SHF: Small: Collaborative Research: Programming Tools for Adaptive Data Analysis

Marco Gaboardi (PI)
Department of Computer Science and Engineering
University at Buffalo, The State University of New York
338 Davis Hall
Buffalo, NY, 14260

Jonathan Ullman (PI)
College of Computer and Information Science
Northeastern University
260 West Village H
Boston, MA, 02115

A proposal for Computing and Communication Foundations (CCF): Core Programs
NSF program solicitation 16-578

Project Summary

Overview. A *false discovery* occurs when an empirical researcher draws a conclusion based on his or her dataset that does not *generalize* to new data. False discovery have been identified as a growing crisis in experimental science. Moreover, this problem persists despite decades of research by the statistics community. A contributing factor to this crisis is that most statistical tools are designed for static data analysis, where a dataset is used once, whereas modern data analysis is adaptive, and the same datasets are reused in complex ways that are difficult to formalize and analyze.

The goal of this project is to combat false discovery by developing algorithmic and programming language tools to ensure statistically sound adaptive data analysis.

Our project builds on a recent surprising connection between false discovery and differential privacy, a robust statistical guarantee that emerged recently to protect the privacy of sensitive data. This line of work shows that when data is analyzed in a differentially private way, then false discoveries cannot occur. Differential privacy is also programmable, and allows complex differentially private algorithms to be built from simple components, so it is an ideal programming framework for adaptive data analysis. Thus, the technical core of our project is to extend the body of work on differentially private programming frameworks to adaptive data analysis. Specifically, we will develop new algorithmic and programming languages tools for adaptive data analysis, and incorporate them into the first prototype system for this application.

Intellectual Merit. Our work will bridge approaches from both algorithms and programming languages to design new algorithms and tools for preventing false discovery in adaptive data analysis via the connection to differential privacy. Our contributions will include:

1. New strong composition theorems for differential privacy and related notions that are tailored to adaptive data analysis, and new formal reasoning methods for these composition theorems.

2. Computationally efficient differentially private algorithms and implementations that ensure privacy prevent false discovery for large numbers of arbitrary, adaptively chosen analyses.

3. A prototype programming system for adaptive data analysis, combining existing differentially private tools with those we develop during this project. This work will significantly extend existing differentially private programming frameworks that are either specific to static analysis or not tailored to the application to adaptive data analysis.

Broader Impacts. Given the increased reliance on machine learning and empirical science in all aspects of life ranging from policy to education to healthcare, we believe this project will have a broad and long-lasting impact. A study in *Nature* has estimated the cost of non-reproducible research to be $28 billion in the biological sciences alone [2], and researchers have identified issues of adaptivity in data analysis as a contributing factor to this problem [44]. Our prototype system will be a first step towards building the necessary tools to combat this problem. Our research will also build bridges between algorithmic and programming languages approaches to the false discovery problem.

Keywords: adaptive data analysis; preventing false discovery; differential privacy; programming languages; type systems; formal verification; algorithms

Contents

A Project Description

A.1 Introduction

Suppose we have a dataset x sampled from some unknown population P. A *false discovery* occurs when a data analyst draws a conclusion based on the dataset x that does not *generalize* to the population P, and the conclusion is unlikely to be *reproduced* on an independent dataset sampled from P. For decades, statisticians have been developing methods to prevent false discovery, including the Bonferroni correction [19, 27] and Benjamini-Hochberg procedure [15]. And yet, false discovery remains a major problem in the scientific community, leading to provocative articles in the popular and scientific press like "Why Most Published Research Findings are False" [50], "The Statistical Crisis in Science" [44], and "Trouble at the Lab" [37].

Why does false discovery occur and why is it so difficult to prevent? Common explanations include the misapplication of statistical methods, multiple comparisons (sometimes called *fishing*, *data dredging*, or *p-hacking*), publication bias, and fraud. While these abuses undoubtedly occur, there is also a disconnect between statistical tools and the way they are used. Statistical methods are typically analyzed as a *static* process where a hypothesis is formed, then data is collected, and finally a conclusion is reached and the data is discarded. In reality, data analysis is an *adaptive* process where the initial conclusion is used to formulate more hypotheses that will be tested against the same dataset. Adaptive data analysis—dubbed "the garden of forking paths" [44]—makes it nearly impossible to even specify the statistical procedure, and challenging to prevent false discovery without severely compromising statistical accuracy.

This problem was recently formalized and investigated in a pair of works, one by Dwork et al. [30] and one by Hardt and PI Ullman [49], revealing a close connection between interactive data analysis and *differential privacy* [31] that yields state-of-the-art methods for prevent false discovery in interactive data analysis. Intuitively, differential privacy is a strong notion of *algorithmic stability* that prevents false discovery in interactive data analysis and yet allows a high degree of statistical power. These results were subsequently been strengthened and extended to other notions of stability in a rapidly evolving line of work [28, 67, 14, 18, 64, 12, 39, 72].

The most attractive feature of differential privacy and related notions of stability is that they are "programmable." They each come with parameters that control the "level of stability" and these parameters degrade gracefully under arbitrary adaptive composition, meaning that programmers can assemble complex algorithms out of simple components without having to derive new proofs that their algorithms are stable.

Programmability makes differential privacy and its relatives an ideal *programming framework* for adaptive data analysis, with the potential to allow non-experts to analyze data freely without false discovery. Several programming languages and systems to support differential privacy have been proposed, e.g. [57, 63, 59, 11, 42, 8], but none of them has been designed with adaptive analysis in mind, and none is tailored to preventing false discovery. Our project has two broad and tightly connected objectives: solve use-inspired algorithmic and programming languages challenges in the application of differential privacy to adaptive data analysis, then incorporate these solutions into a prototype programming system. Specific contributions include:

- Optimal composition theorems for differential privacy and related notions that are tailored to the application to adaptive data analysis, building on the recent *stability odomters* of PI Ullman [61].
- New formal reasoning principles for the verification of strong composition for adaptive differentially private data analysis and stability odometers, building on recent work by PI Gaboardi [6, 5].
- New computationally efficient stable algorithms specialized to our application domain of *adaptive data analysis*, building on recent advances by PI Gaboardi et al. [41] and PI Ullman [66].
- A publicly available prototype system for adaptive data analysis, incorporating the above contributions. We will do a pilot evaluation and make the prototype available for feedback.

Even though our goal is preventing false discovery, and not differential privacy *per se*, many of our methods will also advance the state-of-the-art in interactive differentially private algorithms and systems.

A.1.1 Background and Technical Approach

Approaches to Adaptive Data Analysis. Most work in Statistics on false discovery is for static data analysis, and the guarantees are simply invalid if the dataset is reused. Most of the work on adaptive data analysis in the Statistics literature considers how to correct for *structured adaptivity* where the algorithm may be adaptive, but in a specific way that can be analyzed. The most common example are algorithms that run a specific model-selection procedure and then perform inference on the same dataset (see e.g. [16, 40, 68]).

In contrast, the recent approach initiated in Computer Science [30, 49] deals with *unstructured adaptivity*, where we allow an arbitrary procedure to adaptively choose how the dataset is used and reused. This approach captures realistic use cases like complex iterative optimization methods, human researchers making subtle and informal decisions about the design of the study ("researcher degrees of freedom" [44]), and multi-researcher reuse of datasets. Thus, this recent line of work is well suited to the design of programming frameworks that assume very little about the decisions of the user.

Evaluation. In light of the discussion above, in the evaluation phase of our project, described in Section A.5.1, we will consider both types of uses cases. That is, we want to make sure that our system is competitive with tailored methods for workhorse tasks like inference-after-model-selection, but we also want to make sure that our system prevents overfitting on the sorts of *adversarial* use-cases described in the literature [49, 30, 66, 18] without too much loss of utility. By making our system available to researchers, we also hope to gather feedback on intermediate use case that arise in practice.

Differential Privacy and Algorithmic Stability. There is a rich theory in Statistics and machine learning on preventing false discovery. This theory says that preventing false discovery in static data analysis is essentially equivalent to *output stability* [25, 26, 53, 20, 65]. Informally, an algorithm is output stable if changing one sample in the dataset does not change the output too much in some metric. These notions do not prevent false discovery in adaptive data analysis because they do not compose—if we run two output stable algorithms A_1, A_2, and A_2 depends on the output of A_1, then the composed algorithm is not necessarily output stable and does not prevent false discovery.

In contrast, *differential privacy* [31] is a type of *distributional stability*. A differentially private algorithm is randomized, and changing one sample in the dataset does not change the probability distribution on outputs too much in an appropriate sense. Formally, an algorithm $A : X^n \to R$ is (ε, δ)-differentially private if for every dataset $x = (x_1, \ldots, x_n) \in X^n$ and every dataset x' that differs from x on at most one entry, and every event $S \subseteq R$, $\mathbb{P}[A(x) \in S] \leq e^{\varepsilon} \cdot \mathbb{P}[M(D') \in S] + \delta$. Intuitively, we can think of ε as a privacy/stability level and δ as a probability of total failure.

It was recently shown by Dwork et al. [30], that differential privacy prevents false discovery, with tight bounds given by PI Ullman and collaborators [13]. Moreover, since differential privacy is well known to be preserved under adaptive composition (with graceful degradation of ε and δ), it is suitable for preventing false discovery in adaptive data analysis, making it possible to build stable algorithms in a modular fashion. For concreteness, our project description focuses on differential privacy, but our system will also consider some other related notions of distributional stability that have been shown to be useful for adaptive data analysis [29, 64, 12, 60].

Preventing False Discovery with Differential Privacy. For context, we give an example of the strength of differential privacy for adaptive data analysis. Suppose P is a population we want to study and x is a

dataset of n samples from P. We would like to estimate a sequence of statistics $\phi_1(P), \ldots, \phi_t(P)$, which may be adaptively chosen so that ϕ_i depends on the outcome of $\phi_1, \ldots, \phi_{i-1}$. For this example, we consider *statistical queries* [52], which ask for the probability that a sample from P satisfies some property, although similar results and phenomena apply to much more general families of queries [13].

If the t statistics are *static*, then we can answer each query with the empirical mean $\phi_i(x)$, and guarantee error $\approx (\frac{\log(t)}{n})^{1/2}$. However, if the statistics may be adaptively chosen, then this naïve approach may overfit badly, and have exponentially larger error $\approx (\frac{t}{n})^{1/2}$. A better way to answer the queries is to use differential privacy, and answer each query ϕ_i with $\phi_i(x)+Z$ where Z is appropriately calibrated, independent Gaussian noise. Using the tight bounds of PI Ullman and collaborators [13], this approach guarantees error $\approx (\frac{\sqrt{t}}{n})^{1/2}$, which is already a quadratic improvement over the naïve approach! Even more surprisingly, there are algorithms for adaptive queries that nearly match the performance of the naïve algorithm for static queries [30, 13], meaning that we can answer *exponentially* many queries without overfitting.

Although these algorithms are practical in some use-cases like data science competitions [18], their worst-case running time necessarily scales exponentially in the dimension of the data [49, 66]. In Section A.4, we describe our proposed research on making such algorithms efficient in high-dimensional data, building on recent work by PI Gaboardi and collaborators [41] for static differential privacy.

Composition of Differential Privacy. As we have discussed, differential privacy satisfies very appealing *composition theorems*. If we run t algorithms $A_1, \ldots, A_t$ adaptively chosen algorithms on a dataset x, and each one is (ε, δ)-differentially private, then the resulting algorithm satisfies $(\varepsilon t, \delta t)$-differential privacy [31]. A more careful analysis shows that the resulting algorithm satisfies roughly $(\varepsilon\sqrt{t\log(1/\delta)}, \delta t)$-differential privacy [34], which is much better in the realistic case where t is not too small. We even know several composition theorems that are exactly optimal for every sequence of privacy/stability parameters [51, 58]. These strong composition guarantees where ε degrades sublinearly in t, are crucial for improving over naïve approaches to preventing false discovery. For example, the quadratic improvement described above is only possible using strong composition theorems.

Current composition theorems have an important restriction—while they allow adaptive choice of algorithms, the privacy bounds $(\varepsilon_i, \delta_i)$ for each algorithm must be fixed in advance, which limits our ability to make full use of adaptivity to optimize the algorithm. Moreover, composition theorems in differential privacy are used as a *filter* that shuts off the dataset when a certain privacy budget is depleted. In adaptive data analysis there is no inherent budget we can tolerate, and a more flexible approach is to use composition theorems like an *odometer* which tells us how much to correct for potential overfitting. These questions were first addressed by PI Ullman and collaborators [61], but the concrete bounds given in their work are too loose for our system. In Section A.2 we discuss the challenges in designing useful stability odometers that our work will address.

Programming Language Tools for Differential Privacy. Manually checking that a given query or program is differentially private can be subtle and tedious. The implementation must precisely respect the algorithm designer's specification of the randomness—too much noise makes the answer inaccurate, too little leads to failures of privacy or stability. For this reason, several tools have been proposed to assist and/or automate in the process of checking whether a given program is differentially private or not. By using these tools, usually a programmer cannot directly access to the data; instead, he has at his disposal special-purpose mechanisms to reason about differential privacy for his programs. In the collection of tools proposed so far we can see three approaches based on the techniques, the level of automation, and the level of generality.

The first approach ensures differential privacy using a combination of program annotations (describing the amount of noise each component needs) and runtime program analysis (checking that the total amount of noise respects the differential privacy bound). This approach was first proposed by McSherry and implemented in

PINQ [57]. A similar idea is also used in [63]. More recently, PINQ has been extended using provenance techniques to track the privacy loss of each individual [36]. Interestingly, this technique can also applied in a setting where queries data analysis are performed in an interactive fashion [35], but does not give sublinear composition. Unfortunately this approach provides few formal guarantees.

The second approach provides fully verified differential private programs and it is based on a probabilistic approximate relational Hoare logic built on top of the Coq proof assistant. This approach has been implemented in CertiPriv [11], and further extended in [4]. This approach provides very strong formal guarantees but it is of difficult use and it is hard to automate. To make it more practical, PI Gaboardi and collaborators have provided a translation of the probabilistic approximate relational logic in standard Hoare logic [7]. An interesting outcome of this research direction has been the development of techniques for reasoning about differential privacy by means of probabilistic coupling [9, 6], which enables verification of algorithms whose privacy analysis is based on properties other than merely composition of differential privacy. A similar technique but with more automation has also been proposed recently in [73].

Finally, the third approach uses type systems ideas to ensure differential privacy. More precisely, this approach uses quantitative type-based analysis that checks at compile-time whether the amount of added noise ensures differential privacy. An approach based on types have been implemented in the systems *Fuzz* [59, 24, 46] (further extended in [38]), *DFuzz* [42], and HOARe^2 [8]. This approach combines strong formal guarantees with the use of some trusted primitives, balancing automation with expressiveness.

Among the different approaches, the one that is relevant for this proposal is the approach followed by PI Gaboardi and collaborators for the design of the system HOARe^2 [8] (short for *higher order relational refinement type system*). The idea of this approach is to consider differential privacy as an *approximate probabilistic relational property*, i.e. as a probabilistic property that can be described as an approximate relation between two runs of a program on two adjacent inputs. Here *approximate* refers to the fact that the definition depends on the privacy parameters ϵ and δ. This approach has been further extended by PI Gaboardi and collaborators [5] to address Bayesian inference and the use of f-divergences. Unfortunately, this approach lacks the machinery needed to support reasoning principles for verifying strong composition schemes. In Section A.3, we discuss the challenges provided by these scheme and how they will be addressed by our approach.

A.2 Measuring Overfitting with Stability Odometers

As we discussed, distributional stability notions like differential privacy have the remarkable property that they degrade gracefully under adaptive composition. That is, if $A_1, \ldots, A_t$ are adaptively chosen algorithms, and each one is $(\varepsilon_i, \delta_i)$-differentially private, then the composed algorithm is roughly $(\varepsilon\sqrt{t\log(1/\delta)}, \delta t)$-differentially private [34], with generalizations to the case where each algorithms satisfies a different $(\varepsilon_i, \delta_i)$.

In the context of programming frameworks for differential privacy, these composition theorems act as a *privacy/stability filter*. That is, if we have a global constraint $(\varepsilon_g, \delta_g)$ that must be satisfied, we can design a function $\mathsf{FIL}_{(\varepsilon_g,\delta_g)}(\varepsilon_1, \delta_1, \ldots, \varepsilon_t, \delta_t)$ that decides if the global budget is satisfied by the composition of *any* set of algorithms satisfying the individual constraints $(\varepsilon_i, \delta_i)$.

Composition of differential privacy has been extensively studied (e.g. [31, 34, 29, 35, 33, 21]) for their centrality to the study of privacy. However, these composition theorems are insufficient for the application to adaptive data analysis for two reasons:

- These composition theorems all assume that both the length of the composition t and the individual constraints $(\varepsilon_i, \delta_i)$ are fixed in advance. To handle adaptive choices of these parameters, they require a conservative choice that covers all eventualities. The result is that there is no way to take advantage of, say, iterative algorithms that make adaptive decisions to stop early or increase/decrease the individual parameters $(\varepsilon_i, \delta_i)$ adaptively in order to achieve better utility. Such iterative methods are extremely common in machine learning and statistics (e.g. first-order optimization procedures).

- In adaptive data analysis, there is no inherent reason to set a global constraint on the level of stability. We only use stability to know how much overfitting might have occurred and correct for it. Thus, instead of a privacy filter, we would prefer a *privacy/stability odometer* that simply measures the current level of stability. A privacy odometer is a function $\mathsf{ODO}(\varepsilon_1, \delta_1, \varepsilon_2, \delta_2, \dots)$ that outputs a running measure $(\varepsilon_o, \delta_o)$ of the degree of stability at every point. Unlike a filter, where we only want to know when to shut down the dataset, an odometer must give a correct upper bound on how much to correct for overfitting at *every time step*.

Recent work by PI Ullman and collaborators [61] introduced the notion of privacy odometers for the adaptive-parameter setting, and constructed odometers and filters for the adaptive-parameter setting that are competitive with the optimal fixed-parameter composition theorem of Dwork et al. [34].

We believe that odometers are the right model of composition for the application of differential privacy to adaptive data analysis. The aforementioned preliminary results demonstrate that we can achieve the extra flexibility of privacy odometers with little asymptotic cost, but the concrete performance is too weak for practical use cases. Below we discuss several algorithmic and analytical directions towards achieving optimal privacy odometers. In Section A.3 we will discuss challenges for verifying these tools and incorporating them into our pilot system.

A.2.1 Towards Optimal Stability Odometers

Tight Constants. The state-of-the-art odometer [61] is asymptotically optimal, but suffers from large hidden constants. In contrast, the asymptotically optimal composition theorem for the fixed-parameter setting [34] (henceforth, "the DRV composition theorem") has very small hidden constants and good concrete performance for realistic parameters. This gap is at least partially inherent, as PI Ullman and collaborators have showed that if one allows an extremely large number of rounds of composition (roughly doubly exponential in the size of the dataset), then the true level of stability can sometimes be much larger than the DRV theorem. However, we believe that in reasonable use cases the gap between the odometer and the DRV theorem is simply the result of loose analysis, and tightening this analysis (or showing that it cannot be tightened) is an important problem.

Goal 1. *Construct an asymptotically optimal privacy odometer with tight constant factors.*

As a first step we will consider the special case where the privacy parameters (ε, δ) are the same in every iteration and only the number of rounds of composition (i.e. when to stop) can be chosen adaptively. This setting captures the realistic use case where one wants to study the data using some iterative approximation procedure (e.g. gradient descent) with an adaptive stopping condition. For this case, PI Ullman and Harvard undergraduate Chan Kang have performed a preliminary numerical study, in which they have showed that the DRV theorem can be too low by between 0.5% and 8% in some cases. For these cases the privacy odometer theorem gives a privacy level that is larger than the DRV theorem by nearly 200%. We are confident that one can rigorously establish the privacy odometer for this special case that is within about 8% of the DRV theorem. We further conjecture that this special case is the "hardest" setting for privacy odometers, and one can obtain a tight odometer that is within about 8% of the DRV theorem for all realistic cases.

Our technical approach is to leverage a connection between privacy odometers and the *law of the iterated logarithm* [54]. At a very high-level, the evolution of the privacy parameter over time can be seen as a certain kind of random walk [34, 51, 58, 61], and a privacy odometer can thus be viewed as a bound on the maximum over all time steps of how much the random walk strays from its mean. The law of the iterated logarithm give a sharp bound on a *specific* random walk *in the limit*. In our applications, we need finite-time bounds, and to handle adaptive choices of the parameters we need sharp bounds on a more general family of random walks. We suspect that finite-time versions of the law of the iterated logarithm due to Balsubramani [3] are sufficient to handle special cases, but resolving the general case will require a novel analysis tailored to our application.

Strong Optimality. The privacy odometer and the DRV composition theorem are both asymptotically optimal (up to constants) for a large number of rounds. However, in the fixed-parameter setting, recent work by Kairouz, Oh, and Viswanath [51] as well as Murtagh and Vadhan [58] showed that the DRV theorem can be rather loose when the number of rounds is relatively small. The issue is essentially that for a specific number of rounds, one may prefer the weaker composition bound of $(\varepsilon t, \delta t)$ to the generally stronger bound $(\varepsilon\sqrt{t\log(1/\delta)}, \delta t)$, and there is a "grey area" in the middle where neither bound is quite right.

The work [51, 58] gave privacy filters that are *strongly optimal*—for every sequence $\varepsilon_1, \delta_1, \ldots, \varepsilon_t, \delta_t$ and given δ_g, they compute the smallest ε_g such that the composition is $(\varepsilon_g, \delta_g)$-differentially private. The composition theorem of [58] is crucial to achieving optimal utility in practice (for example, it is a crucial ingredient of the PSI (Ψ) system to which the PIs contributed [43]). Thus, we believe that strongly optimal privacy odometers and filters for the adaptive parameter setting are similarly important in our application.

Goal 2. *Design strongly optimal privacy filters and odometers for the adaptive-parameter setting.*

As a first step, we will consider the setting described above where the parameters are fixed and only the number of rounds is adaptive. In this case we conjecture that one can compute the tightest possible privacy odometer using an extension of the algorithm of [58] for computing a strongly optimal composition bound in the fixed-parameter setting. The general case where the parameters can be chosen adaptively will be more delicate, since one has to consider an arbitrary process for generating the stability parameters, however we believe this is a crucial goal and any progress we can make will be valuable.

Other Stability Notions. As we discussed, differential privacy is not the only notion of distributional stability that is sufficient to prevent false discovery. There are a wealth of weaker notions that prevent false discovery such as KL-stability [13], max-information [29], concentrated differential privacy [33, 21], and typical stability [12]. These notions give weaker guarantees about generalization, but they are easier to satisfy, so *a priori* it is difficult to tell which guarantee will be tightest in each application.

Thus, we believe that the ideal approach is to measure stability with respect to *all* of these notions simultaneously in order to give the tightest bound on how much we need to correct for adaptivity. However, currently valid odometers and filters are not known for any of these notions in the adaptive setting.

Goal 3. *Construct asymptotically (or strongly) optimal privacy odometers and filters for other notions of distributional stability in the adaptive-parameter setting.*

A first step will be to consider *concentrated differential privacy* [33, 21], which is a very slight relaxation of differential privacy that yields better utility than differential privacy in practice.

A.3 Verification of Composition and Odometers

Several programming language techniques have been developed to formally guarantee differential privacy. Here we aim at extending these techniques for ensuring that a data analysis is able to generalize without false discoveries. Concretely, we want to design a type system where this guarantee corresponds to some typing judgment as

$$\Gamma \vdash t : T$$

This judgment should only be valid if t is a data analysis that meets a mathematical requirement G (expressed by T) that ensures the generalization along the lines discussed in previous sections. This approach is similar to the one that PI Gaboardi and collaborators have used in [8] where differential privacy is expressed at the type level by means of relational refinement types. Unfortunately, this approach lacks the machinery needed to support reasoning principles for strong composition schemes. These schemes are not only important for differential privacy but they are foundamental to guarantee that stable adaptive data analyses can be combined to achieve precise results.

A.3.1 Type-based Reasoning At the Level of Entire Distributions

The DRV composition [34], the optimal composition theorems [51, 58], the composition theorem for concentrated differential privacy [33, 21], the moment accountant method [1] and the pay-as-you-go composition [61] are useful tools to achieve data analysis with improved accuracy, and are crucial to achieve non-trivial guarantees for preventing false discovery. Thus, they are natural target for our work. Let us consider the DRV composition [34] in more details. Informally, the theorem states that if we run T mechanisms, $M_1(D), \ldots, M_T(D)$ on the same dataset D and each of them satisfies (ε, δ)-differential privacy, then the resulting mechanism is roughly $(\sqrt{T}\varepsilon, T\delta)$-differentially private. This claim remains true even if the choice of the mechanism M_t is allowed to depend on the output of $M_1, \ldots, M_{t-1}$.

In the recent work [6], PI Gaboardi and collaborators have devised a method for lifting the DRV composition to a relational program logics with deterministic assertions. While this approach allows using this composition in the verification of programs with great saving in the privacy parameters, it is still unsatisfying. The main problem is that this approach uses the DRV composition as a black-box. This is unsatisfactory for two reasons: first, each new composition method would require the design of a new black-box; second, a bug in any black-box that goes unnoticed may compromise the entire analysis, generating a false discovery. Hence, it would be desirable to have ways to verify the composition scheme before using them as black-box primitives.

Let's consider the proof of the DRV composition [34] (see [32] for a textbook treatment). Its structure escaped so far to program verification tools. At a high level, the proof consists of two main steps: first, providing a bound on the privacy loss in expectation of a variable C_i corresponding to the output of M_t conditioned over $M_1, \ldots, M_{t-1}$; second, providing a bound for the overall privacy loss using a concentration of measure argument. Technically, the bound on the privacy loss of each C_i is given via a reformulation of differential privacy in term of max-divergence measures and statistical distance. The overall bound is instead given by using Azuma's Inequality to provide an high concentration bound. Azuma's inequality can be stated as a bound on the probability that the sum of a given martingale sequence of real-valued random variables C_i is significantly greater than its expectation. For appropriate values a and b it can be stated as

$$\mathbb{P}\left[\sum_i C_i > a\right] < e^{-b}$$

The use of max-divergence and statistical measures between distributions and Azuma's inequality to give a high concentration bound gives interesting challenges to the verification of the DRV composition.

The verification of this proof requires mathematical properties that go beyond differential privacy such as properties of expectation, concentration of measure, union bounds, etc. as well as reasoning principles on other measures on probability distributions. The distinguishing feature of these properties is that they describe properties of distributions as a whole, in ways that go beyond what can be expressed by the individual values that random variables are allowed to assume. With a concrete example, reasoning about a fact like $\mathbb{P}[X \in V] \leq \beta$ for a discrete random variable X requires *summing* the probabilities of all the values of the support $\mathrm{supp}(X)$ that are in V and checking that this sum is less than β. These are methods that are not currently supported by type-based tools like the one developed in [8, 5]

Goal 4. *Devise type-level support for reasoning at the level of entire distributions.*

To achieve this goal, we will design a relational refinement type system where *refinement types can express properties about distributions as a whole*. For instance we will integrate in the refinement-level language constructions like $\mathbb{P}[X]$ and $E(X)$ for reasoning explicitly about distributions and expected values of random variables. These construction combined with standard refinement types will permit to write statements like $\mathbb{P}[x \geq \alpha] \leq \beta$ useful to describe accuracy guarantees and concentration bounds. A main technical challenge

here will be to design effective techniques to integrate the solving of verification conditions for refinements that express properties about distributions with SMT solvers. This will be discussed in more details in Section A.5.

A.3.2 Reasoning about Adversaries

Another aspect of the proof of the DRV composition theorem that is shared also by some other composition theorems [61, 33, 21] is that it requires to reason in specific ways about a more general class of adversaries, something that is in general not needed for basic composition scheme in differential privacy. In these proofs, the adversary is used to formally describe the *model* in which these composition theorems hold. For example the proof of the DRV composition theorem [34] formalizes the meaning of composition in terms of *experiments* and *views*. Experiments are a way to describe composition in terms of an adversary who choses the next mechanism to run. Views are a way to describe the input to the adversary.

Several verification techniques have been developed to reason about adversary in settings like cryptography [10, 17]. Following these approaches, PI Gaboardi and collaborators [6] have also introduced a way to reason about adversary in a program logic for differentially privacy. This extension enabled proving differential privacy for a query online algorithm where the choice of the query at each step can be considered adversarial. In this approach one represents the adversary as an external procedure call.

Unfortunately, this approach has some restrictions that limit its applicability for strong composition scheme, so far. First, it is unclear how to tailor the power of this adversary to express the experiments and the views of the adversaries in the proof of the DRV composition theorem [34], the filters and odometers for the pay-as-you-go composition from [61], etc. Second, the reasoning about the adversary cannot directly depend on the differential privacy parameters ϵ and δ, which is instead clearly needed in describing settings like the pay-as-you-go composition [61]. Third, it is unclear whether this approach would be effective also in a relational refinement type system where assertions express properties about entire distributions as the one discussed in the previous section.

Goal 5. *Developing relational refinement types principles for reasoning about more general adversaries in composition scheme.*

To achieve this goal we will start by looking for ways to formulate the notions of experiments, views, privacy filters and odometers, etc. in a relational type system. Ideally, we would like to formulate these notions in terms of probabilistic approximate coupling following recent work by PI Gaboardi and collaborators [6, 9]. One approach that we will explore for this goal is to reformulate probabilistic approximate coupling through the lenses of the privacy region framework by Kairouz et al. [51]. This can provide an abstract setting for specifying the adversarial requirements in refinement types.

A.3.3 Verification of Filters and Odometers Beyond Differential Privacy

As we discussed in previous sections, there are several relaxed notions of distributional stability that are useful for preventing false discovery, including KL-stability [13], max-information [29], concentrated differential privacy [33, 21], and typical stability [12]. Besides developing optimal privacy odometers for them we also plan to design methods to verify their strong composition scheme. While these stability notions are arguably quite different from differential privacy, we expect that it would be possible to extend the verification methods discussed in the previous section and useful to prove composition for filters and odometers based on differential privacy also to other notions.

Goal 6. *Developing refinement type system tools for proving composition for filters and odometers based on other stability notions.*

For some of them we can use an approach based on f-divergences that PI Gaboardi and collaborators have recently integrated in a relational refinement type system for differentially private Bayesian inference [5]. This approach permits to use the type-level assertions to reason about probabilistic measures such as KL-divergence, statistical distance, etc. A technical challenge here is how to combine the complex reasoning principles for the different divergences with the complex requirements of strong composition scheme in a way that does not blow up refinement types.

A.4 Algorithmic Support

In addition to building a prototype system using verification for supporting the design of adaptive data analysis, our project will also expand the toolkit of algorithms available for common use cases in adaptive data analysis. Below we outline the main algorithmic tools that we believe our system should have that are not currently available, although we emphasize that this list is not exhaustive.

A.4.1 Stable Algorithms for Many Queries

We discussed how simple Gaussian noise can be used to answer t statistical queries with error $\approx (\frac{\sqrt{t}}{n})^{1/2}$, yielding non-trivial accuracy for nearly n^2 such queries. However, there are much more sophisticated algorithms that answer t queries with error $\approx (\frac{\sqrt{d}\log(t)}{n})^{1/3}$ where d is the dimension of the data (i.e. the support of P is $\{\pm 1\}^d$) [62, 48, 45, 70]. This bound nearly matches the guarantees achievable for static data analysis, and means that we can answer exponentially many queries accurately!

However, all algorithms approaching this level of utility are computationally infeasible, as their running time is at least 2^d per query. Thus, a central challenge is to design *computationally efficient* stable algorithms that accurately answer more than n^2 queries. Unfortunately, we know from the work of PI Ullman [71, 69, 55] that any stable algorithm answering more than n^2 queries must have exponential worst-case running time.

Not only do these algorithms have exponential worst-case running time, they actually have exponential running time on every input. The bottleneck is that the algorithm explicitly manipulates a probability distribution over the exponentially large set $\{\pm 1\}^d$. Experimental results [47] show that these algorithms indeed achieve very good utility, but become infeasible for $d > 32$.[1]

Recently, PI Gaboardi and collaborators [41] designed the differentially private *DualQuery* algorithm that achieves essentially optimal accuracy, and is efficient in practice on datasets with $d \approx 10^4$ despite (necessarily) having exponential worst-case running time. The algorithm makes use of the powerful *CPLEX* tool for solving NP-hard optimization problems (specifically integer programming) in such a way that stability and statistical accuracy are guaranteed, and the running time depends only on the heuristic performance of *CPLEX*. Unfortunately, *DualQuery* requires access to the entire set of queries in a way that seems rather inherent, so it is unsuitable for adaptive data analysis.

A main goal of our project is to develop an analogue of *DualQuery* for adaptive data analysis. We remark that such an algorithm would also significantly advance the state-of-the-art in differential privacy.

Goal 7. *Design a stable algorithm that is guaranteed to be accurate for adaptively chosen queries, and achieves good running time in practice on high-dimensional datasets (d between 10^2 and 10^4).*

The algorithms we will produce will be empirically evaluated in different ways. First, we will compare our algorithm to *DualQuery* for the non-adaptive setting. Although we are solving a harder problem, we want to establish that our algorithm is competitive with *DualQuery* even for non-adaptive data analysis while also supporting adaptive data analysis. Second, we will evaluate our algorithm on various *variable/model selection*

[1] The experimental results of [47] are in a setting where the queries are fixed, but the underlying algorithm generalizes seamlessly to the adaptive setting.

tasks, which is a fundamental setting for privacy and for adaptive data analysis in the Statistics literature. Specifically, we will compare our algorithm to both independent noise (a very general technique) and to specialized statistical methods for inference after model selection [16, 40, 68] (a very specific technique). More details about the evaluation of the algorithms will be given in Section A.5.1.

Technical Approach. Our initial approach is based on the private multiplicative weights (PMW) algorithm. This algorithm was introduced for statistical queries by Hardt and Rothblum [48] and was subsequently generalized [45] and then extended to convex minimization queries by PI Ullman [70]. The algorithm was also shown to be achieve good utility and running time for low-dimensional datasets [47], and achieves optimal accuracy for a wide variety of statistical applications [23, 66], making it an appealing starting point.

As we discussed, this algorithm needs to explicitly maintain a distribution D over the (huge) domain $\{\pm 1\}^d$, which is a computational bottleneck. However, all the algorithm needs to do is to estimate the expectation of certain bounded functions $\psi(D) = \mathbb{E}_{z \sim D}[\psi(z)]$, which we can hopefully do without storing D explicitly. A natural way to compute the expectation is to take *samples* from the distribution D using *Markov Chain Monte Carlo (MCMC)*. At a high-level, the idea is to perform a random walk over the domain $\mathcal{D}$ where the stationary distribution of the random walk is D. The hope is that after a small number of steps of the walk, we will arrive at an element x that is distributed as a random sample from D.

MCMC is by far the most common tool for computing expectations over complex distributions. It can be shown to be efficient and accurate for many families of distributions over exponentially large domains and often has good practical performance on natural instances of NP-hard problems like sampling from the posterior distributions of many Bayesian models. Although PMW can produce distributions D for which the random walk of MCMC does not rapidly mix, we believe that these distributions are unlikely to arise in practice the same way many NP-hard problems can be solved efficiently in practice using tools like CPLEX, belief propagation, and MCMC. Moreover, the distributions generated by multiplicative weights are *Gibbs distributions* (also known as *maximum entropy distributions*), which are exactly the sorts of distributions that arise in statistics and machine learning applications where MCMC is extremely popular.

An important observation, based on the work of PI Ullman and collaborators [45], is that even if we use a heuristic approach to computing the expectation $\psi(D)$, stability is still guaranteed, and only accuracy or running time will depend on the performance of the heuristic. Thus, we will still have sound, conservative guarantees on the degree of overfitting due to adaptivity.

A.4.2 Improving on Independent Noise

Looking more closely at independent additive noise, we see that additive noise can be used to answer t insensitive queries given a sample of size n up to error $c(\frac{\sqrt{t \log(t)}}{n})^{1/2}$ for some small absolute constant $c > 0$. Although it may seem like a small issue from a theoretical perspective, the $\sqrt{\log(t)}$ factor is quite significant in practice. Roughly, that factor comes from the fact that independent Gaussian noise will occasionally take extreme values, and those extreme values are precisely where false discoveries can occur.

Our project will explore more accurate algorithms that eliminate the additional logarithmic error. In a preliminary work of PI Ullman and Thomas Steinke [66], we showed that removing this factor requires *correlated noise*, and also gave an algorithm that achieves improved asymptotic error $c'(\frac{\sqrt{t \log \log(t)}}{n})^{1/2}$. Unfortunately, the constant c' is too large for this algorithm to be useful in practice. We conjecture that there is an algorithm with error $c''(\frac{\sqrt{t}}{n})^{1/2}$ for c'' small enough to be useful in practice.

One bottleneck in our analysis is the use of the *exponential mechanism* [56] for stable selection from a discrete set of objects. In general the exponential mechanism incurs an $O(\log(s))$ overhead when selecting from s objects. However, we suspect that in our application, where we only need to select the largest from s draws of a Gaussian, it incurs error $O(1)$. This suspicion is based on the tails of the closely related *log-normal*

distribution. Rigorously establishing our suspicion is likely to be enough to obtain a new algorithm, maybe even establishing our conjecture.

A.5 Prototype Implementation and Evaluation

To make more concrete the developments we discussed in the previous sections and to have a concrete way to evaluate our progresses, we will design and implement a prototype, named *TrueDisc*, to support the design of stable algorithms useful to prevent false discoveries. This prototype will be designed following the *reusable holdhout* model [28] where an arbitrary data analysis can be trained on some training data but it can only access the validation data through some mechanism that guarantees stability. This model with the components of the prototype is described in Figure 1. We will aim at developing this prototype as a first proof of concept and for individual uses but we can imagine that in further development such a prototype could be a component of a larger system in a data curator model where the data are provided by an external mechanism and the prototype is in charge of mediating access to it.

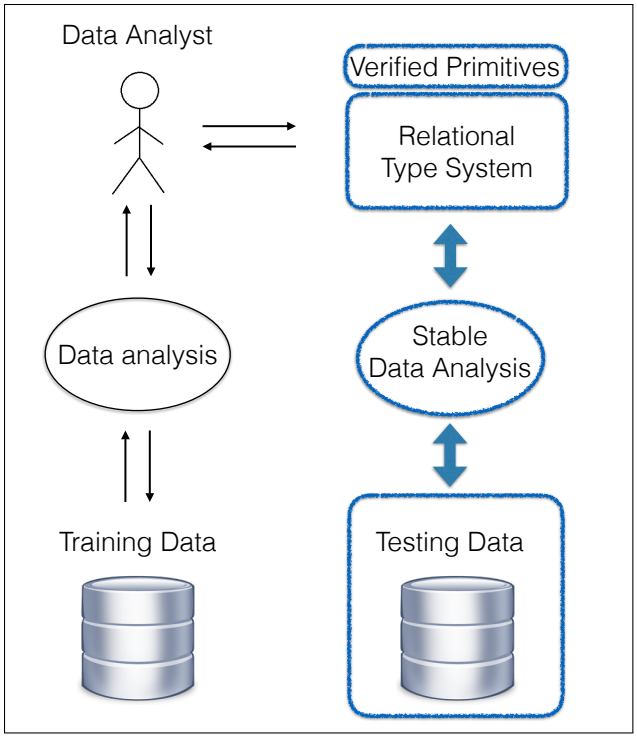

Figure 1: Workflow of *TrueDisc*. Following the *reusable holdout* approach, the data analyst has direct access to the training data while her access to the testing data is instead mediated by the components of *TrueDisc*, marked in blue. The use of *TrueDisc* creates a barrier guarantee stability and protection against false discoveries.

To achieve this workflow, *TrueDisc* will implement a type system along the lines of the one described in Section A.3. This type system will be carefully designed to achieve two different tasks:

A Providing to the data analyst a tool for ensuring the stability of her data analysis by using some trusted primitives/components,

B Providing to the data analyst a tool for verifying the stability of newly designed components.

The separation of these two tasks reflects two different uses of program verification that we want to integrate in our framework. On the one hand, we want to have a system that is easily usable by a user that doesn't necessarily have expertise in software verification, on the other hand we want a system that can also support the design of complex algorithms (e.g. strong composition scheme) by an expert in software verification. Specifically, our approach combines two traditions that have been followed in the design of tools for differential privacy. On the one hand, we will use a type system to provide a formal guarantee that a data analysis built out of some primitives is stable similarly to the approach that has been followed for differential privacy in works by Reed and Pierce [59] and by the PI Gaboardi and collaborators [42, 5]. On the other hand, we will use a type system to provide fully-fledged verification of stability similarly to the approach that has been followed for differential privacy in works by Barthe et al. [11] and more recently by the PI Gaboardi and collaborators [9, 6].

One of the challenges here will be to carefully design the type checking procedure. Indeed, more complex reasoning principles we will add at the refinement type level and more difficult will be for an SMT solver

to solve the verification conditions in an automated way. As an example, we cannot hope that the Azuma's inequality can be solved automatically by the solver. For this reason, we will carefully choose the principles to integrate in the refinement types language and we will design domain-specific macros that can be used to address particular task such as discharging verification conditions with concentration bounds. These macros can be provided by the programmer, making the verification semi-automated.

Goal 8. *Design an expressive type-checking procedure that allows to discharge most of the verification conditions in an automated or semi-automated way.*

It is important to stress that the complexity of the type checker must not affect the overall usability of the *TrueDisc*. Indeed, we expect that complex verification conditions will be needed only in the verification of primitives/components implementing the different strong composition schemes. We envision to integrate as built-in primitives the implementations of the algorithms that will be the outcome of the work discussed in Section A.4. These algorithm will be implemented, verified and optimized to provide good efficiency.

A.5.1 Prototype Evaluation

The validation of the results obtained in the project wiIl go through an experimental evaluation of the prototype. We do not expect a throughout evaluation of every aspect of the prototype and we will instead focus on two aspects:

1. the expressivity and usefulness of the verification tools.
2. the efficienty and accuracy of the primitives implementing the algorithms we design.

To evaluate 1) we will implement and verify different algorithms from the literature. We will mainly focus on two classes of algorithm: algorithms implementing strong composition scheme, and algorithms that have already been verified but only using composition as a black box or without considering adaptivity. As algorithms implementing strong composition scheme, of course we will start with the DRV composition theorem [34], but we will then focus on the optimal composition theorems [51, 58], and on the pay-as-you-go composition from the PI Ullman and collaborators [61]. Concerning instead the algorithms that have already been verified we will focus on providing new verification for the MWEM algorithm and the *DualQuery* algorithm previously verified by the PI Gaboardi and collaborators using only basic composition [8], and the Sparse Vector Between Threshold algorithm by the PI Ullman and collaborators [22], that PI Gaboardi and collaborators have recently verified [6] using composition as black box. Moreover, we will consider the recent notion of concentrated differential privacy [33, 21] and the moment generating accountant composition from [1].

To evaluate 2) we will perform two classes of experiments. First, similarly to the work by Dwork et al. [28] we will compare the use of our algorithms on synthetic data, for which we can control the ground truth, with baselines algorithms that do not guarantee generalization. We expect this class of experiments to help us to discriminate the kinds of analysis on which our algorithms are particularly useful. Second, similarly to the work by the PI Gaboardi and collaborators [41], we will use a combination of synthetic data and real-world data to compare our approach to other algorithms from the literature. This class of experiments will help us identifying the kind of data on which our algorithm is more efficient or more accurate than others.

One challenge in evaluating a system like this is that adaptive data analysis arises in a very wide range of use cases. We will evaluate first on "easy" cases like inference-after-model-select to make sure that we are competitive with tasks that are tailored for common use cases. Then we will evaluate on "hard" cases like the adversarial use cases described in the literature [49, 30, 66, 18]. Thus, we will sandwich the utility of our system in between these two extremes. By making our system available to researchers, we also hope to gather feedback on intermediate use case that arise in practice.

In all these experiments we will choose datasets with increasingly more attributes and increasingly more samples to understand where is the limit of our approach. We will also use different metrics to measure the accuracy and the efficiency of our analyses. The real-world data will come from the Harvard `Dataverse` depository, a data repository infrastructure that the PIs have already used in a recent collaboration [43].

A.6 Timeline and Collaboration Plan

This project will involve collaboration across institutions (University at Buffalo and Northeastern University) as well as research areas (programming languages and algorithms). We have worked out the following plan to ensure smooth collaboration and facilitate interaction between the PIs as well as the Ph.D. students that will be funded by this project. We remark that the PIs have a history of successful interdisciplinary collaboration with one another as part of Harvard University's Privacy Tools group.

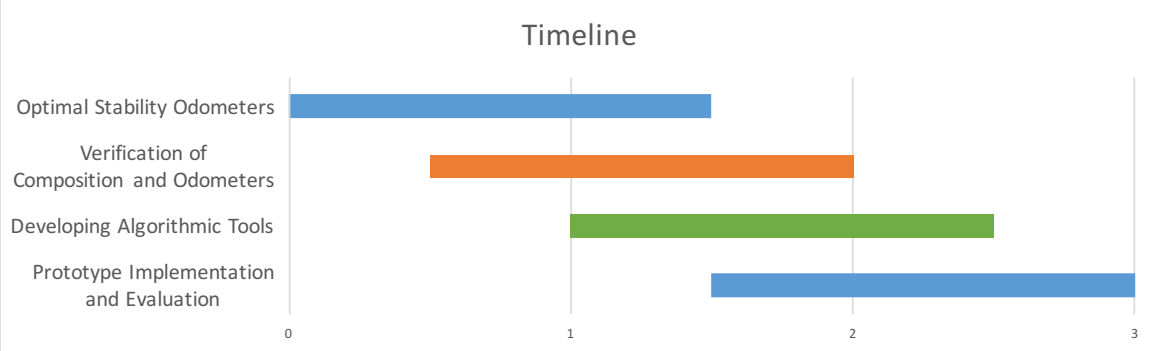

Figure 2: Approximate timeline showing the various phases of the project.

Figure 2 shows an approximate timeline of the different phases of the project and how they overlap.

- **Year 1:** We will begin work on optimal odometers, which are a must-have for our system. PI Ullman will lead this phase of the work. After the initial algorithmic work on this topic, the verification and algorithmic aspects will proceed together. PI Ullman will fund a student from the beginning to work on these algorithmic aspects of the project.
- **Year 2:** We will continue the verification of optimal odometers, while also beginning the development of new algorithmic tools. In the second half of the year, with the odometers work completed, we will begin implementing the prototype system. PI Gaboardi will begin funding a student in this year to further progress on the verification work and begin the prototype implementation work.
- **Year 3:** We will focus on the completion, evaluation, and dissemination of the prototype system, while also continuing to refine our algorithmic tools as the system's needs dictate.

PI Ullman will support a student in Years 1-3 to work on the algorithmic aspects of the project. PI Gaboardi will support a student in Years 2-3 who will be focused on the programming languages and implementation aspects of the project. We expect students as well as the PIs to be engaged in all aspects of the project, especially the points of close intersection between the two lines of work. PI Ullman's student will be expected to help interface between the construction and verification of stability odometers, and both students will help interface between the design, implementation, and evaluation of the algorithmic tools.

We will facilitate these collaborations using biweekly project meetings conducted over Skype. These meetings will be used to keep all members engaged, share progress, and adjust goals. There will also be biannual in-person project meetings taking place in Boston and Buffalo. We will also use the requested travel funds to have project members spend time at the other members' institution for in-person collaboration.

B Broader Impacts of the Proposed Work

Given the increased reliance on machine learning and empirical science in all aspects of life ranging from policy to education to healthcare, we believe this project will have a broad and long-lasting impact. A study in *Nature* has estimated the cost of non-reproducible research to be $28 billion in the biological sciences alone [2], and researchers have identified issues of adaptivity in data analysis as a contributing factor to this problem [44]. While there is no magic bullet for solving this problem, our algorithmic and PL research will advance the state-of-the-art in the foundations of adaptive data analysis. Our pilot system will be a first step towards building tools that empirical scientists can use to address the problem, and we believe it can serve as a bridge to empirical scientists, so that computer scientists can better understand the needs of these scientists.

In addition to disseminating our scientific results broadly, these results will converge in a preliminary system that we will test on realistic data. Our code, as well as documents, research papers, experimental results, etc., will be made available to developers and the public on our project web-site. Although this will be a pilot system, intended for research purposes only, we believe it will be an important step towards gathering the necessary feedback and exposure to build a more complete system in future work. We will further the impact of our project by training students in interdisciplinary research. These students will be exposed to an atypical combination of algorithmic and programming languages techniques, and will learn to think critically about the cross-cutting problem of false discovery. Finally, we will broaden our impact of our project by continuing to grow and promote the Theory and Practice of Differential Privacy (TPDP) Workshop.

Student Involvement in Research. The two Ph.D. students on this project will have different goals, depending on whether they are working primarily on the algorithmic aspects of the project or the programming languages aspects of the project. However, a key component of their work and education through this project will be to learn interdisciplinary research skills. The PIs have extensive experience doing cross-cutting research (in large part due to the highly collaborative nature of Harvard University's Privacy Tools group they both collaborate with) that they will pass on to the students. For example, PI Ullman's student will be closely involved in the analytic work on privacy odometers and PI Gaboardi's student will be closely involved in the work on verification of privacy odometers. Although each student will have a main area of focus and expertise, these two lines of work will necessarily overlap—PI Gaboardi's student will need to learn the necessary probabilistic and algorithmic arguments underlying the odometers, and PI Ullman's student will need to learn the necessary background on verification in order to help abstract and modify the proofs to be suitable for state-of-the-art verification methods. This style of research is exemplified in PI Gaboardi et al.'s recent work on verifying differentially private algorithms via probabilistic coupling [6], in which many of the standard proofs in differential privacy were modified in order to make them suitable for verification. Taking this truly cross-cutting approach will prepare these students well for the increasingly collaborative and cross-cutting nature of CS research.

Curriculum Development Activities. PI Gaboardi will teach a graduate level topics course on differential privacy at the University at Buffalo in Spring 2017. The course will consist of readings on advanced topics in differential privacy and applications from the programming language, algorithm, machine learning, and system perspective. Some of the classes will focus on the application of differential privacy to prevent false discoveries. PI Gaboardi taught a similar class in Spring 2016. In future editions of this class more material about generalization, stability and preventing false discoveries will be presented. Moreover, PI Gaboardi will teach a graduate level course on type-based verification at the University at Buffalo starting in Fall 2017. The course will consist of lectures on advanced topics in type systems for verification with a particular focus on tools and methods for verifying randomized algorithms and differential privacy applications.

PI Ullman plans to develop a regular course on privacy-preserving data analysis at Northeastern, which

will build off of an expand the course in differentially private data analysis that he co-taught with Salil Vadhan at Harvard in Spring 2013. This course will include a significant component about statistics, false discovery, and the connection with differential privacy—which is already natural when discussing privacy-preserving data analysis. This will complement the traditional presentation of privacy as a *constraint*, with the setting of interactive data analysis where privacy serves an important role in preventing false discovery. This course also bridges a wide variety of topics in machine learning, statistics, theoretical computer science, security and to a lesser extent law, which will help expose the issue of false discovery to a much wider audience than would typically think about this problem. This course is slated to be taught as a PhD level topics course in Spring 2017 and will transition into a regular course offering roughly once every two years.

Creating research synergies. PIs Gaboardi and Ullman are heavily involved in the organization of a new workshop "Theory and Practice of Differential Privacy" (TPDP). This workshop brings together researchers from different fields working on formal approaches to privacy research and applications of privacy research (such as false discovery). The first workshop was held in April 2015 associated with ETAPS. PI Gaboardi was an organizer and chair, and PI Ullman provided the keynote talk. The second workshop was held in June 2016 associated with ICML, and had nearly 120 participants. The third installation will is chaired by PI Ullman and will be hosted at Northeastern in June 2017. This workshop will help disseminate the outputs of this proposal, and in the spirit of this proposal we expect that it will be colocated with statistics or machine learning venues roughly every other year and colocated with programming languages, systems, or databases venues in alternating years. The PIs are also planning to organize a workshop on interdisciplinary approaches to adaptive data analysis during this project, focused on tools in addition to statistical methods.

C Results from Prior NSF Support

PI Gaboardi's current support consists of *TWC: Large: Collaborative: Computing over Distributed Sensitive Data:* Award #1565365. PI Gaboardi is supported for the period May 1st 2016 to April 30th 2020 with a $527,869 share of the budget. The project is joint with Stephen Chong, Salil Vadhan, Kobbi Nissim, and James Honaker at Harvard University, and Or Sheffet at Ottawa University.

Intellectual Merit: The goal of this project is to develop a collection of tools based on differential privacy for sharing the results of distributed computations without sharing the sensitive data. These goals are quite different from the one presented in the current proposal: preventing false discoveries in adaptive data analysis. The technical problems and the techniques proposed are definitely different: developing techniques for distributed differential privacy, verification tools for the R programming language, and techniques based on remote attestation. The most close relation between this project and the current proposal is that both of them plan to use data sets from Harvard Dataverse repository infrastructure to validate experimentally the tools developed. We believe that this is not a fundamental overlapping.

Broader Impact: The project started on May 1st 2016 and the main impact so far has been through published works.

Publications: The current results include the following publications [6, 5].

PI Ullman is a beginning investigator and has not received any NSF grants. PI Ullman has a pending proposal *SaTC: Medium: Collaborative: Blending Differential Privacy and Secure Computation*. PI Ullman's share of the requested budget is $300,000. The goal of this proposal is to combine techniques from secure computation and differential privacy to analyze sensitive data in a distributed setting. The goals of this pending proposal are very different from the current proposal because the 1) the pending proposal is focused on privacy as an end in itself, rather than privacy as a tool for preventing false discovery and 2) the pending proposal is focused on distributed data, which is not an issue in the current proposal.

References

[1] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 308–318, 2016.

[2] M. Baker. Irreproducible biology research costs put at $28 billion per year. *Nature*, 2015.

[3] A. Balsubramani. Sharp finite-time iterated-logarithm martingale concentration. *arXiv preprint arXiv:1405.2639*, 2014.

[4] G. Barthe, G. Danezis, B. Grégoire, C. Kunz, and S. Zanella-Béguelin. Verified computational differential privacy with applications to smart metering. In *IEEE CSF 2013*, 2013.

[5] G. Barthe, G. P. Farina, M. Gaboardi, E. J. G. Arias, A. Gordon, J. Hsu, and P. Strub. Differentially private bayesian programming. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 68–79, 2016.

[6] G. Barthe, N. Fong, M. Gaboardi, B. Grégoire, J. Hsu, and P. Strub. Advanced probabilistic couplings for differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 55–67, 2016.

[7] G. Barthe, M. Gaboardi, E. J. G. Arias, J. Hsu, C. Kunz, and P. Strub. Proving differential privacy in hoare logic. In *IEEE 27th Computer Security Foundations Symposium, CSF 2014*, pages 411–424, 2014.

[8] G. Barthe, M. Gaboardi, E. J. Gallego Arias, J. Hsu, A. Roth, and P.-Y. Strub. Higher-order approximate relational refinement types for mechanism design and differential privacy. In *The 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '15*. ACM, 2015.

[9] G. Barthe, M. Gaboardi, B. Grégoire, J. Hsu, and P. Strub. Proving differential privacy via probabilistic couplings. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 749–758, 2016.

[10] G. Barthe, B. Grégoire, and S. Z. Béguelin. Formal certification of code-based cryptographic proofs. In *Proceedings of the 36th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2009, Savannah, GA, USA, January 21-23, 2009*, pages 90–101, 2009.

[11] G. Barthe, B. Köpf, F. Olmedo, and S. Z. Béguelin. Probabilistic relational reasoning for differential privacy. In *Proceedings of the 39th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2012*, pages 97–110, 2012.

[12] R. Bassily and Y. Freund. Typicality-based stability and privacy. *CoRR*, abs/1604.03336, 2016.

[13] R. Bassily, K. Nissim, A. Smith, T. Steinke, U. Stemmer, and J. Ullman. Algorithmic stability for adaptive data analysis. *CoRR*, abs/1503.04843, 2015.

[14] R. Bassily, K. Nissim, A. D. Smith, T. Steinke, U. Stemmer, and J. Ullman. Algorithmic stability for adaptive data analysis. In *Proceedings of the Forty-Eighth Annual ACM on Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA*, 2016.

[15] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[16] R. Berk, L. Brown, A. Buja, K. Zhang, L. Zhao, et al. Valid post-selection inference. *The Annals of Statistics*, 41(2):802–837, 2013.

[17] K. Bhargavan, C. Fournet, and A. D. Gordon. Modular verification of security protocol code by typing. In *Proceedings of the 37th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2010, Madrid, Spain, January 17-23, 2010*, pages 445–456, 2010.

[18] A. Blum and M. Hardt. The ladder: A reliable leaderboard for machine learning competitions. *CoRR*, abs/1502.04585, 2015.

[19] C. E. Bonferroni. Teoria statistica delle classi e calcolo delle probabilita. *Pubbl. d. R. Ist. Super. di Sci. Econom. e Commerciali di Firenze.*, 8, 1936.

[20] O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[21] M. Bun and T. Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. *arXiv preprint arXiv:1605.02065*, 2016.

[22] M. Bun, T. Steinke, and J. Ullman. Make up your mind: The price of online queries in differential privacy. *CoRR*, abs/1604.04618, 2016.

[23] M. Bun, J. Ullman, and S. P. Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *STOC*, pages 1–10. ACM, May 31 – June 3 2014.

[24] L. D'Antoni, M. Gaboardi, E. J. G. Arias, A. Haeberlen, and B. C. Pierce. Sensitivity analysis using type-based constraints. In *FPCDSL'13*, 2013.

[25] L. Devroye and T. J. Wagner. Distribution-free inequalities for the deleted and holdout error estimates. *IEEE Transactions on Information Theory*, 25(2):202–207, 1979.

[26] L. Devroye and T. J. Wagner. Distribution-free performance bounds for potential function rules. *IEEE Transactions on Information Theory*, 25(5):601–604, 1979.

[27] O. J. Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56:52–64, 1961.

[28] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. Generalization in adaptive data analysis and holdout reuse. In *Advances in Neural Information Processing Systems (NIPS)*, Montreal, December 2015.

[29] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. Generalization in adaptive data analysis and holdout reuse. In *Advances in Neural Information Processing Systems (NIPS)*, Montreal, December 2015.

[30] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. Preserving statistical validity in adaptive data analysis. In *STOC*. ACM, June 14–17 2015.

[31] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284. Springer, March 4-7 2006.

[32] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[33] C. Dwork and G. N. Rothblum. Concentrated differential priacy. *arXiv preprint arXiv:1603.01887*, 2016.

[34] C. Dwork, G. N. Rothblum, and S. P. Vadhan. Boosting and differential privacy. In *FOCS*, pages 51–60. IEEE, Oct 23–26 2010.

[35] H. Ebadi and D. Sands. Featherweight pinq. *arXiv preprint arXiv:1505.02642*, 2015.

[36] H. Ebadi, D. Sands, and G. Schneider. Differential privacy: Now it's getting personal. In *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 69–81, 2015.

[37] Economist. Trouble at the Lab. *The Economist*, 19 October 2011.

[38] F. Eigner and M. Maffei. Differential privacy by typing in security protocolss. In *IEEE CSF 2013*, 2013.

[39] S. Elder. Challenges in bayesian adaptive data analysis. *CoRR*, abs/1604.02492, 2016.

[40] W. Fithian, D. Sun, and J. Taylor. Optimal inference after model selection. *arXiv preprint arXiv:1410.2597*, 2014.

[41] M. Gaboardi, E. J. G. Arias, J. Hsu, A. Roth, and Z. S. Wu. Dual query: Practical private query release for high dimensional data. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1170–1178, 2014.

[42] M. Gaboardi, A. Haeberlen, J. Hsu, A. Narayan, and B. C. Pierce. Linear dependent types for differential privacy. In *The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '13*, pages 357–370, 2013.

[43] M. Gaboardi, J. Honaker, G. King, J. Murtagh, K. Nissim, J. Ullman, and S. P. Vadhan. PSI (Ψ): a private data sharing interface. *CoRR*, abs/1609.04340, 2016.

[44] A. Gelman and E. Loken. The statistical crisis in science. *Am Sci*, 102(6):460, 2014.

[45] A. Gupta, A. Roth, and J. Ullman. Iterative constructions and private data release. In *Theory of Cryptography Conference*, pages 339–356. Springer, 2012.

[46] A. Haeberlen, B. C. Pierce, and A. Narayan. Differential privacy under fire. In *Proceedings of the 20th USENIX Security Symposium*, Aug. 2011.

[47] M. Hardt, K. Ligett, and F. McSherry. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 2348–2356, 2012.

[48] M. Hardt and G. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Proc. 51st Foundations of Computer Science (FOCS)*, pages 61–70. IEEE, 2010.

[49] M. Hardt and J. Ullman. Preventing false discovery in interactive data analysis is hard. In *FOCS*. IEEE, October 19-21 2014.

[50] J. P. A. Ioannidis. Why most published research findings are false? *PLoS Medicine*, 2(8):124, August 2005.

[51] P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1376–1385, 2015.

[52] M. J. Kearns. Efficient noise-tolerant learning from statistical queries. In *STOC*, pages 392–401. ACM, May 16-18 1993.

[53] M. J. Kearns and D. Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural Computation*, 11(6):1427–1453, 1999.

[54] A. Khintchine. über einen satz der wahrscheinlichkeitsrechnung. *Fundamenta Mathematicae*, 6(1):9–20, 1924.

[55] L. Kowalczyk, T. Malkin, J. Ullman, and M. Zhandry. Strong hardness of privacy from weak traitor tracing. In *Manuscript*, 2016.

[56] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103. IEEE, Oct 20–23 2007.

[57] F. D. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *SIGMOD*. ACM, 2009.

[58] J. Murtagh and S. P. Vadhan. The complexity of computing the optimal composition of differential privacy. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pages 157–175, 2016.

[59] J. Reed and B. C. Pierce. Distance makes the types grow stronger: A calculus for differential privacy. In *ICFP'10*. ACM, 2010.

[60] R. Rogers, A. Roth, A. Smith, and O. Thakkar. Max-information, differential privacy, and post-selection hypothesis testing. *arXiv preprint arXiv:1604.03924*, 2016.

[61] R. Rogers, A. Roth, J. Ullman, and S. Vadhan. Privacy odometers and filters: Pay-as-you-go composition. In *NIPS*, 2016.

[62] A. Roth and T. Roughgarden. Interactive privacy via the median mechanism. In *STOC*, pages 765–774. ACM, June 5–8 2010.

[63] I. Roy, S. T. V. Setty, A. Kilzer, V. Shmatikov, and E. Witchel. Airavat: security and privacy for MapReduce. In *Proc. NSDI*, 2010.

[64] D. Russo and J. Zou. Controlling bias in adaptive data analysis using information theory. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 1232–1240, 2016.

[65] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11:2635–2670, 2010.

[66] T. Steinke and J. Ullman. Between pure and approximate differential privacy. *CoRR*, abs/1501.06095, 2015.

[67] T. Steinke and J. Ullman. Interactive fingerprinting codes and the hardness of preventing false discovery. In *Proceedings of The 28th Conference on Learning Theory (COLT'15)*, pages 1588–1628, 2015.

[68] J. Taylor and R. J. Tibshirani. Statistical learning and selective inference. *Proceedings of the National Academy of Sciences*, 112(25):7629–7634, 2015.

[69] J. Ullman. Answering $n^{2+o(1)}$ counting queries with differential privacy is hard. In *STOC*, pages 361–370. ACM, June 1-4 2013.

[70] J. Ullman. Private multiplicative weights beyond linear queries. In *PODS*. ACM, May 31–June 4 2015.

[71] J. Ullman and S. P. Vadhan. PCPs and the hardness of generating private synthetic data. In *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, pages 400–416, 2011.

[72] Y.-X. Wang, J. Lei, and S. E. Fienberg. A minimax theory for adaptive data analysis. *CoRR*, abs/1602.04287, 2016.

[73] D. Zhang and D. Kifer. Lightdp: Towards automating differential privacy proofs. In *The 44th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '17*, 2016. to appear.