x_1 \sim P_1$. Then*

$$D_{KL}\left(Q(x_0)||Q(x_1)\right) + D_{KL}\left(Q(x_1)||Q(x_0)\right) \leq \min(4, e^{2\varepsilon}) \cdot (e^{\varepsilon} - 1)^2 ||P_0 - P_1||_{TV}^2.$$

Here, we let P_1 be the distribution for $S_t \mid M_{1:t-1} = m$, P_2 for $S_t \mid X = 0, M_{1:t-1} = m$, and P_3 for $S_t \mid X = 1, M_{1:t-1} = m$. Q_1 is then the distribution for $M_t \mid M_{1:t-1} = m$, Q_2 for $M_t \mid X = 0, M_{1:t-1} = m$, and Q_3 for $M_t \mid X = 1, M_{1:t-1} = m$. Lemma 29 then gives

$$\begin{aligned}
(8.13) &\leq 4(e^{\varepsilon} - 1)^2 [\mathbb{P}\left[X = 0 \mid M_{1:t-1} = m\right] ||P_1 - P_2||_{TV}^2 \\
&\quad + \mathbb{P}\left[X = 1 \mid M_{1:t-1} = m\right] ||P_1 - P_3||_{TV}^2] \\
&\leq 2(e^{\varepsilon} - 1)^2 [\mathbb{P}\left[X = 0 \mid M_{1:t-1} = m\right] D_{KL}\left(P_1||P_2\right) \\
&\quad + \mathbb{P}\left[X = 1 \mid M_{1:t-1} = m\right] D_{KL}\left(P_1||P_3\right)] \\
&= 2(e^{\varepsilon} - 1)^2 I(X; S_t \mid M_{1:t-1} = m)
\end{aligned}$$

where the second inequality uses Pinsker's inequality. Now we can quantify the loss in information between the sample S_t and the private message M_t:

$$\begin{aligned}
I(X; M_{1:m}) &= \sum_{t=1}^{m} I(X; M_t \mid M_{1:t-1}) \\
&\leq \sum_{t=1}^{m} 2(e^{\varepsilon} - 1)^2 I(X; S_t \mid M_{1:t-1}) \\
&\leq \sum_{t=1}^{m} 2(e^{\varepsilon} - 1)^2 I(X; V_t \mid M_{1:t-1}, J_t) \qquad (8.14)
\end{aligned}$$

and, by the same reasoning as in the proof of Theorem 1,

$$I(X; V_t \mid M_{1:t-1}, J_t) = O\left(\frac{\alpha^2}{k}\mathbb{E}_{M_{1:t-1}}\left[\sum_{j=1}^{k}\mathbb{E}\left[Y_j \mid M_{1:t-1}\right]^2\right]\right). \tag{8.15}$$

and in turn rewrite the RHS inside $O(\cdot)$ as

$$\frac{\alpha^2}{k}\sum_{i=1}^{t-1}\sum_{j=1}^{k}\left(\mathbb{E}_{M_{1:i}}\left[\mathbb{E}\left[Y_j \mid M_{1:i}\right]^2\right] - \mathbb{E}_{M_{1:i-1}}\left[\mathbb{E}\left[Y_j \mid M_{1:i-1}\right]^2\right]\right). \tag{8.16}$$

We now fix some i and want to upper bound

$$\sum_{j=1}^{k}\left(\mathbb{E}_{M_{1:i}}\left[\mathbb{E}\left[Y_j \mid M_{1:i}\right]^2\right] - \mathbb{E}_{M_{1:i-1}}\left[\mathbb{E}\left[Y_j \mid M_{1:i-1}\right]^2\right]\right).$$

Choose one term j and define $\gamma_j = \mathbb{P}\left[Y_j = 1 \mid M_{1:i}\right]$. Then we get

$$\begin{aligned}
\mathbb{E}_{M_{1:i}}\left[\mathbb{E}\left[Y_j \mid M_{1:i}\right]^2\right] &= \mathbb{E}_{M_{1:i}}\left[(\gamma_j - (1-\gamma_j))^2\right] \\
&= \mathbb{E}_{M_{1:i}}\left[4\gamma_j^2 - 4\gamma_j + 1\right] \\
&= 4\mathbb{E}_{M_{1:i}}\left[\gamma_j^2\right] - 4\mathbb{E}_{M_{1:i}}\left[\gamma_j\right] + 1 \\
&= 4\mathbb{E}_{M_{1:i}}\left[\gamma_j^2\right] - 1
\end{aligned}$$

where the last equality uses $4\mathbb{E}_{M_{1:i}}\left[\gamma_j\right] = 4\mathbb{E}_{M_{1:i}}\left[\mathbb{P}\left[Y_j = 1 \mid M_{1:i}\right]\right] = 4\mathbb{P}\left[Y_j = 1\right] = 2$. By similar reasoning, if we define $\eta_j = \mathbb{P}\left[Y_j = 1 \mid M_{1:i-1}\right]$ then we get

$$\mathbb{E}_{M_{1:i-1}}\left[\mathbb{E}\left[Y_j \mid M_{1:i-1}\right]^2\right] = 4\mathbb{E}_{M_{1:i-1}}\left[\eta_j^2\right] - 1.$$

Tracing back, our goal is now to upper bound

$$\mathbb{E}_{M_{1:i}}\left[\mathbb{E}\left[Y_j \mid M_{1:i}\right]^2\right] - \mathbb{E}_{M_{1:t-1}}\left[\mathbb{E}\left[Y_j \mid M_{1:i-1}\right]^2\right] = 4\left(\mathbb{E}_{M_{1:i}}\left[\gamma_j^2\right] - \mathbb{E}_{M_{1:i-1}}\left[\eta_j^2\right]\right). \tag{8.17}$$

Our analysis will be easier if we restrict the message space for $M_1, \ldots, M_i$ to be binary. We

do so by a result from Bassily and Smith [12]. This again relies on the local privacy of the protocol.

Lemma 30 (Theorem 4.1 in Bassily and Smith [12])**.** *Given a sequentially interactive ε-locally private protocol with expected number of randomizer calls T, there exists an equivalent sequentially interactive ε-locally private protocol with expected sample complexity $e^{\varepsilon}T$ where each user sends a single bit from a single randomizer call.*

The cost of this transformation is an e^{ε} blowup in expected sample complexity and an additional $O(n \log(\log(n)))$ bits of public randomness. First, since we assumed $\varepsilon = O(1)$, by Markov's inequality we can trade an arbitrarily small constant c decrease in overall success probability for a constant ($O(e^{\varepsilon}/c) = O(1)$) blowup in sample complexity. Combined with our assumption of arbitrary access to public randomness for locally private protocols, it is without loss of generality to assume all of our $M_1, \ldots, M_i$ are binary.[1]

Returning to $4\left(\mathbb{E}_{M_{1:i}}\left[\gamma_j^2\right] - \mathbb{E}_{M_{1:i-1}}\left[\eta_j^2\right]\right)$ in Equation (8.17), suppose we fix $M_{1:i-1}$ below. Then

$$\begin{aligned}
\mathbb{E}_{M_{1:i}}\left[\gamma_j^2\right] &= \mathbb{P}\left[M_i = 1\right] \cdot \mathbb{P}\left[Y_j = 1 \mid M_i = 1\right]^2 + \mathbb{P}\left[M_i = 0\right] \cdot \mathbb{P}\left[Y_j = 1 \mid M_i = 0\right]^2 \\
&= \frac{\left[\mathbb{P}\left[M_i = 1 \mid Y_j = 1\right] \cdot \mathbb{P}\left[Y_j = 1\right]\right]^2}{\mathbb{P}\left[M_i = 1\right]} + \frac{\left[\mathbb{P}\left[M_i = 0 \mid Y_j = 1\right] \cdot \mathbb{P}\left[Y_j = 1\right]\right]^2}{\mathbb{P}\left[M_i = 0\right]} \\
&= \eta_j^2 \left[\frac{\mathbb{P}\left[M_i = 1 \mid Y_j = 1\right]^2}{\mathbb{P}\left[M_i = 1\right]} + \frac{\mathbb{P}\left[M_i = 0 \mid Y_j = 1\right]^2}{\mathbb{P}\left[M_i = 0\right]}\right]
\end{aligned}$$

where the second equality uses Bayes' rule. Now, using $-2x + 2y - 2(1-x) + 2(1-y) = 0$ with $x = \mathbb{P}\left[M_i = 1 \mid Y_j = 1\right]$ and $y = \mathbb{P}\left[M_i = 1\right]$, we get

$$-2\mathbb{P}\left[M_i = 1 \mid Y_j = 1\right] + 2\mathbb{P}\left[M_i = 1\right] - 2\mathbb{P}\left[M_i = 0 \mid Y_j = 1\right] + 2\mathbb{P}\left[M_j = 0\right] = 0.$$

[1] Note that this step relies on the fact that, in sequentially interactive protocols, the number of randomizer calls is the same as the sample complexity. For fully interactive protocols, the number of randomizer calls may arbitrarily exceed the sample complexity. However, using the transformation given by Joseph et al. [47], our argument also extends to any $O(1)$-compositional fully interactive protocol.

We can now add 0 inside the bracketed term to get

$$\eta_j^2 \left[\frac{\mathbb{P}\left[M_i = 1 \mid Y_j = 1\right]^2}{\mathbb{P}\left[M_i = 1\right]} + \frac{\mathbb{P}\left[M_i = 0 \mid Y_j = 1\right]^2}{\mathbb{P}\left[M_i = 0\right]} \right] = \eta_j^2 \left[A + B\right]$$

where

$$\begin{aligned} A &= \frac{\mathbb{P}\left[M_i = 1 \mid Y_j = 1\right]^2 - 2\mathbb{P}\left[M_i = 1 \mid Y_j = 1\right]\mathbb{P}\left[M_i = 1\right] + 2\mathbb{P}\left[M_i = 1\right]^2}{\mathbb{P}\left[M_i = 1\right]} \\ &= \frac{\left(\mathbb{P}\left[M_i = 1 \mid Y_j = 1\right] - \mathbb{P}\left[M_i = 1\right]\right)^2}{\mathbb{P}\left[M_i = 1\right]} + \mathbb{P}\left[M_i = 1\right] \end{aligned}$$

and

$$\begin{aligned} B &= \frac{\mathbb{P}\left[M_i = 0 \mid Y_j = 1\right]^2 - 2\mathbb{P}\left[M_i = 0 \mid Y_j = 1\right]\mathbb{P}\left[M_i = 0\right] + 2\mathbb{P}\left[M_i = 0\right]^2}{\mathbb{P}\left[M_i = 0\right]} \\ &= \frac{\left(\mathbb{P}\left[M_i = 0 \mid Y_j = 1\right] - \mathbb{P}\left[M_i = 0\right]\right)^2}{\mathbb{P}\left[M_i = 0\right]} + \mathbb{P}\left[M_i = 0\right]. \end{aligned}$$

Thus we may rewrite $\eta_j^2[A + B]$ as

$$\eta_j^2 \left[1 + \frac{\left(\mathbb{P}\left[M_i = 1 \mid Y_j = 1\right] - \mathbb{P}\left[M_i = 1\right]\right)^2}{\mathbb{P}\left[M_i = 1\right]} + \frac{\left(\mathbb{P}\left[M_i = 0 \mid Y_j = 1\right] - \mathbb{P}\left[M_i = 0\right]\right)^2}{\mathbb{P}\left[M_i = 0\right]} \right].$$

For neatness, let $C = \mathbb{P}\left[M_i = 1 \mid Y_j = 1, J_i = j\right]$ and $D = \mathbb{P}\left[M_i = 1 \mid Y_j = -1, J_i = j\right]$. Recall that J_i denotes which of k bin pairs is chosen. Then

$$\begin{aligned} \mathbb{P}\left[M_i = 1 \mid Y_j = 1\right] &= \mathbb{P}\left[M_i = 1 \mid Y_j = 1, J_i \neq j\right] \cdot \mathbb{P}\left[J_i \neq j \mid Y_j = 1\right] \\ &\quad + \mathbb{P}\left[M_i = 1 \mid Y_j = 1, J_i = j\right] \cdot \mathbb{P}\left[J_i = j \mid Y_j = 1\right] \\ &= \frac{k-1}{k} \cdot \mathbb{P}\left[M_i = 1 \mid Y_j = 1, J_i \neq j\right] + \frac{C}{k} \end{aligned}$$

since J_i is independent of Y_j and $\mathbb{P}[J_i = j] = \frac{1}{k}$. Similarly,

$$\begin{aligned}
\mathbb{P}[M_i = 1] &= \mathbb{P}[M_i = 1 \mid J_i \neq j] \cdot \mathbb{P}[J_i \neq j] + \mathbb{P}[M_i = 1 \mid J_i = j] \cdot \mathbb{P}[J_i = j] \\
&= \frac{k-1}{k} \cdot \mathbb{P}[M_i = 1 \mid Y_j = 1, J_i \neq j] \\
&\quad + \frac{1}{k} \cdot \mathbb{P}[M_i = 1 \mid J_i = j, Y_j = 1] \cdot \mathbb{P}[Y_j = 1] \\
&\quad + \frac{1}{k} \cdot PM_i = 1 \mid J_i = j, Y_j = -1 \cdot \mathbb{P}[Y_j = -1] \\
&= \frac{k-1}{k} \cdot \mathbb{P}[M_i = 1 \mid Y_j = 1, J_i \neq j] + \frac{1}{k}(\eta_j C + (1-\eta_j)D)
\end{aligned}$$

where the second equality uses the conditional independence of M_i and Y_j given $J_t \neq j$ and fixed $M_{1:i-1}$. We substitute these expressions for $\mathbb{P}[M_i = 1 \mid Y_j = 1]$ and $\mathbb{P}[M_i = 1]$ and get

$$\begin{aligned}
(\mathbb{P}[M_i = 1 \mid Y_j = 1] - \mathbb{P}[M_i = 1])^2 &= \left[\frac{(1-\eta_j)(C-D)}{k}\right]^2 \\
&= (\mathbb{P}[M_i = 0 \mid Y_j = 1] - \mathbb{P}[M_i = 0])^2
\end{aligned}$$

where the last equality follows from $\mathbb{P}[M_i = 0 \mid Y_j = 1] = 1 - \mathbb{P}[M_i = 1 \mid Y_j = 1]$ and $\mathbb{P}[M_i = 1] = 1 - \mathbb{P}[M_i = 0]$. Returning to $\eta_j^2[A + B]$, we have

$$\begin{aligned}
\eta_j^2[A+B] &= \eta_j^2 \left[1 + \left(\frac{(1-\eta_i)(C-D)}{k}\right)^2 \left(\frac{1}{\mathbb{P}[M_i = 1]} + \frac{1}{\mathbb{P}[M_i = 0]}\right)\right] \\
&= \eta_j^2 \left[1 + \left(\frac{(1-\eta_i)(C-D)}{k}\right)^2 \cdot \frac{1}{\mathbb{P}[M_i = 1]\,\mathbb{P}[M_i = 0]}\right] \qquad (8.18)
\end{aligned}$$

since $\mathbb{P}[M_i = 1] + \mathbb{P}[M_i = 0] = 1$. We now analyze $\frac{|C-D|}{\mathbb{P}[M_i=1]}$. It will be useful to recall the sampling thought experiment used in the proof of Lemma 28: at each time t, we first uniformly sample bin pair $J_t \sim_U [k]$ and then sample the bin from a mixture: having sampled bin pair j, with probability $1-\alpha$ we take a uniform random draw from $\{2j-1, 2j\}$. With the remaining probability α, if $Y_j = 1$ then we sample $2j-1$, and if $Y_j = -1$ then we sample $2j$. Finally, we define $E_{j,t}^{\alpha} = 1$ if $J_t = j$ and we sample from the α mixture

component and $E^{\alpha}_{j,t} = 0$ otherwise.

Under this equivalent sampling method, we can rewrite

$$\begin{aligned}
C &= \mathbb{P}\left[M_i = 1 \mid Y_j = 1, J_i = j\right] \\
&= \mathbb{P}\left[M_i = 1 \mid E^{\alpha}_{j,i} = 1, Y_j = 1, J_i = j\right] \mathbb{P}\left[E^{\alpha}_{j,i} = 1 \mid Y_j = 1, J_i = j\right] \\
&\quad + \mathbb{P}\left[M_i = 1 \mid E^{\alpha}_{j,i} = 0, Y_j = 1, J_i = j\right] \mathbb{P}\left[E^{\alpha}_{j,i} = 0 \mid Y_j = 1, J_i = j\right] \\
&= \alpha\mathbb{P}\left[M_i = 1 \mid Y_j = 1, E^{\alpha}_{j,i} = 1\right] + (1-\alpha)\mathbb{P}\left[M_i = 1 \mid E^{\alpha}_{j,i} = 0, J_i = j\right]
\end{aligned}$$

where the last equality uses the fact that M_i is independent of J_i conditioned on $E^{\alpha}_{j,i} = 1$ and M_i is independent of Y_j conditioned on $E^{\alpha}_{j,i} = 0$. Similarly

$$D = \alpha\mathbb{P}\left[M_i = 1 \mid Y_j = -1, E^{\alpha}_{j,i} = 1\right] + (1-\alpha)\mathbb{P}\left[M_i = 1 \mid E^{\alpha}_{j,i} = 0, J_i = j\right].$$

Thus we can rewrite

$$\begin{aligned}
\frac{|C - D|}{\mathbb{P}\left[M_i = 1\right]} &= \frac{|\alpha(\mathbb{P}\left[M_i = 1 \mid Y_j = 1, E^{\alpha}_{j,i} = 1\right] - \mathbb{P}\left[M_i = 1 \mid Y_j = -1, E^{\alpha}_{j,i} = 1\right])|}{\mathbb{P}\left[M_i = 1\right]} \\
&\leq \frac{|\alpha(e^{\varepsilon} - e^{-\varepsilon})\mathbb{P}\left[M_i = 1\right]|}{\mathbb{P}\left[M_i = 1\right]} \\
&= O(\alpha\varepsilon)
\end{aligned}$$

where the inequality uses the ε-local privacy of M_i (recalling that we have been conditioning on $M_{1:i-1}$), and the equality uses $\varepsilon = O(1)$. Similarly, we get

$$\begin{aligned}
1 - C &= \mathbb{P}\left[M_i = 0 \mid Y_j = 1, J_i = j\right] \\
&= \alpha\mathbb{P}\left[M_i = 0 \mid Y_j = 1, E^{\alpha}_{j,i} = 1\right] + (1-\alpha)\mathbb{P}\left[M_i = 0 \mid E^{\alpha}_{j,i} = 0, J_i = j\right]
\end{aligned}$$

and

$$1 - D = \alpha\mathbb{P}\left[M_i = 0 \mid Y_j = -1, E^{\alpha}_{j,i} = 1\right] + (1-\alpha)\mathbb{P}\left[M_i = 0 \mid E^{\alpha}_{j,i} = 0, J_i = j\right].$$

This gives us

$$\begin{aligned}\frac{|C-D|}{\mathbb{P}\left[M_i=0\right]} &= \frac{|(1-C)-(1-D)|}{\mathbb{P}\left[M_i=0\right]} \\ &= \frac{|\alpha(\mathbb{P}\left[M_i=0 \mid Y_j=1, E^{\alpha}_{j,i}=1\right]-\mathbb{P}\left[M_i=0 \mid Y_j=-1, E^{\alpha}_{j,i}=1\right])|}{\mathbb{P}\left[M_i=1\right]} \\ &\leq \frac{|\alpha(e^{\varepsilon}-e^{-\varepsilon})\mathbb{P}\left[M_i=0\right]}{\mathbb{P}\left[M_i=0\right]} \\ &= O(\alpha\varepsilon)\end{aligned}$$

as well. Thus by Equation 8.18 $\eta_j^2[A+B] = \eta_j^2 + O\left(\frac{\eta_j^2(1-\eta_i)^2\alpha^2\varepsilon^2}{k^2}\right) = \eta_j^2 + O\left(\frac{\alpha^2\varepsilon^2}{k^2}\right)$ because $\eta_j^2(1-\eta_j)^2 < 1$. Returning to Equation (8.17), we can now bound

$$\mathbb{E}_{M_{1:i}}\left[\mathbb{E}\left[Y_j \mid M_{1:i}\right]^2\right] - \mathbb{E}_{M_{1:i-1}}\left[\mathbb{E}\left[Y_j \mid M_{1:i-1}\right]^2\right] = O\left(\frac{\alpha^2\varepsilon^2}{k^2}\right).$$

Since this analysis was for an arbitrary j, we get

$$\sum_{j=1}^{k}\left(\mathbb{E}_{M_{1:i}}\left[\mathbb{E}\left[Y_j \mid M_{1:i}\right]^2\right] - \mathbb{E}_{M_{1:i-1}}\left[\mathbb{E}\left[Y_j \mid M_{1:i-1}\right]^2\right]\right) = O\left(\frac{\alpha^2\varepsilon^2}{k}\right).$$

We substitute this into Equation (8.16) and get $I(X; V_t \mid M_{1:t-1}, J_t) = O\left(\frac{\alpha^4\varepsilon^2 t}{k^2}\right)$. Finally, substituting back into Equation (8.14) and using $t \leq m$ and $\varepsilon = O(1)$, $I(X; M_{1:m}) = O\left(\frac{\alpha^4\varepsilon^4 m^2}{k^2}\right)$. Since the output of a locally private algorithm is a function of the transcript, a uniformity tester with sample complexity m requires $I(X; M_{1:m}) = \Omega(1)$. We therefore get sample complexity $m = \Omega\left(\frac{k}{\alpha^2\varepsilon^2}\right)$. □

We conclude with a brief recap of this section. Focusing on the dependence on k, we have shown that the optimal sample complexity for ε-central, ε-pan, and sequentially interactive ε-local uniformity testing is respectively $\Theta(\sqrt{k})$, $\Theta(k^{2/3})$, and $\Theta(k)$, where the latter two results are new.

Chapter 9

Folklore and Future Directions

We conclude this dissertation by recapping two folklore results on pan-privacy which, to the best of our knowledge, have not appeared in writing elsewhere but are almost immediate from prior work. Possible directions for future work appear in the next section.

9.1. Pan-Privacy Folklore

First, pan-private summation of bits behaves (up to constants) like centrally private summation. This is because the optimal centrally private solution [34] is to compute the sum directly and add a draw of $\mathsf{Lap}(1/\varepsilon)$ noise. As applied in our pan-private tester (Theorem 12), the pan-private analogue simply adds an initial draw of $\mathsf{Lap}(1/\varepsilon)$ noise to a counter, increments the counter deterministically during the stream, and adds a second draw of $\mathsf{Lap}(1/\varepsilon)$ noise once the stream ends. This obtains the same asymptotic $O(1/\varepsilon)$ error and extends to real sums. In contrast, no locally private algorithm obtains $o(\sqrt{n})$ error [26]. We collect this information in the following corollary, which is a simple separation between pan- and local privacy.

Corollary 2. *For the problem of summing bits $x_1, \ldots, x_n$, there exist ε-central and ε-pan-private algorithms achieving additive error $O(1/\varepsilon)$, but for $\varepsilon < \ln(99)$ and $\delta < \frac{1}{4n}$, no (ε, δ)-locally private algorithm can achieve additive error $o(\sqrt{n})$.*

This basic result has implications beyond sums. For example, Canonne et al. [23] showed that the optimal centrally private simple hypothesis tester computes a sum of truncated log-likelihood ratios for each data point, adds Laplace noise scaled to the truncated sum's sensitivity, and compares the result to a threshold. As the basic algorithm is still a sum, we get a straightforward pan-private analogue with identical error as described above. However, our locally private hypothesis testing lower bound (Theorem 4) now separates pan-privacy

and local privacy as well. As before, we state a simpler but worse version of the guarantee of Canonne et al. [23] for brevity.

Corollary 3. *For the problem of simple hypothesis testing between distributions P_0 and P_1, there exist ε-central and ε-pan-private algorithms requiring only $O\left(\frac{1}{\varepsilon H^2(P_0,P_1)}\right)$ samples, but any (ε,δ)-locally private algorithm with $\delta < \min\left(\frac{\varepsilon^3\alpha^2}{48n\ln(2n/\beta)}, \frac{\varepsilon^2\alpha^2}{64n\ln(n/\beta)e^{7\varepsilon}}\right)$ requires at least $\Omega\left(\frac{1}{\varepsilon^2 H^2(P_0,P_1)}\right)$ samples.*

By a similar token, we can import results from central privacy to separate approximate and pure pan-privacy. The problem of releasing 1-way marginals is bit summation but for d-dimensional bit vectors. Focusing on the d parameter, Hardt and Talwar [44] showed that $\Omega(d)$ error is necessary for ε-central privacy, but $O(\sqrt{d})$ error is possible under (ε,δ)-central privacy by a single addition of Laplace noise (with the appropriate analysis). Both lower and upper bound thus extend to pan-privacy.

Corollary 4. *For the problem of releasing 1-way marginals with L_1 error α, every ε-pan-private algorithm requires $n = \Omega\left(\frac{d}{\alpha\varepsilon}\right)$ samples, but there is an (ε,δ)-pan-private algorithm that only requires $O\left(\frac{\sqrt{d\log(1/\delta)}}{\alpha\varepsilon}\right)$ samples.*

9.2. Future Directions

This dissertation has focused on exploring the relationships between the central, pan-, and local models of differential privacy. Many questions in this area remain. We conclude with a few such questions, ordered roughly by decreasing specificity:

1. How hard is parity?

   The problem of learning parity, and variants thereof, has proven useful for local privacy lower bounds [50]. For d-dimensional parity, $O(d)$ samples suffice in the central model but $\Omega(2^d)$ are required in the sequentially interactive local model. However, the sample complexity of learning parity under *fully interactive* local privacy is not known. The lower bounds of Kasiviswanathan et al. [50] only guarantee that such a fully interactive

protocol would require communication that is exponential in d. This still does not tell us anything about sample complexity. Along similar lines, the sample complexity of pan-private parity learning is unknown beyond the immediate $\Omega(d)$ and $O(2^d)$ bounds. Unfortunately, the central solutions given by Kasiviswanathan et al. [50] rely crucially on having d raw labelled examples as input to the algorithm (the first, inefficient solution passes these to the exponential mechanism; the second, efficient solution uses Gaussian elimination and then post-processes the solution in a randomized way). This makes adapting centrally private parity learners for pan-privacy difficult.

2. What is the relationship between pan-privacy and robust shuffle privacy?

   Very recent work by Balcer, Cheu, Joseph, and Mao [10] showed that the $\Theta(\sqrt{k})$ and $\Theta(k^{2/3})$ dependencies on the domain size k for pan-private distinct elements and uniformity testing, respectively, also extend to the model of robust shuffle privacy. It is not known whether (1) a more general connection between pan-privacy and robust shuffle privacy exists, or (2) there is a problem separating the two models.

3. How useful is approximate privacy beyond the central model?

   By the work of Bun et al. [19] and Cheu et al. [27] (combined here as Lemma 7), approximate and pure sequentially interactive local privacy are essentially equivalent. However, a naive extension of their results to the fully interactive setting requires $\delta = o(1/T)$, where T is the number of randomizer calls. It is not clear if this dependence is necessary. At the same time, it is not known if there are any problems where (necessarily fully interactive) approximate local privacy obtains meaningfully better utility than pure local privacy. In pan-privacy, the only known separation is inherited directly from central privacy (Corollary 4).

4. Do any practical problems separate sequentially and fully interactive local privacy?

   The only problems known to separate sequentially and fully interactive local privacy

are very artificial. For example, when $k = 3$ the tree in the hidden layers problem has a number of leaves well in excess of the estimated number of atoms in the universe, but the problem may be solved with $O(1)$ users absent local privacy. It would be useful to either find practical separations or show that that, in some meaningful sense, they do not exist. Theorem 2 offers a first step in the second direction by implying that any such problem would still require $\omega(1)$ randomizer calls to a single user.

5. What is the relationship between pan-privacy and algorithm memory?

   All papers on pan-privacy [8, 36, 53] have observed either informal or formal connections between pan-privacy and algorithms with limited memory. Mir et al. [53] borrowed techniques from the sketching literature to show that a pan-private algorithm can solve the problem of counting distinct elements in a stream to optimal error using polylogarithmic space, and showed that additional space does not help. Amin et al. [8] adapted a lower bound for limited memory uniformity testing to prove their pan-private uniformity testing lower bound. Intuitively, this makes sense: maintaining a private state forbids memorizing data, which should lead to lower memory requirements overall. However, no formal statement connecting pan-privacy and memory is known.

BIBLIOGRAPHY

[1] Jayadev Acharya, Ashkan Jafarpour, Alon Orlitsky, and Ananda Suresh. A competitive test for uniformity of monotone distributions. In *Artificial Intelligence and Statistics (AISTATS)*, 2013. 97

[2] Jayadev Acharya, Constantinos Daskalakis, and Gautam Kamath. Optimal testing for properties of distributions. In *Neural Information Processing Systems (NIPS)*, 2015. 97, 99

[3] Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. Differentially private testing of identity and closeness of discrete distributions. In *Neural Information Processing Systems (NeurIPS)*, 2018. 96, 97, 103, 109

[4] Jayadev Acharya, Clément Canonne, Cody Freitag, and Himanshu Tyagi. Test without trust: Optimal locally private distribution testing. In *Artificial Intelligence and Statistics (AISTATS)*, 2019. 12, 96, 97, 103, 105, 108

[5] Jayadev Acharya, Clément L Canonne, Yanjun Han, Ziteng Sun, and Himanshu Tyagi. Domain compression and its application to randomness-optimal distributed goodness-of-fit. *arXiv preprint arXiv:1907.08743*, 2019. 104, 108

[6] Jayadev Acharya, Clément L Canonne, and Himanshu Tyagi. Inference under information constraints: Lower bounds from chi-square contraction. In *Conference on Learning Theory (COLT)*, 2019. 12

[7] Maryam Aliakbarpour, Ilias Diakonikolas, and Ronitt Rubinfeld. Differentially private identity and equivalence testing of discrete distributions. In *International Conference on Machine Learning (ICML)*, 2018. 103

[8] Kareem Amin, Matthew Joseph, and Jieming Mao. Pan-private uniformity testing. *arXiv preprint arXiv:1911.01452*, 2019. 3, 14, 96, 131

[9] Differential Privacy Team Apple. Learning with privacy at scale. Technical report, Apple, 2017. 1

[10] Victor Balcer, Albert Cheu, Matthew Joseph, and Jieming Mao. Connecting robust shuffle privacy and pan-privacy. *arXiv preprint arXiv:2004.09481*, 2020. 130

[11] Borja Balle, James Bell, Adria Gascon, and Kobbi Nissim. The privacy blanket of the shuffle model. In *International Cryptology Conference (CRYPTO)*, 2019. 21, 29, 30

[12] Raef Bassily and Adam Smith. Local, private, efficient protocols for succinct histograms. In *Symposium on the Theory of Computing (STOC)*, 2015. 64, 123

[13] Amos Beimel, Kobbi Nissim, and Eran Omri. Distributed private data analysis: Simultaneously solving how and what. In *International Cryptology Conference (CRYPTO)*, 2008. 5

[14] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Symposium on Operating Systems Principles (SOSP)*, 2017. 1

[15] Mark Braverman. A hard-to-compress interactive task? In *Allerton Conference on Communication, Control, and Computing (Allerton)*, 2013. 68

[16] Mark Braverman and Jieming Mao. Simulating noisy channel interaction. In *Innovations in Theoretical Computer Science (ITCS)*, 2015. 66

[17] Mark Braverman and Anup Rao. Toward coding for maximum errors in interactive communication. *IEEE Transactions on Information Theory*, 2014. 67

[18] Mark Braverman, Ankit Garg, Tengyu Ma, Huy L Nguyen, and David P Woodruff. Communication lower bounds for statistical estimation problems via a distributed data processing inequality. In *Symposium on the Theory of Computing (STOC)*, 2016. 42, 43, 51, 55

[19] Mark Bun, Jelani Nelson, and Uri Stemmer. Heavy hitters and the structure of local privacy. In *Principles of Database Systems (PODS)*, 2018. 41, 130

[20] Bryan Cai, Constantinos Daskalakis, and Gautam Kamath. Priv'it: private and sample efficient identity testing. In *International Conference on Machine Learning (ICML)*, 2017. 97, 103

[21] Clément L Canonne. A survey on distribution testing: Your data is big. but is it blue? In *Electronic Colloquium on Computational Complexity (ECCC)*, 2015. 96

[22] Clément L. Canonne. A short note on poisson tail bounds, 2017. URL http://www.cs.columbia.edu/~ccanonne/files/misc/2017-poissonconcentration.pdf. 96

[23] Clément L. Canonne, Gautam Kamath, Audra McMillan, Adam Smith, and Jonathan Ullman. The structure of optimal private tests for simple hypotheses. In *Symposium on the Theory of Computing (STOC)*, 2019. 39, 44, 128, 129

[24] Siu-On Chan, Ilias Diakonikolas, Paul Valiant, and Gregory Valiant. Optimal algorithms for testing closeness of discrete distributions. In *Symposium on Discrete Algorithms (SODA)*, 2014. 97

[25] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Transactions on Information and System Security*, 2011. 5, 46

[26] TH Hubert Chan, Elaine Shi, and Dawn Song. Optimal lower bound for differentially private multi-party aggregation. In *European Symposium on Algorithms (ESA)*, 2012. 5, 128

[27] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Conference on the Theory and Applications of Cryptographic Techniques (CRYPTO)*, 2019. 41, 130

[28] Amit Daniely and Vitaly Feldman. Learning without interaction requires separation. In *Neural Information Processing Systems (NeurIPS)*, 2019. 5, 73

[29] Ilias Diakonikolas, Themis Gouleakis, Daniel M. Kane, and Sankeerth Rao. Communication and memory efficient testing of discrete distributions. In *Conference on Learning Theory (COLT)*, 2019. 108, 109, 110, 113

[30] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In *Neural Information Processing Systems (NIPS)*, pages 3574–3583, 2017. 1

[31] John Duchi and Ryan Rogers. Lower bounds for locally private estimation via communication complexity. In *Conference on Learning Theory (COLT)*, 2019. 5, 108

[32] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy, data processing inequalities, and statistical minimax rates. *arXiv preprint arXiv:1302.3203*, 2013. 5, 12, 42, 44, 86, 89, 121

[33] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *Foundations of Computer Science (FOCS)*, 2013. 5

[34] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC)*, 2006. 1, 2, 5, 7, 10, 128

[35] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. Differential privacy under continual observation. In *Symposium on the Theory of Computing (STOC)*, 2010. 4

[36] Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N Rothblum, and Sergey Yekhanin. Pan-private streaming algorithms. In *Innovations in Computer Science (ICS)*, 2010. 2, 4, 5, 9, 108, 131

[37] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 2014. 7, 99, 115

[38] Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Principles of Database Systems (PODS)*, 2003. 2, 5, 10

[39] Marco Gaboardi, Ryan Rogers, and Or Sheffet. Locally private mean estimation: Z-test and tight confidence intervals. In *Artificial Intelligence and Statistics (AISTATS)*, 2019. vi, 47, 48

[40] Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of communication and external information. In *Symposium on the Theory of Computing (STOC)*, 2016. 68, 69

[41] Garfinkel, Simson L. Deploying differential privacy for the 2020 census of population and housing, 2019. Accessed: 09-12-2019. 1

[42] Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 2000. 96, 97

[43] Miguel Guevara. Enabling developers and organizations to use differential privacy. developers.googleblog.com/2019/09/enabling-developers-and-organizations.html, 2019. Accessed: 09-12-2019. 1

[44] Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Symposium on the Theory of Computing (STOC)*, 2010. 41, 129

[45] Svante Janson. Tail bounds for sums of geometric and exponential variables. *arXiv preprint arXiv:1709.08157*, 2017. 38

[46] Matthew Joseph, Janardhan Kulkarni, Jieming Mao, and Zhiwei Steven Wu. Locally private gaussian estimation. In *Neural Information and Processing Systems (NeurIPS)*, 2019. 3, 48

[47] Matthew Joseph, Jieming Mao, Seth Neel, and Aaron Roth. The role of interactivity in local differential privacy. In *Foundations of Computer Science (FOCS)*, 2019. 3, 20, 39, 123

[48] Matthew Joseph, Jieming Mao, and Aaron Roth. Exponential separations in local differential privacy. In *Symposium on Discrete Algorithms (SODA)*, 2020. 3, 39, 59, 72, 108

[49] Vishesh Karwa and Salil Vadhan. Finite Sample Differentially Private Confidence Intervals. In *Innovations in Theoretical Computer Science (ITCS)*, 2018. 47

[50] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *Symposium on the Theory of Computing (STOC)*, 2008. 2, 5, 10, 21, 73, 129, 130

[51] Andrew McGregor, Ilya Mironov, Toniann Pitassi, Omer Reingold, Kunal Talwar, and Salil Vadhan. The limits of two-party differential privacy. In *Foundations of Computer Science (FOCS)*, 2010. 4, 108

[52] Solomon Messing, Christina DeGregorio, Bennett Hillenbrand, Gary King, Saurav Mahanti, Chaya Nayak, , Nathaniel Persily, Bogdan State, and Arjun Wilkins. Facebook privacy-protected urls light table release. socialscience.one/files/partnershipone/files/facebook_urls-light_codebook_v2.0.pdf, 2019. Accessed: 09-18-2019. 1

[53] Darakhshan Mir, Shan Muthukrishnan, Aleksandar Nikolov, and Rebecca N Wright. Pan-private algorithms via statistics on sketches. In *Principles of Database Systems (PODS)*, 2011. 4, 5, 9, 108, 131

[54] Jack Murtagh, Kathryn Taylor, George Kellaris, and Salil Vadhan. Usable differential privacy: A case study with psi. *arXiv preprint arXiv:1809.04103*, 2018. 1

[55] Jerzy Neyman and Egon Sharpe Pearson. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 1933. 39

[56] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Symposium on the Theory of Computing (STOC)*, 2007. 44

[57] Liam Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Transactions on Information Theory*, 2008. 96, 97, 109

[58] Ryan M. Rogers, Aaron Roth, Jonathan Ullman, and Salil Vadhan. Privacy odometers and filters: Pay-as-you-go composition. In *Neural Information Processing Systems (NIPS)*, 2016. 23

[59] Leonard J Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 1996. 67

[60] Maurice Sion. On general minimax theorems. *Pacific Journal of mathematics*, 1958. 45

[61] Gregory Valiant and Paul Valiant. An automatic inequality prover and instance optimal identity testing. In *Foundations of Computer Science (FOCS)*, 2014. 96, 97

[62] Stanley L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 1965. 10, 11