

Differentially Private Data Privatization of College Campus

Mobility Data

Jonah Langlieb

Spring 2020

Contents

1	Introduction	3
2	Background	5
2.1	Network Usage is Changing	5
2.2	Smarter Resource Planning Can Help	6
2.3	Privacy	7
2.4	De-identification Techniques: k -Anonymity	9
2.5	De-identification Techniques: Variants on k -Anonymity	11
2.6	Differential Privacy	12
2.6.1	Definition	12
2.6.2	Differential Privacy: Basic Building Blocks	13
2.6.3	Useful Properties of Differential Privacy	14
2.6.4	Differentially Private Algorithms	15
2.6.5	Comparing Differentially Private Algorithms	16
3	Results and Discussion	17
3.1	Our Data	17
3.2	Data Pipeline; Building Models	17
3.3	Count Queries Comparison Pipeline	18
3.4	Count Queries Comparison Results	20
3.4.1	Service Time	20
3.4.2	Session Start Time	21

3.4.3	Understanding Workload-Aware Trade-offs	24
3.4.4	Count Queries Discussion	24
3.5	Transitioning To Trajectory Queries	29
3.6	GIS Data Integration	29
3.6.1	Transition Probability Matrix	31
3.7	Privatizing Trajectory Queries	31
3.7.1	NGRAM: Variable-Length N-Grams Trajectory Privatization	33
3.8	NGRAM Trajectory Query Results	34
3.8.1	Trajectory Query Discussion	38
4	Conclusion	38
5	Acknowledgments	40

1 Introduction

Over the past few decades, the Internet has developed into an increasingly integral part of everyday life and, therefore, needs to be fast, reliable, and accessible. The difficulty of these requirements is exacerbated both by the exponential rise in usage, from average number of devices to total traffic [10], as well as by the mobile nature of modern, wireless Internet access, which makes usage patterns less predictable across time and place. In order for computer science researchers and network engineers to achieve these difficult requirements, they need to be well-informed about the modern Internet’s “real-world” usage. To give one example of the rapidly evolving nature of Internet usage, when Dartmouth College set up wireless internet in 2001 (they were one of the first colleges to do so), they debated installing it in dorms, as the existing wired ports were thought to be sufficient. They ultimately found, after installing it in dorms and observing real-world usage, that residential wireless usage actually represented the *bulk* of all usage [23, p. 12].

Accurate, real-time data is the requisite backbone for network engineers and researchers to design new

network topologies, plan resource allocation strategies, and to model new network protocols. But such real-world network data is inherently personal so unless proper precautions are taken, publishing this data publicly can be a profound privacy breach for the participants whose data makes up the dataset. Network usage information can reveal participants' visit locations, usage patterns, the number of owned devices, and even identify co-travelers.

The balancing of these competing goals: **utility** – giving more insight for network researchers and, by extension, innovation for the Internet-using public – and **privacy** – preserving privacy for the data-set's participants' – leads to unavoidable trade-offs. Trying to understand and manage this trade-off is the subject of the rest of this paper.

Our goal is to take private network usage data and create a publicly-releasable 'privatized' version that preserves participant privacy, while still maintaining reasonable accuracy/utility. While difficult, the advent of the 'Big Data Era', in which more and more personal data is being collected by governments and corporations, has led to a lot of legislative recognition on, and academic research in, balancing these types of utility-privacy concerns. For a while, the privatization field focused on using *aggregation* of private attributes of a dataset as the main privatization mechanism. But this turns out to give insufficient privacy (discussed later). The modern cutting-edge mechanism, which we will be using and discussing throughout this paper, is called **differential privacy (DP)**. Colloquially, the main idea of differential privacy is to create a privatized dataset by adding enough "noise" to the ground-truth dataset so that a precise statistical condition for "privacy" is met. In particular, we say a dataset is "private" if *any* one person's contribution to *any* query is masked within provable statistical insignificance. We consider this to be a good definition of privacy because *everyone's* information is protected (so that no one would mind being part of the released dataset) but it allows us to glean information about the information *in aggregate*, which is exactly the property we want: to preserve utility while maintaining privacy. Additionally, a nice property of differential privacy, compared to previous privatization mechanisms, is that it gives us a tunable parameter, ϵ , which allows us to mathematically guarantee the relative stringency of the privacy-accuracy trade-off.

In this paper, we want to apply differentially private algorithms to a real-world network dataset in a way which provides high utility and quantifiable privacy guarantees. In particular, we focus on fine-grained user mobility data collected from **Wi-Fi access points (APs)** which we think is uniquely situated to revolutionize the development and evaluation of mobile network research. We identified two representative classes of queries, which we think will be especially useful for networks research, and so we want to focus on optimizing:

- (1) **Histogram (Count) Queries**: the numerical results of such queries are useful for network resource

dimensioning, especially for anticipating congested network APs and (2) **Trajectory Queries**: the characterization of individual/aggregate mobile trajectories are useful for understanding how the network is used at different times/locations. Our dataset is an 802.11 Wi-Fi network trace of a large-scale campus network with more than 3,000 access points and nearly 40,000 daily users [39]. Building on existing DP research, we evaluated a range of state-of-the-art DP algorithms: for count queries (MWEM [19]; GreedyH [27]; H_b, DAWA [36], Identity [16], Uniform (see [20])), aided by the open-source DP framework, ϵ ktelo [44]. And for trajectory queries, we evaluated a prefix-tree based and N-gram based model [8]. Especially for trajectory queries, this algorithm did not give sufficient accuracy vs privacy, so our work focused on (1) understanding the performance of existing DP algorithms on our data set and (2) understanding the ground-truth trajectory data itself to gain possible insights for algorithmic improvements.

The organization of the paper is as follows: Section 2 gives background on privacy, from the history, to previous privatization techniques, to the privacy paradigm we focus on in this paper, differential privacy. We give an overview of the main differential privacy definitions, mechanisms, and algorithms. Section 3 gives the results of our work with two types of queries: count queries and trajectory queries, and discusses the relative accuracy of differential private algorithms on both sets of queries. Section 4 gives the overall conclusion and future directions.

2 Background

2.1 Network Usage is Changing

It is well-known but still awe-inspiring how explosive Internet usage has been, across any recent time-scale. Each year Cisco releases a networking forecast to try to estimate future Internet traffic trends and, as of February 27, 2019, they estimate the annual IP traffic will reach 4.8 ZB by 2022, up from 1.5 ZB in 2017 [10]. These trends are also mirrored in academic institutions. Dartmouth, as discussed above, has been a pioneering institution for networks research and was an early installer of wireless Internet. They noted that in 2001 [25, p. 4] their campus had 1,076 unique devices which generated a total of 3.3 TB in one *semester* [23]. More than a decade later, at the University of Michigan, Ann Arbor Campus (a larger campus than Dartmouth), they measured 100,087 devices and 146 TB in one *day* in 2017; this was also a 3x increase in University of Michigan network traffic compared to the previous year [31]!

Besides traffic, Cisco notes two trends especially pertinent to the rest of this paper: that the bandwidth

requirements of the busiest hour of the day is (projected to) grow much faster than the average daily Internet traffic (a 4.8x vs 3.7x increase from 2017 to 2022) and that by 2022, wireless/mobile traffic will account for 71% of traffic [10]. This second change, the dramatic shift toward mobile users, is especially interesting because it was entirely unexpected when the current Internet architecture was first proposed [11]. While the engineers who designed the Internet were surprisingly prescient (e.g. giving IP addresses 32-bits, enough for ~ 4 billion IPv4 addresses), mobility was entirely unexpected and thus unplanned for; they assumed the Internet would be accessed by fixed-point physical Ethernet links [11]. So, as the Internet is accessed more and more via mobile phones or laptops over Wi-Fi [10] and cellular data, Internet researchers will need to devise new methods to help manage increased wireless usage with increased wireless bandwidth. Currently network engineers rely on layers of messy abstraction and complex infrastructure, such as country-spanning ‘Carrier-Grade NATs’ [37] to provide transparent mobility to users on protocols not designed for it.

2.2 Smarter Resource Planning Can Help

For both of these problems, user mobility and an unprecedented increase in Internet usage, a broad class of solutions center around **smart resource dimensioning**. These solutions try to anticipate and preemptively coordinate Internet traffic by taking advantage of two other shifts in the modern Internet: an increase in data from a variety of sources and the ability to distribute computation to the ‘edge’ of the network (e.g. to the router or access point). The first shift allows for data-driven models to be created which can capture basic trends in network usage; the second shift allows for such models to inform dynamic low-level changes in the network, such as routing decisions.

This idea, of using data-driven models to improve network performance, has been studied for many years with fruitful results. For instance, in 2006, researchers at Dartmouth College used data collected from their college wireless network to predict when a user would be ‘handed-off’ between wireless access points in order to help prevent real-time applications (i.e. VoIP phone calls) from dropping the connection. They found only moderate success in accurately modeling hand-off time but by integrating these imperfect models into the application, still found they significantly helped reduce the number of dropped calls [38].

In a similar vein, there has been a recent move towards a network management approach called *Software-defined networking* (SDN), especially in large data-centers and corporations. This approach is explicitly designed for programmable resource planning by centralizing and separating decisions about how to handle traffic from the devices which actually forward it [17].

However, the Achilles’s heel of any such data-driven approach is the data. In order to have any chance of using the data to make helpful and accurate predictions, the data itself must be accurate and pertinent to the desired models. This is especially a problem in wireless network research because there are few publicly released datasets which are accurate, recent, and comprehensive. The main source of publicly-available wireless network datasets is from Dartmouth College and is called CRAWDAD (*A Community Resource for Archiving Wireless Data At Dartmouth*) but there is a dearth of good datasets. One of the most recent, comprehensive, and widely-cited datasets is the `Dartmouth/campus` dataset, for data collected in Dartmouth from 2001-2006 [26], released in 2007. And while it contains five years of data, as we discussed above, a lot has changed since in the past decade. The most recent dataset as of this thesis is the `kth/campus` [34] dataset from 2014-2015, released July 2019, which captures associations with the educational network *Eduroam* in the KTH Royal Institute of Technology of Stockholm.

The relative paucity of such public datasets is at odds with the ease with which data can be collected, as modern network resource infrastructure is already built with the capability for logging. So large network managers (e.g. in corporations, universities, etc.) already have the capability to collect fine-grained, comprehensive, and real-time data which matches their exact network conditions. Additionally, while a given network’s data is helpful to its network managers, there is also evidence that insights gained from one network can be transferred into insights to other similar networks. Authors of [4] and [14] found reasonably robust similarity in network usage across Wi-Fi hot-spots and city-wide Wi-Fi deployments.

One reason for this difference between the capability to capture and the public availability of such datasets is that this data is valuable enough to be a marketable product in its own right. But another, more pressing reason, is that this data can encode private information about the users who participate. And since the Internet is becoming so integral into daily life, this private information can have wide-reaching implications.

In order to understand the privacy implications of releasing network analysis data, we first need to understand what we mean by privacy and what solutions we can expect to help us preserve the privacy of network data.

2.3 Privacy

The right to privacy is a core tenant of political theory and legal structures in much of modern world. It has been generally enconced in Article 12 of the United Nation’s Universal Declaration of Human Rights [6] and, in the United States, was first articulated in “The Right to Privacy”, a 1890 article in the *Harvard Law*

Review [7]. In that article, they react to how “recent inventions and business methods” demonstrate how action must be taken to protect the right of people “to be let alone” [p 195]. In particular, they mention how the combination of photographs and newspapers have “invalidated the sacred precincts of private and domestic life” and, presciently, how “numerous mechanical devices threaten to make good the prediction that ‘what is whispered in the closet shall be proclaimed from the house-tops’ ” [p 195]. This article is remarkable because, while it is 130 years old, many of the issues it grapples with remain (if not becoming more-so). For one, it demonstrates how there is a constant interplay between technology – which can quickly makes privacy invasions cheaper, easier, and more accurate – and societal expectations – which can be caught off-guard by these sudden shifts.¹ Additionally, it demonstrates how whole industries can foment around acquiring and distributing this (private) information. While *The Right to Privacy* is particularly concerned about the damage gossip columns in newspapers could do to a person (“invasions of privacy [have] subjected him to mental pain and distress, far greater than could be inflicted by mere bodily injury’ [p. 196]), the concept of an ‘information economy’ which can sometimes depend on distributing private information remains true today.

But as Paul Ohm notes in his 2010 article “Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization” [33], the American notion of privacy went from a largely *court-driven* definition used in privacy torts – a civil lawsuit which uses invasion of privacy as a way to justify compensation or to have their privacy reinforced – to a *legislature-driven* approach. In the 1960’s, in the midst of the computer revolution, the U.S. government started computerizing their vast records into large databases, to public outrage over the privacy implications. So the U.S. government started working to understand what the harms of collecting certain information could be to a person and to try to preemptively prevent this. This fundamental change from punishing privacy breaches to *protecting* privacy, is critical because it acknowledges the universal nature of privacy concerns (by not requiring a specific expensive case to be litigated) and brought to the forefront the trade-offs between the ‘riskiness’ of data [33] and its usefulness. These trade-offs, which we will underscore throughout the paper, are significant because, as Ohm discusses, (1) some private information has the potential for society-wide improvements. So the enforcement of privacy

¹One of my favorite examples of this is given by Dr. Latanya Sweeney. She notes that the 1983 Sony camcorder was one of the first consumer device which allowed simultaneous video and audio recording. In U.S. law, stemming from telephone wire-tapping laws, there was distinction between legally video-taping a person in public and illegally recording their private conversations. However, there was no ‘audio-mute’ button on the camcorder so no way for consumers to always follow wire-tapping law. It would have taken pennies to add such a button in. At first, there were highly publicized cases of people being charged with wire-tapping for recording private conversations with public servants [3] but slowly has become more common-place. Now police routinely have body-cameras, many school-buses have video-cameras (to give one example of changing societal mores, the state of Pennsylvania originally was going to sue a bus company for illegally audio recording the bus but a decade later mandated that all buses must have video cameras. [35]), and there is generally a greater recognition of being recorded. This is not necessarily a causal relationship but Dr. Sweeney gives it as one example where technological design forces societal examination (and instigation) of changing norms.

can become a restriction of free information flow, in some ways, a censorship which society agrees upon and the state enforces. But, (2), such private information can be some of the most sensitive information a person cares about, and they have a fundamental right to protect it.

One example which starkly encompasses both of these trade-offs most extreme aspects is medical data. On one hand, medical conditions and histories are some of the most sensitive information we have and would want to keep private (from random strangers as well as employers, friends, etc). On the other hand, such data has the power to revolutionize health care and insights gleaned from them by sharing this private information with medical researchers can save countless lives with precise data-driven medicine. For legislators trying to balance individual rights with society betterment, there is no clear-cut solution to these problems and compromise has to try to balance these outcomes carefully.

One proposed solution for legislating the health care data, still enacted today, is the *Health Insurance Portability and Accountability Act* (HIPPA) which, among other things, is a ground-breaking law to protect the privacy of patients while still allowing research to be done on medical data. It relies on the idea of **Anonymization** or de-identification of data. The idea is to preserve relevant medical data fields but remove fields which could be used to “uniquely identify an individual” [22, see the Safe Harbor section]. It tries to exhaustively list such fields (18 in total) such as name, address, exact birth-date, social security number, medical record number, etc. In order to be considered “de-identified” (and therefore able to be communicated with other providers), such fields need to be removed.

2.4 De-identification Techniques: k -Anonymity

This idea of formally identifying personally identifiable information (**PII**) and removing or obfuscating it was the main method for privatizing datasets for many years and has some intuitive appeal. Coming back to the example of medical data, it seems intuitive that medical researchers would care about, for instance, the symptoms presented, any diagnoses made, and any treatments given. But that they should not care about the social security number, exact name of the patient, their exact birth date, or other PII. Furthermore, if we had only the general symptoms, diagnosis, and treatments, it would seem intuitively implausible for anyone to identify the person, especially if over a large population.

This intuition has been generalized into many different anonymization techniques. In 2002, Latanya Sweeney published a landmark paper which introduces one such technique called **k -anonymity**. The idea behind k -anonymity is to ensure that any personally-identifiable information about one person in a database

would be indistinguishable from at least $k - 1$ other people in the database. Using our medical example, a medical database could not be published if I was the only person with my birthdate and zip code. But other non-personally-identifiable information would be left alone. In order to ensure this does not happen, Sweeney proposes two methods: suppression and generalization, to ensure the k -anonymity criteria holds. Without going into too much detail, suppression removes data from the database while generalization aggregates data. So, a data-holder might suppress social security numbers but generalize zip codes by taking only the three number prefix of the full zip code.

While foundational, there are some subtle issues with using k -anonymity to preserve privacy. The first issue is that the data-holder needs to decide what parts of the dataset could be personally-identifiable for the participants. These parts are called ‘quasi-identifiers’ and it can be very unintuitive, if not impossible, to know exactly what information could be used to identify someone. For one, surprisingly many datasets have surprisingly small ‘unicity’, the number of data-points which uniquely identify a person. Sweeney found that in the 1990 U.S. Census, 87% of the U.S. (216 million / 248 million) was uniquely identified based on only [5-digit zip code, gender, date of birth] and 18% were uniquely identified by [county, gender, data of birth] [40]. Looking at human mobility data using mobile-data carrier antennas, four spatio-temporal points (where the temporal resolution was an hour and the spatial resolution depended on the density of carrier antennas from 0.15 km² to 15 km²), was enough to uniquely identify 95% of individuals [12].

Additionally, any released dataset does not exist in a vacuum and there can be seemingly unrelated datasets which be combined together to identify people, called a *linkage attack*. One of the most famous examples of linkage re-identification also investigated by Sweeney in 1997, and is the motivation behind k -anonymity. In Massachusetts, a state insurance company sold ‘anonymized’ health care data to researchers which included fields for [diagnosis, procedure, medication, zip-code, birth data, sex] but not, of course, name. Dr. Sweeney bought this data for \$20 and combined this with public voter registration lists, which included [name, zip code, birth-data, sex, party affiliation]. In the intersection, she uniquely identified Governor William Weld, the governor of Massachusetts, in the medical data and learned private information about his disease diagnosis. This study was so momentous that it was quickly noticed by law-makers and is one of the main reasons for HIPAA including a list of personally identifiable information which includes zip code and birthday.

However, even though k -anonymity was designed to counteract parts of these linkage and unicity attacks, it is still insufficient. The design of k -anonymity depends on the data-holder being able to identify all quasi-

identifiers in the dataset but the unicity could be unintuitively low and there could exist current datasets which allow linkage attacks. Even harder, once the data is in the public domain, it remains there forever, so there could be *future* datasets which allow linkage attacks. Identifying all current and future possibilities for such attacks is a tremendous challenge.

Finally, there is a related method which holds some intuitive charm but does not actually protect privacy. The idea is to use well-established hashing algorithms, which are designed to give deterministic non-invertible outputs for a given input, on the private data fields. This technique has many benefits because they are fast, easy, and consistent so allow linking between different datasets. So, in other words, if another data-holder has similar data (i.e. email addresses), they can combine datasets by using the same hash, without having to share ‘ground-truth’ datasets. This is especially popular with companies who use email hashes to identify users as they claim they can freely sell/distribute this data without infringing on privacy. But, while hashes can be made cryptographically non-invertible against brute-force attacks, they are *not* secure against linkage attacks. The existence of data breaches (e.g. HaveIBeenPwned which has 9 billion breached accounts [24]) as well as commercial data brokers have created massively comprehensive lists of people’s email addresses. These can easily be hashed and used as a look-up table (in cryptographic terms, a rainbow table) to invert the hash output for emails. Hashing is so fast and optimized, that if we assume around 5 billion email addresses in use, then all their MD5 hashes could be computed in ~ 10 milliseconds on an Amazon EC2 machine for \$0.0069 [2].

2.5 De-identification Techniques: Variants on k -Anonymity

There are other privatizing techniques which use a similar idea on k -anonymity but try to improve upon it in some way. To give two brief examples, ℓ -**diversity** is an extension of k -anonymity which tries to create equivalence classes of values of quasi-identifiers (i.e. grouping data rows by whether they share values for quasi-identifiers). Then it requires that at least ℓ ‘well-represented’ different values of the sensitive attributes exist within each equivalence class. To give an example in the medical setting, in k -anonymity we might have k different people with the same zip code and birthday, all of whom have different life-threatening diagnoses. If I know my neighbor is in such a zip code, I wouldn’t know *which* illness they have, but I know they have one of them; a breach in privacy. So, we require there to be different values for the diagnosis field, ‘well-represented’ such that it includes a non-life-threatening value. So I wouldn’t know if my neighbor has a life-threatening illness.

An extension of ℓ -diversity is **t -closeness** which tries to address the above problem as well as linkage attacks. It does this by codifying background/outside information and separating that from the data given by a new dataset. In particular, it also looks at equivalence classes of quasi-identifiers but enforces that the *distribution* of sensitive information within the equivalence class is close to the distribution of that sensitive information in the entire dataset. The idea is that the global information for the sensitive attribute throughout the entire dataset should be/is already public (e.g. the height distribution of all adults) but the distribution for any particular small group of people should be private (e.g. the height distribution of people in a zip code with a certain exact birthday).

While both of these methods are clever and each offers more robust privacy, they both suffer some of the same overarching issues as k -anonymity. Namely there is still a worry about linkage attacks and it is not easy to confidently identify what attributes are quasi-identifiable (or, for t -closeness, sensitive). Furthermore, if we are attempting the (maybe overly) ambitious goal of canonizing a mathematical definition of ‘privacy’ which encapsulates our vague intuitive understanding, these definitions are somewhat unsatisfying. While t -closeness seems to get closer, with its separation of the global and intra-equivalence-class distributions, none of them really capture what a layperson might think of when they think of privacy.

These desires, to have a more intuitive mathematical definition for privacy as well as to feel more confident about the mathematical protections which our definition gives against privacy infringing attacks, such as linkage attacks, leads to our final privatization technique, differential privacy.

2.6 Differential Privacy

2.6.1 Definition

The driving force behind our definition for differential privacy is a new mathematical basis for our intuitive understanding of privacy. Informally, given a data set, D , we focus on on a single individual (a row) in the dataset and create a new dataset, D^- , in which that individual is removed. Then for D to be differentially private, we require that any query returns ‘similar’ results whether it is run on D or D^- . In other words, a dataset can’t breach any person’s privacy if no individual is allowed to affect any result ‘much’, so no one would mind being included and the potential risks of being included are ‘small’ [28].

Another way of thinking about this is to imagine an attacker tries to breach the data-set’s privatization by going from the privatized output to the ground-truth dataset. The attacker could exhaust all the com-

binatorial possibilities for the ground-truth dataset and see if the ‘plausibility probability’ was higher for some of the enumerated datasets. Any release will say *something* about the the landscape of plausibility probabilities but we want to guarantee that any plausibility probability peaks are reasonably small, so the attacker cannot learn much. In fact, there is a powerful result called the Database Reconstruction Theorem [13] which says that if too many query results are published with low enough noise addition (on the order of \sqrt{N}), then an attacker can reconstruct the entire dataset with high certainty.

Formally we give the following definition:

Definition 2.1. ϵ -Differential Privacy Given in [15, Defn. 2.4].

A randomized algorithm \mathcal{A} provides ϵ -differential privacy ($\epsilon \in \mathbb{R}$, $\epsilon > 0$) if, for all neighboring datasets D_1 and D_2 which differ by one row and for all possible outputs O of \mathcal{A} , we have that

$$\frac{\Pr[\mathcal{A}(D_1) = O]}{\Pr[\mathcal{A}(D_2) = O]} \leq e^\epsilon.$$

To unpack this definition, if we look at *all* possible outputs of \mathcal{A} , we guarantee that the probability of producing a specific output O from D_1 and D_2 is nearly the same. More specifically, we refer to ϵ as the **privacy budget** or the privacy stringency parameter because it constrains the maximum difference that any queries from neighboring databases can be. Looking at typical small value of $\epsilon = 0.1$, we get $e^\epsilon \approx 1.1$, enforcing that for any participant present in D_1 , their absence in D_2 is not able to change the result by more than a small factor.

While the choice of an appropriate value of ϵ is an open question, its existence enables us to trade off privacy and accuracy in a definitive, reasoned way. As an example, the 2020 U.S. Census will use differential privacy to privatize its results and their use of DP allows them examine this privacy-accuracy trade-off as it relates to an estimated marginal social benefit of data-release [1, Pg. 36]. This type of analysis would be impossible in, for instance, k -anonymity which doesn’t have such quantifiable guarantees.

2.6.2 Differential Privacy: Basic Building Blocks

We will not go into too much detail on the particulars of DP algorithms but attempt to give an overview of the algorithmic landscape. A seminal paper by Dwork and Roth [15] provides one simple algorithm called the **Laplace algorithm** which satisfies Definition 2.6.1. In this algorithm, given a desired (numerical) output function, you perturb it by adding random noise from a Laplace distribution with a specific scale

parameter (the parameter which governs the standard deviation of the distribution). This DP-satisfying scale parameter depends on two things: (1) the desired privacy budget ϵ and (2) the ‘ ℓ_1 sensitivity of f ’. This **sensitivity** measures the frequency of attributes of any single row in the dataset.

Formally:

Definition 2.2. The ℓ_1 sensitivity of f Given in [15, Defn. 3.3]

Given a real-valued query on a dataset, $f : D \rightarrow \mathbb{R}$, we define the the ℓ_1 sensitivity of f , denoted Δf , as

$$\Delta f = \max_{\text{Neighboring datasets } D_1, D_2} \|f(D_1) - f(D_2)\|_1.$$

This sensitivity gives the maximum magnitude difference for the function f on any neighboring datasets (i.e. datasets differing by one row). The proof that the Laplace algorithm satisfies ϵ -differential privacy is given in [15, Thm. 3.6] and is elegant and 5 lines long, using only the definitions of sensitivity, ϵ -differential privacy, and the triangle inequality. This algorithm gives the foundation that many more complicated DP algorithms use.

2.6.3 Useful Properties of Differential Privacy

One of the most convenient aspects of our definition for a differentially private algorithm is that it allows us to compose multiple differentially private sub-algorithms and easily reason about the privacy guarantees of the ensemble algorithm. This allows theoreticians to focus on (im-)proving privacy guarantees for manageably small sub-algorithms and algorithm developers to create complex ensemble algorithms using these parts.

In fact, we can elegantly state theorems about both parallel and sequential composition of multiple DP algorithms.

Theorem 2.1. Sequential Composition Given in [15, Corollary 3.15].

If M_1, M_2, \dots, M_k are algorithms where each M_i satisfies ϵ_i -differential privacy then, running all k algorithms sequentially on a dataset D (such that we first run M_1 on D , then M_2 on that output, and so on) satisfies ϵ -differential privacy overall where $\epsilon = \epsilon_1 + \epsilon_2 + \dots + \epsilon_k$.

Theorem 2.2. Parallel Composition Given in [30, Thm. 4].

If M_1, M_2, \dots, M_k are algorithms where each M_i satisfies ϵ_i -differential privacy then, running all k algorithms in parallel (such that each M_i is run on a disjoint dataset D_i and then they’re all put together) satisfies ϵ -differential privacy overall where $\epsilon = \max\{\epsilon_1, \epsilon_2, \dots, \epsilon_k\}$.

Parallel composition is especially useful when applying DP algorithms to individual bins of a histogram as these are disjoint by definition.

Another especially useful property of our DP definition is that being differentially private is invariant under any post-processing steps. Formally,

Theorem 2.3. Post-Processing Given in [15, Propn. 2.1].

If $M : D \rightarrow O$ is an ε -differentially private algorithm and $f : O \rightarrow O'$ is an arbitrary randomized mapping, then $f \circ M : D \rightarrow O'$ is ε -differentially private.

Combined, these foundational results allow us to make complex algorithms from post-processed combined building blocks and still guarantee differential privacy with the desired stringency.

2.6.4 Differentially Private Algorithms

The goal of all these algorithms is to get the best possible accuracy with the most stringent privacy requirements (lower ε). To give an overview of the algorithmic landscape, we'll characterize DP algorithms on two axes: being (1) data-dependent and (2) workload-aware. Both of these strategies try to exploit extra information to help achieve better privacy-accuracy trade-offs. The pain and panacea of taking this approach is that the error now depends on (potentially unknown) properties of the dataset and the workload so it can be difficult to compare different algorithms and even harder to know *a priori* which of these algorithms will work best on a given dataset and workload.

One way of doing this is to exploit properties of dataset itself, in which case the algorithm is called **data-dependent**. An example of a data-dependent DP strategy is to partition the data into buckets, similar to making a histogram of the data and then apply a DP algorithm to each bucket. Critically, these algorithms depend on the assumption that the data is uniform within a given bucket but the relative accuracy of this assumption depends on the dataset itself. Such data-dependent algorithms can try to adaptively partition the dataset into buckets which are more uniform (e.g. [19]) but this can be challenging to do while maintaining the properties of differential privacy.

Another strategy is to exploit properties of a representative workload. Given a dataset, there might be a set of queries ('workload') which are especially critical or particularly relevant to data-analysis on which we want to be accurate. An example of such a workload-aware DP strategy is given in [36], which also focuses on range queries on a histogram dataset (i.e. a set of queries on the cumulative count of ranges over histogram

bins). The idea behind such algorithms is to adaptively generate a partition for the data which minimizes the error for a given workload of queries (as opposed to uniformity in some data-dependent algorithms). Any query can use this partitioning to find an approximate answer but the noise is most optimized for the desired workload.

We give an overview of classifying different DP algorithms in the following Figure: 1





 Workload-aware MWEM, DAWA, GreedyH	 Data-dependent Uniform, MWEM, AHP, DAWA, SF, AGRID
 Workload-unaware Identity, H, H _b , Privelet, Uniform, AHP, SF, AGRID	 Data-independent Identity, H, H _b , Privelet, GreedyH

Figure 1: Table showing classification of different differentially-private algorithms classified as workload-(un-)aware and data-(in-)dependent. See [20]

2.6.5 Comparing Differentially Private Algorithms

A large part of the impetus for this project is the overall *lack* of evaluation surveys for DP algorithms. However in 2016 Hay *et al.* proposed and used a set of metrics to evaluate many different DP algorithms on a wide range of datasets and across many different ϵ [20]. They considered the following properties about the dataset - the size (# rows), scale (# input dimensions, and shape (distribution) of the data to quantify an algorithm’s error on a workload. They use an error metric called the scaled average per-query error (**SPQE**) which we will also use in our analysis.

Definition 2.3. Scaled Average Per-Query Error (SPQE) Given in [20, Defn. 3]. Let W be a workload of q queries, x a data vector and its scale be $s = \|x\|_1$. Let $\hat{y} = \mathcal{A}(x, W, \epsilon)$ be the noisy output of algorithm \mathcal{A} . Then given a loss function L , we define the scale average per-query error as $\frac{1}{s \cdot q} L(\hat{y}, Wx)$.

3 Results and Discussion

This study represents the beginning of the investigation into the application of differential privacy to networks data so our results are a combination of foundational results on the networks data of a prototypical university campus and a compelling set of future directions.

3.1 Our Data

Our data was collected from the University of Massachusetts, Amherst [39]. The campus has around 27,000 students across the undergraduate and graduate schools with approximately 9,000 staff members and sits on a 1,450 acre campus. Its network infrastructure, at the time our data was collected was 4,500 802.11 ARUBA™ Wi-Fi access points connected to ARUBA controllers [41].

3.2 Data Pipeline; Building Models

One requisite step for this and future work was to build up the infrastructure in order to go from raw data to representative statistics which can be used in network models. Starting from raw syslogs collected by the ARUBA controllers the data was first pre-processed to extract the (dis-)association and (de-)authentication messages per MAC address, and cleaned to remove ping-pong effects. The pipeline [39] uses the steps outlined above to transform the syslog messages into a sequence of “presences” at each AP. A presence corresponds to the length of association of a device at an AP. Each presence is a 4-tuple of the form [MAC ADDR] : (START_TIME, END_TIME, AP_NAME, DURATION). Every MAC address found in the Syslogs has an aggregated list of presences ordered by time of arrival to the campus network.

Additionally, our data was collated into a **session**, which we colloquially think of as a person in-transit. Steshenko *et al.* [39] empirically found presences separated by a gap of approximately 15 minutes to be a reasonable trade-off for specificity of the session cutoff [39] We validated this finding in our dataset as well by plotting the separation times between presences and found the “knee-of-the-curve” at approx. 15 minutes. For a given MAC address’ sequence of associations we start a new session grouping whenever the time between subsequent associations was more than 15 minutes.

In particular, the data we used in our experiment is 1.3GB of network log data spanning 15 days from 2014-09-16 to 2014-09-30. (Table 1).

Number of Days	15 (<i>2014-09-16 to 2014-09-30</i>)
Total # Log Lines	16,254,839
# Access Points	4,179
# Buildings	131
# Unique MAC Addresses	$\sim 60,000$ ($\gg 36k$ <i>students + staff</i>)
Total # Sessions	1,817,975
Average Number of Sessions per MAC	31.7
Average Number of Visited AP's per Session	8.9

Table 1: Basic statistics of our networks dataset, collected from the University of Massachusetts, Amherst. Each ordered set of presences is delineated into sessions whenever there is more than 15 minutes between successive AP associations.

In order to use differential privacy to give privatized real-world datasets to the networks research community, we tried two different approaches. One approach was to focus on *count queries* and optimize the privatization for specific, well-known mobility statistics such as average number of users on campus or the average arrival rate of new users into the campus network. These statistics can provide insight into network location occupancy queries; e.g., anticipating the top "k" most congested access points. A guiding principle behind exploring these statistics was to eventually use well-known 'ground-truth'/unprivatized network models, which were shown to be accurate, and ultimately privatize all their input parameters. In this way we could get the both the accuracy of a well-developed model and the privacy from a differentially private algorithm.

Another approach was to release trajectories of users, i.e., AP affiliation over time. We can then optimize the privatization to be able to characterize individual mobile node trajectories and ask queries such as the number of mobile users who are most often resident at the same or nearby AP, the number of trajectories sharing more than a given number of common APs, and more generally queries about network AP occupancy *over time*.

3.3 Count Queries Comparison Pipeline

As discussed above, our approach which used count queries was based around privatizing well-known mobility statistics which previous literature had used to model network mobility. We focused especially on the mixed-queuing model given by Chen *et al.* [9] as it focuses on modeling and predicting AP occupancy distribution, the average number of AP transitions per user, and anticipated network capacity for a predefined performance threshold. These are exactly the type of queries we want to address with our differentially private dataset.

Our main contribution in this area was an empirical comparison of how DP algorithms performed on

our dataset for mobility related queries. This comparative analysis is helpful to give insights into which algorithms other data managers might want to apply, especially when comparing complicated data/workload aware algorithms against simpler algorithms. Furthermore, in an ideal world, privatizing data managers would not empirically compare multiple algorithms to get the best performance before release because the (public) choice of algorithm itself could give information about the underlying unprivatized distribution.

This sort of analysis is relatively rare in the literature and one of the only papers to do so, were Hay *et al.* in 2016 in their pivotal work on DPBench [20]. They compared a wide range of publicly available datasets across a set of DP algorithms and found complex interactions between the size (# rows), scale (# input dimensions), and shape (distribution) of the data. DPBench assumes that the input is discretized, either by already being discrete or by binning a continuous dataset. The more fine-grained the bins are, the more data needs to be preserved under privatization. While they tested 15 algorithms over 27 datasets with many different parameters and demonstrated that such a complex interdependence exists, it is not that helpful to *a priori* decide which algorithm works best with a given dataset, much less a given dataset with desired workload. There are a few mobility datasets in this benchmark but they are all based on the start/end of GPS trajectories which is different from our dataset in both the continuous nature of GPS coordinates and the lack of the intra-trajectory information. There was only one time-series dataset and no network mobility datasets.

Even beyond the qualitative differences of trying to apply insights from non-mobility data to mobility data, DPBench showcases the large effect of shape (distribution) on the effectiveness of different data-dependent algorithms but it is unclear how to transfer accuracy information from this set of datasets to other datasets.

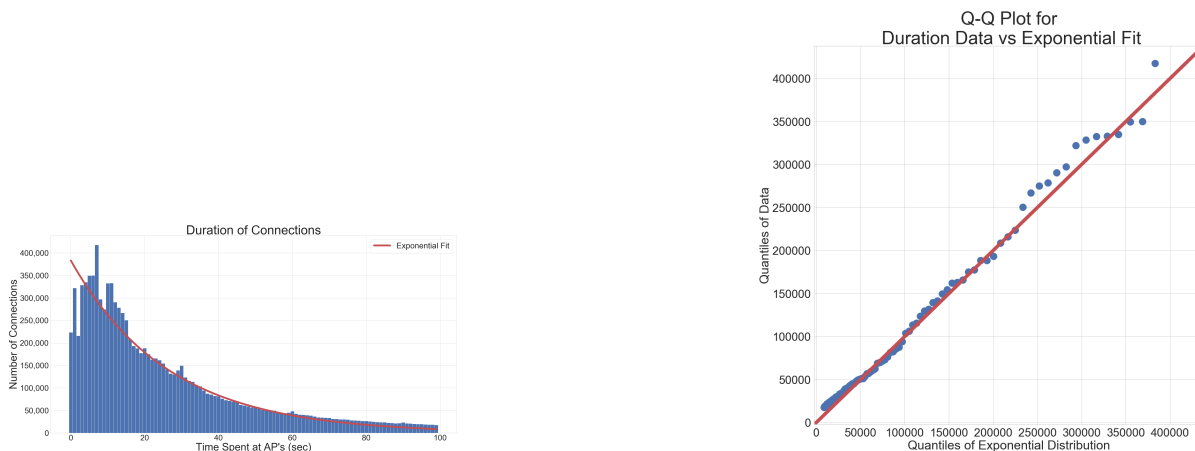
In order to make this comparison, we used an open source framework called ϵ ktelo [44] which is designed to abstract over different DP algorithms in a standardized, comparable, and scalable way. It has pre-programmed versions of many different published algorithms for count queries and allows switching between them easily. Additionally, there are myriad subtle details in implementing differentially private algorithms. As an example of a interesting and surprising example, Mironov [32] found a DP leakage attack based on the finite precision of floating point numbers which forces (a naive implementation of) the sampled Laplace distribution to be ragged and non-translation-invariant. Their ultimate solution, a so-called ‘snapping algorithm’ is complex and unintuitive, highlighting why an open source framework allows the community to fix such subtle bugs, instead of re-implementing an algorithm from scratch.

We used ϵ ktelo with the SciPy Python library to ingest, transform, and examine the dataset.

3.4 Count Queries Comparison Results

3.4.1 Service Time

To give an example of DP privatization as applied to a count query, we’ll first focus on the **service times** of network associations in our dataset. This is the duration for each MAC address’s association with an access point and can be calculated readily from the data. Figure 2a gives a histogram for the service time distribution, along with an exponential fit. Using a Q-Q plot (Figure 2b) and running a Kolmogorov–Smirnov test indicates that this exponential fit is a reasonable fit for the empirical distribution ($D = 0.07, p \ll 0.5$).



(a) Histogram of service time (duration of connection) for associations in the dataset along with the fitted exponential regression.

(b) A Q-Q plot of the service time distribution from (a) vs the exponential regression. Note that the Kolmogorov–Smirnov test indicates this is a significant fit.

Figure 2

Next we privatized this distribution through a range of DP algorithms and for different privacy budgets (the ϵ parameter) given by ϵ ktelo (Fig. 3). We chose the MWEM [19]; GreedyH [27]; H_b , DAWA [36], Identity [16], Uniform (see [20]) algorithms as they represent a wide variety of complexity and whether they are data-dependent or workload aware. See the differential privacy section and [20] for an overview of these algorithms.

Figure 3 visually shows how GreedyH, H_b , Identity, and DAWA look similar across a wide-range of ϵ stringency values with MWEM and Uniform not matching the distribution well. However the large variability in x values can make it difficult to notice subtle deviations (such as how DAWA at small epsilons forces sections of the curve to become flat steps). So we also looked at Q-Q plots (Fig 4) and the scaled error (Fig 5) for these privatization algorithms.

In particular, the scaled error we will use in the rest of this analysis is the **Scaled Average Per-Query Error (SPQE)** from DPBench [20, Defn. 3] as discussed in the previous works section.

The scaled error is particularly helpful for quantifying the accuracy for a range of algorithms, across a range of ε values and it matches our visual intuition with MWEM and Uniform being uniformly worse than the rest while DAWA is much better but still worse than GreedyH, H_b , and Identity. It also shows how DAWA is worse at more stringent ε while the SPQE of the three best algorithms are linear with increasing ε .

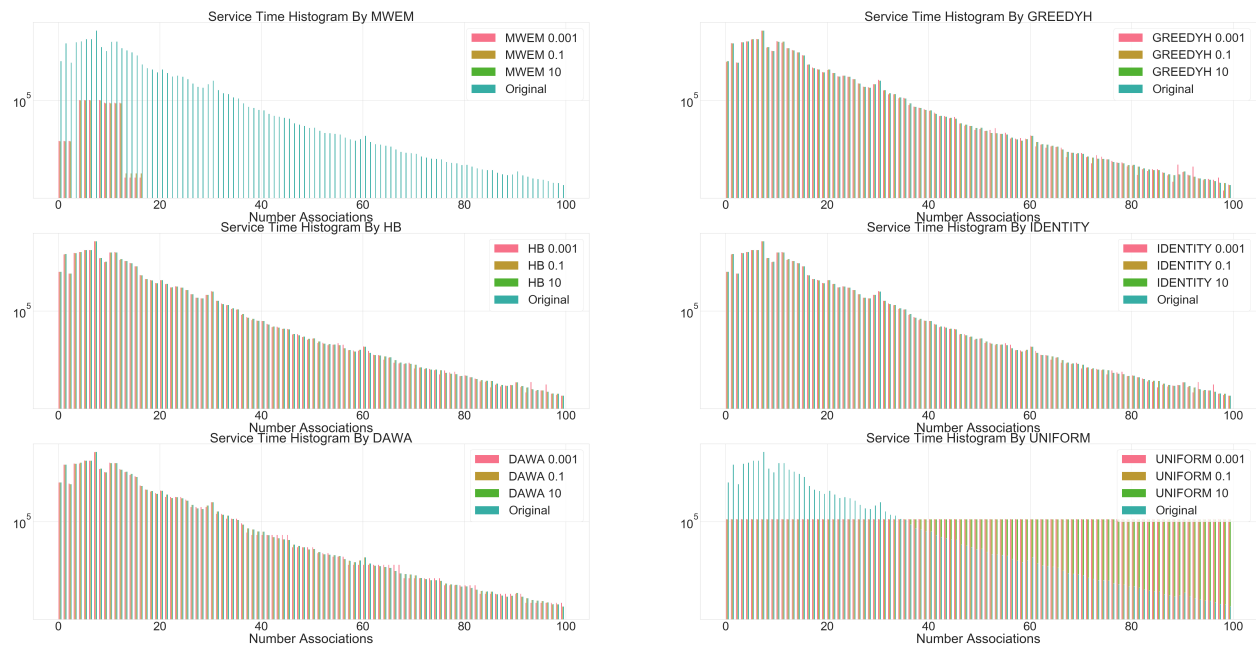


Figure 3: Service time histogram privatized by six DP algorithms – MWEM [19]; GreedyH [27]; H_b , DAWA [36], Identity [16], Uniform (see [20]) – over a range of the privatization stringency parameter ε . Note how all algorithms except for MWEM and UNIFORM give a visually reasonable error rate (quantified in Figure 5) but that the large range in count values can hide relative error values.

3.4.2 Session Start Time

Another statistic which is useful for mobility models is to create a histogram of session start times (Figure 6a). This allows us to model the diurnal rhythm of campus Wi-Fi usage and see how the network arrivals on campus change throughout the day. Note that this is different than the system load at a given time, displayed for comparison in Figure 6b which shows the devices that are present at an AP; a product of the arrival rate at an AP and the mean duration at an AP. From our definition of session, a new session starts when a user changes AP's after more than 15 minutes of not changing, so the session start time distribution gives more information about movement, rather than occupancy.

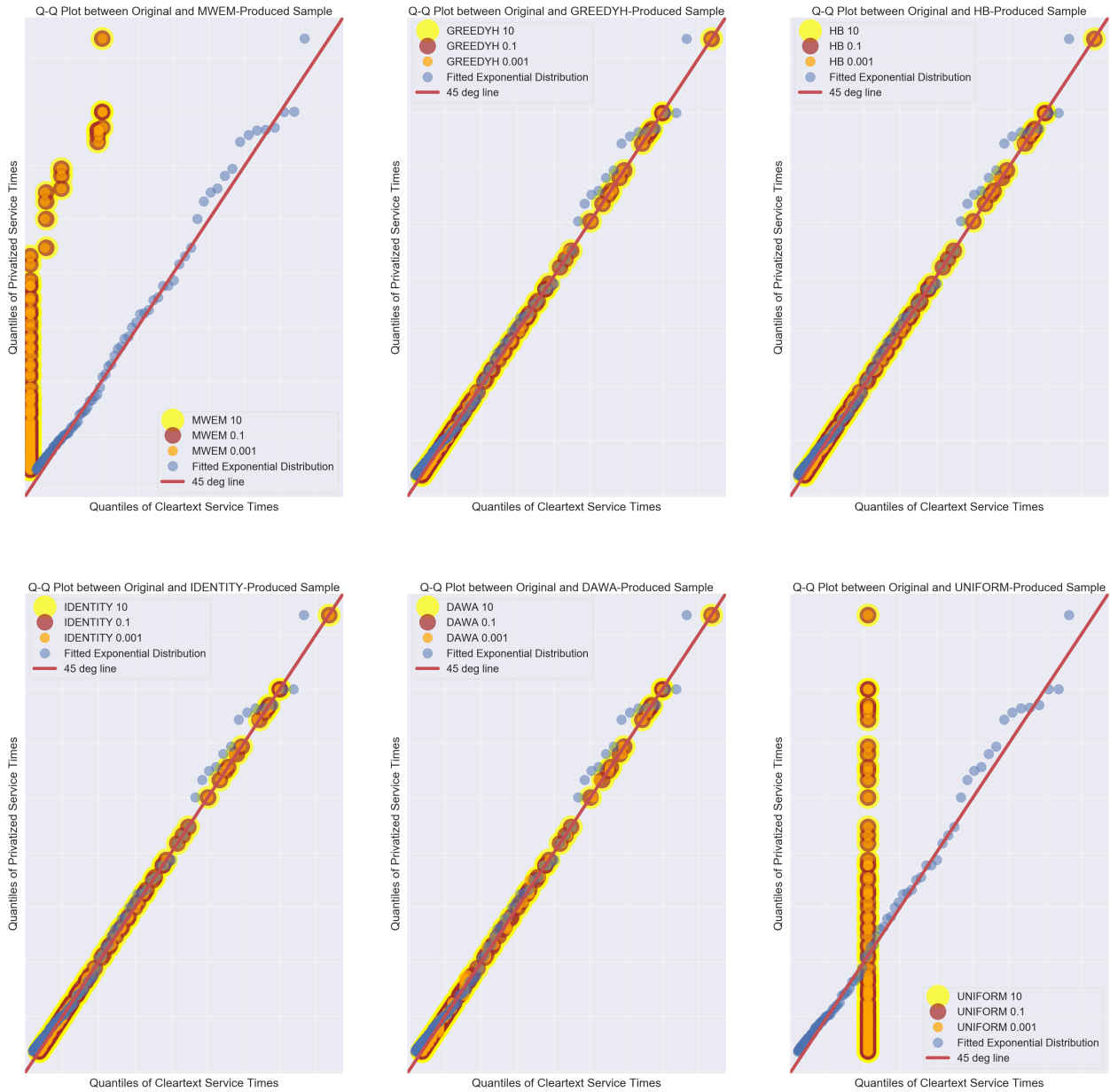


Figure 4: Q-Q plots of the ground-truth service time distribution vs (1) the exponential regression and (2) six DP algorithms – MWEM [19]; GreedyH [27]; H_b , DAWA [36], Identity [16], Uniform (see [20])– over a range of privatization stringency parameter ϵ . Note how all but MWEM and UNIFORM give closer quantile fit than the statistically significant exponential fit (blue dots). Also note that that all algorithms show a consistent quantile agreement even across a large range of privacy budget ϵ .

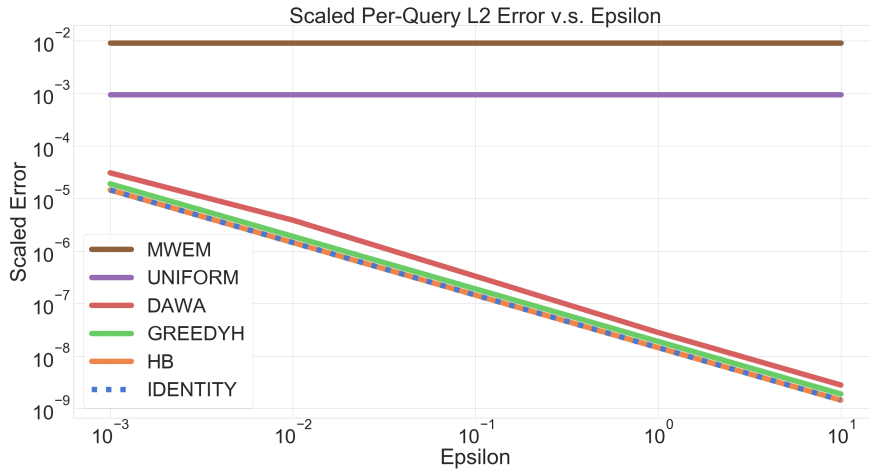


Figure 5: Plot of the scaled average per-query error vs ϵ (privacy stringency) for six different DP algorithms (MWEM [19]; GreedyH [27]; H_b , DAWA [36], Identity [16], Uniform (see [20])) on the service time distribution. See Figures 3, 4. This error metric was standardized by DPBench [20, Defn. 3]. Note how, besides MWEM and UNIFORM, the algorithm’s error decreases linearly with increasing privacy budget ϵ and how all linear algorithms performed very similarly to each other.

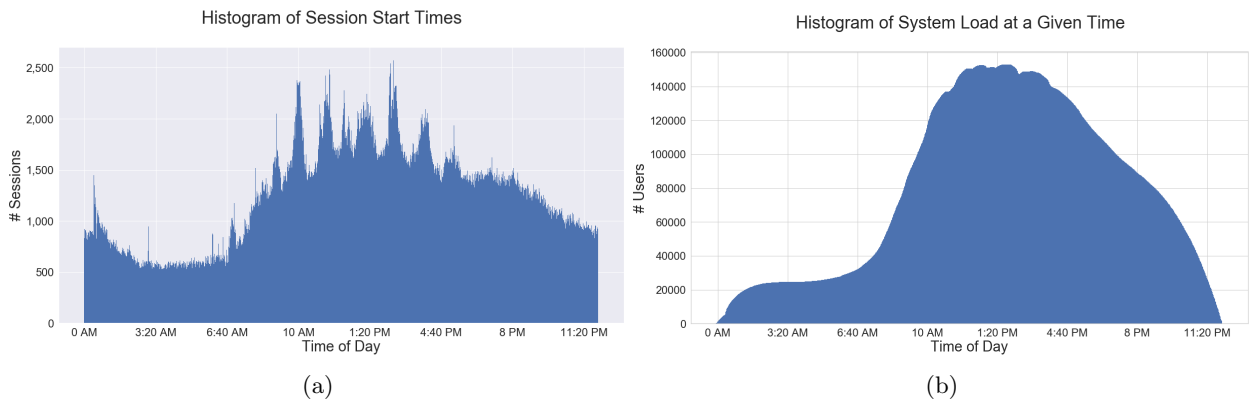


Figure 6: Comparison of the histogram of session start times (a) vs histogram of the system load (b) (number of devices associated with the network) at a given time. Each ordered set of presences is delineated into sessions whenever there is more than 15 minutes between successive AP associations. Note how the session start time has higher spikes, as new sessions are created during higher campus mobility and that the system load displays the expected diurnal distribution.

Then, in a similar vein, we wanted to apply our DP privatization algorithms to this arrival time histogram to examine their relative performance. Figure 7 shows the result of creating noisy histograms of the arrival times, by running the same six DP algorithms. It is particularly interesting because it unveils the relative techniques and limitations of the different algorithms. For instance many of the algorithms have sharp outliers at $\epsilon = 0.001$ except for DAWA and Uniform. DAWA’s technique for ‘sub-binning’ the data, in which it creates larger uniform bins which try to mirror the larger distribution, is also shown at $\epsilon = 0.001$.

We also plot the same SPQE metric as above (Fig 8 which quantifies our qualitative understanding, showing how DAWA has the least error for the most stringent ϵ but at less stringent ϵ , Identity and GreedyH have smaller error.

3.4.3 Understanding Workload-Aware Trade-offs

Another aspect of differentially private count queries we were interested in was how to best take advantage of workload-aware algorithms such as MWEM [19] and GreedyH [27]. In these algorithms, as discussed in section on DP algorithms, we use information about a typical set of queries (called a ‘workload’) in order to inform the allocation of error and hopefully reduce the amount of error for these queries. This presents a double-edged sword because it could allow us to make our privatization ‘aware’ of a typical query to more accurately calculate the statistics we think will be most useful to the networks community (e.g. duration). But it also means that our resulting privatization could become much less useful for general queries a researcher could be interested in.

So, to give a simple proof of concept, we used the duration data as above and made our algorithms ‘aware’ of a Random Range workload, a set of queries that asked for the total number of associations which were in a random interval of duration sizes. Then we tested our algorithms error using the SPQE metric when it was tested on the same Random Range workload as well as tested on the Identity workload, when we query for the value of individual duration histogram bins.

3.4.4 Count Queries Discussion

To summarize our count query results, we transformed access-point syslog data in order to generate privatized network statistics. In particular, we focused on noisy histograms for service time and session start times statistics across a range of six ϵ -differentially private algorithms.



Figure 7: Session start time histogram privatized by six DP algorithms – MWEM [19]; GreedyH [27]; H_B , DAWA [36], Identity [16], Uniform (see [20])– over a range of the privatization stringency parameter ϵ . Note how this data-set offers a greater challenge for privatization, compared to service time, and that most of the algorithms have sharp spikes at the lowest $\epsilon = 0.001$. Comparatively, DAWA creates adaptive sub-bins for its histogram and then assumes uniformity within the bins, which does not lead to spikes but stepped edges.

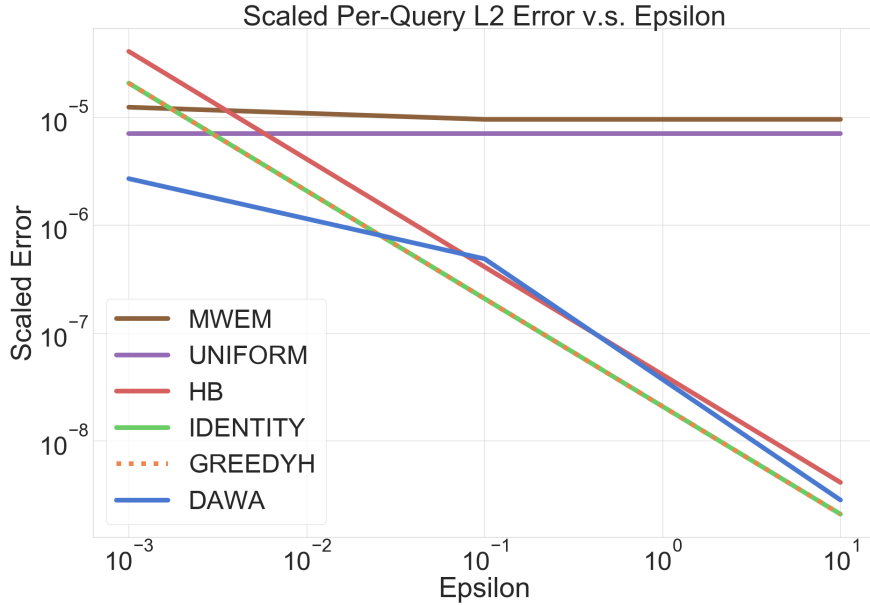


Figure 8: Plot of the scaled average per-query error vs ϵ (privacy stringency) for six different DP algorithms (MWEM [19]; GreedyH [27]; H_b , DAWA [36], Identity [16], Uniform (see [20])) on the histogram of session start times. Note how the error of IDENTITY and GreedyH decreases linearly with increased privacy budget ϵ but DAWA has a knee, in which it's more comparatively accurate at the smallest ϵ . See the count query discussion for one putative explanation.

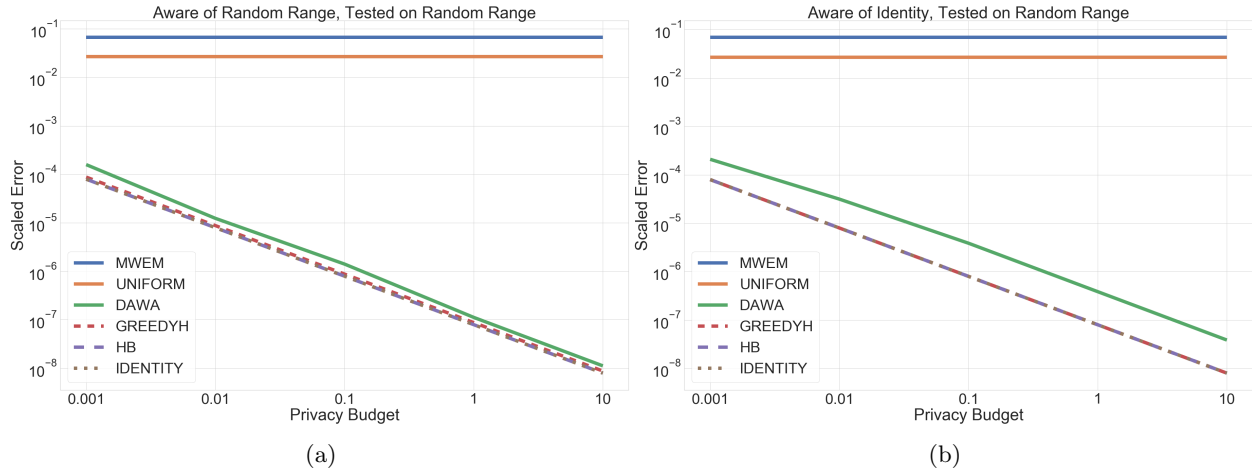


Figure 9: Plots of the scaled average per-query error vs ϵ (privacy stringency) for six different DP algorithms (MWEM [19]; GreedyH [27]; H_b , DAWA [36], Identity [16], Uniform (see [20])) on the service time distribution. All algorithms were made 'aware' of a Random Range workload with the error for (a) was also calculated against a Random Range workload while the (b) was calculated against an Identity Workload. Note how the workload-unaware algorithms don't change error, as expected, but DAWA and H_b both get less accurate when the workload it's aware of differs from the test workload.

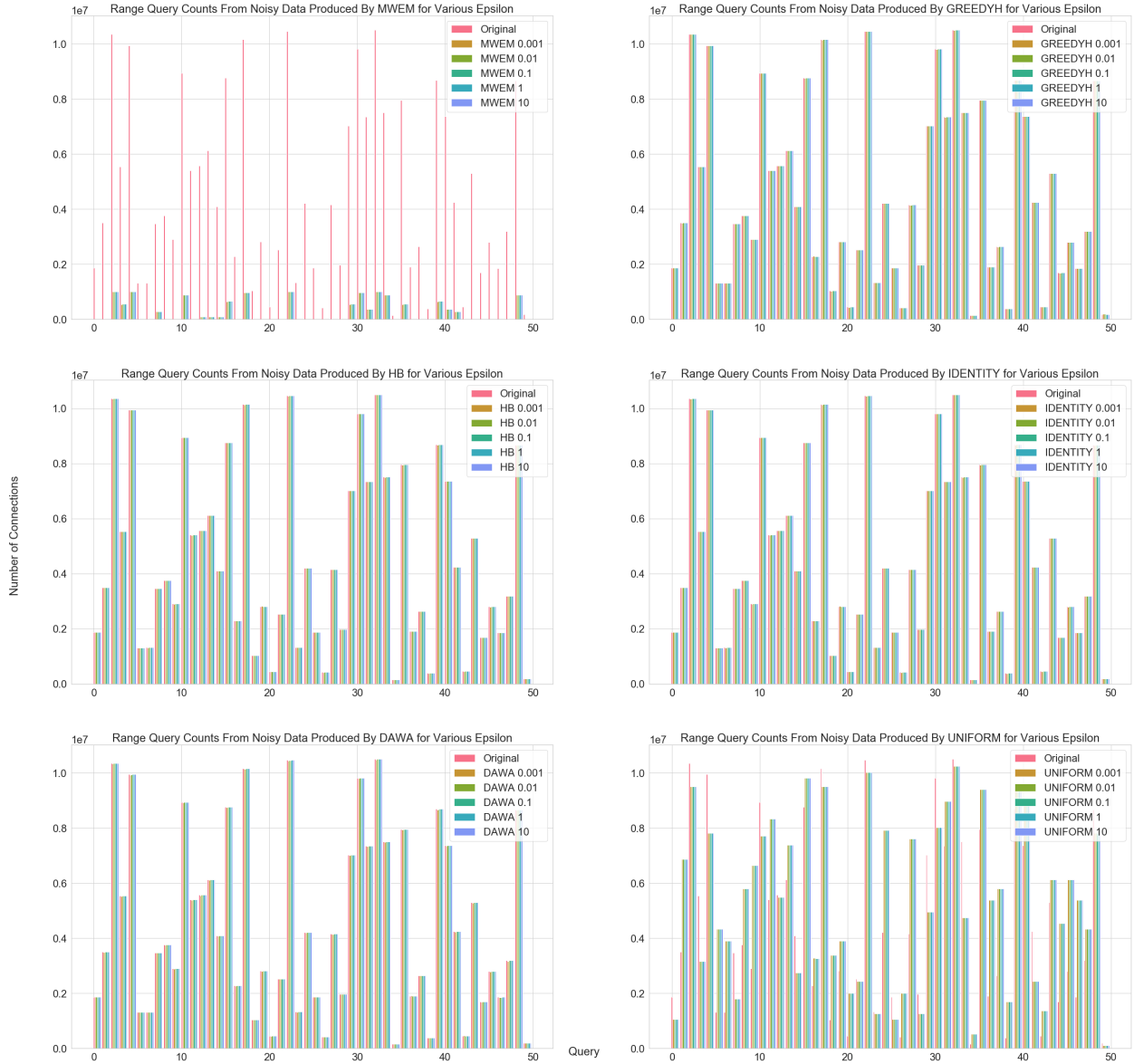


Figure 10: Service time histogram privatized by six different DP algorithms (MWEM [19]; GreedyH [27]; H_b , DAWA [36], Identity [16], Uniform (see [20])) over a range of privatization stringency parameter ϵ which were made ‘aware’ of the Random Range workload but tested on a different, Identity, workload. This corresponds to the SPQE error shown in Figure 9b.

Comparing the SPQE error metric for our algorithms across both data-sets (Figs. 5, 8), we see that the most accurate algorithms for the service-time data-set are H_b , Identity, and GreedyH while the most accurate algorithms for the session-start data-set are Identity, GreedyH, and DAWA. All but one of these algorithms are data-independent (H_b , Identity, and GreedyH are data-independent while DAWA is data-dependent). *This is in agreement with DPBench [20] which found that the effectiveness of data-dependence decreases as the data-set gets larger.*

However this previous finding, shown across a wide range of data-sets, makes it especially intriguing that DAWA (**D**ata-**a**nd **W**orkload-**A**ware algorithm) also has the least error (by almost an order of magnitude) for session-start times at the most stringent ϵ ($\epsilon \leq 10^{-2}$). One possible explanation for this can be seen by combining insights from [20], as well as an understanding of the underlying privatization strategy of DAWA. One property of many DP algorithms which DPBench identified was ‘scale-epsilon interchangeability’, which says that the effect on scaled error is the exact same between increasing the data-set’s scale by a factor as well as increasing the epsilon by that same factor. So, even though data-dependent algorithms (including DAWA) normally perform worse at large scales, as DAWA was proven to be scale-epsilon interchangeable, the very small ϵ could have canceled out the effect of large scale and allowed its data-dependence to help again. Furthermore, the main privatization strategy DAWA employs is to use workload-awareness and data-dependence to generate adaptive hierarchical histogram bins that fit the data at a given workload. Then it assumes uniformity within each bin to privatize. The relative accuracy for this technique depends on the exact context it’s run in and comparing DAWA’s privatization in Figure 3 versus in Figure 7 we see that, at the lowest ϵ , the coarse bins in session-start time matches allow DAWA to avoid the sharp spikes that other algorithms show at low ϵ . But in the service-time distribution, the exponential nature of the underlying data makes the uniformity assumption less valid.

It’s also interesting to note how well the Identity workload, which simply adds Laplacian noise to each histogram bucket, does on both data-sets. For almost all ϵ (except for DAWA at the smallest ϵ), the Identity algorithm has the least error. DPBench also notes how Identity can perform better over larger data-sets but it is striking how the complexity of other algorithms are not always justified when compared to simpler algorithms which can consistently perform better. It also points to the possibility for theoretical advances in differential privacy which could allow for better baseline error rates.

Finally all this complexity demonstrates the difficulty for a data-analyst to *a priori* decide which DP algorithm to use, as it can depend on subtle interactions between the required privacy stringency, the underlying data-set, and desired workload.

3.5 Transitioning To Trajectory Queries

Along with count query statistics, we were also interested in trajectory queries, so we can understand the network occupancy over time. One thing to note is that our trajectory analysis focuses on the granularity of *buildings* not individual AP's. We made this decision because the building-level data is at a more useful scale (131 buildings vs 4179 AP's), gives less exponential branching factors, and allows us to more easily examine whole-campus trajectories.

3.6 GIS Data Integration

First we wanted to visualize the campus movement in a model-free manner to see if we could glean any insights for future analysis. To do so, we used University of Massachusetts-Amherst's comprehensive GIS data [29] in order to convert the building ID's to latitude/longitude. Then using the EsriTM's World Street Map basemap [42] we could visualize trajectories across campus.

We generated multiple visualizations: Figure 11 visualizes all the building transitions en-mass, showing there are both many self-transitions and wide-reaching single-hop transitions. It also agrees with the intuition that many of the most popular transitions would cluster around nearby buildings, and within the center of campus.

But more than a static map of overall common transitions, we wanted to understand the rich information encapsulated in how these transitions change over time and across different buildings. To examine how the transitions change over time, we created videos where each frame was the transitions within a given timer period. Figure 12 exhibits three frames showing the different characteristics of campus mobility in the morning, afternoon, and evening.

In a similar vein, we were interested in what the transitions out of individual buildings looked like, exhibited in Figure 13. Some mobility models (e.g. [43]) work better when the single-hop transitions are somewhat adjacent. However, as shown in Figure 13 and consistent for the majority of buildings, almost all buildings have a one-hop transition to almost every other building and with non-trivial frequency. Some possible explanations for this include faster modes of transportation (e.g. a majority of students use the campus bus to commute around campus), poor Wi-Fi coverage in outdoor walkways, and intermittent connections from non-mobile devices such as laptops. This non-co-localization can make trajectory modeling difficult because it can lead to models predicting unrealistically far travel in overly short amounts of time.

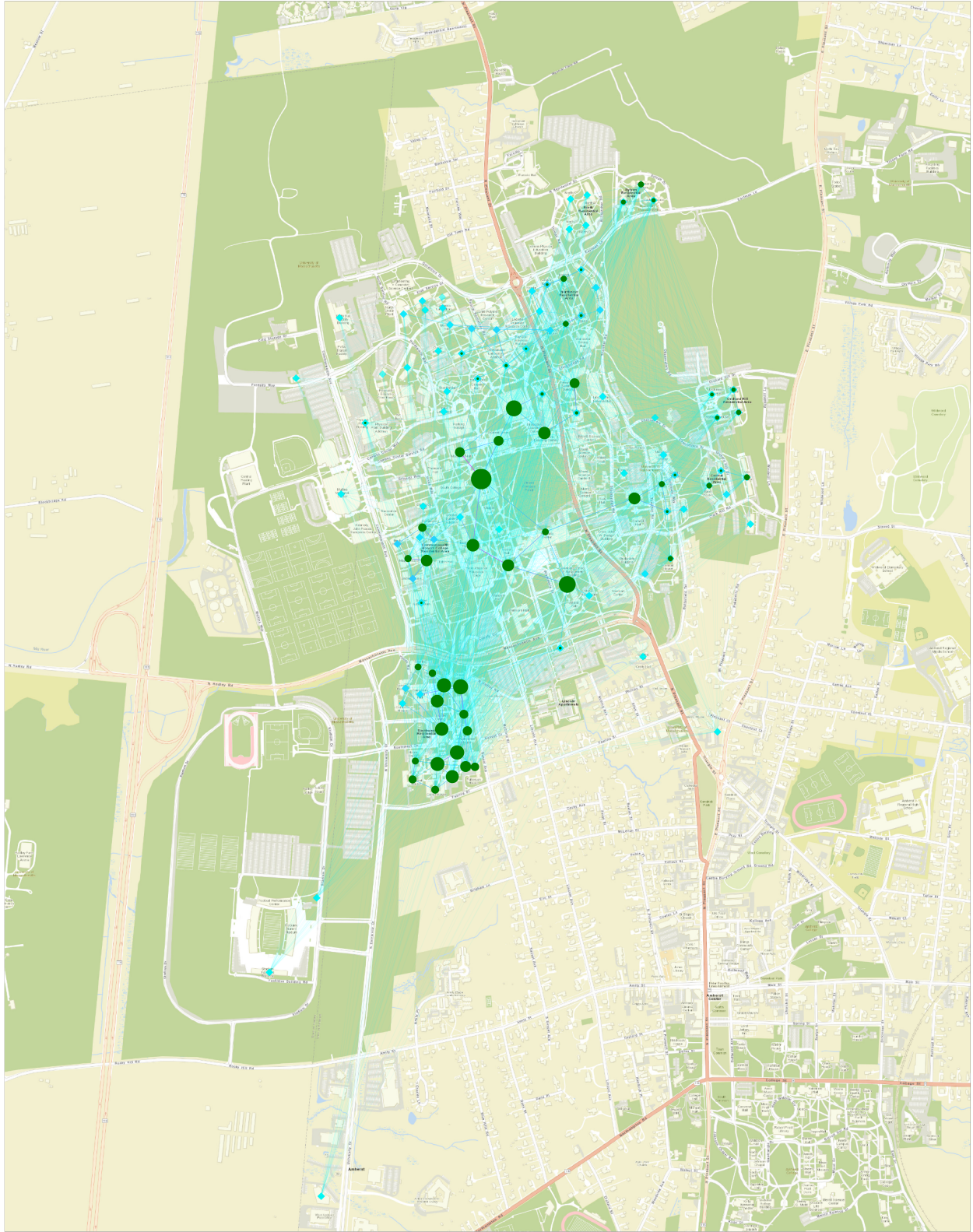


Figure 11: Visualization of all building transitions. The green circles represent self-transitions (within a single building to different AP's) while the blue lines represent single-hop building transitions. The size of the blue circle and opacity of the blue lines signify the relative number of transitions. Note how the campus has a dense network of transitions in the main part of campus and how many academic buildings (at the bottom) have a high number of self-transitions. Many of the buildings with large self-transitions in the center of campus are libraries and dining halls.

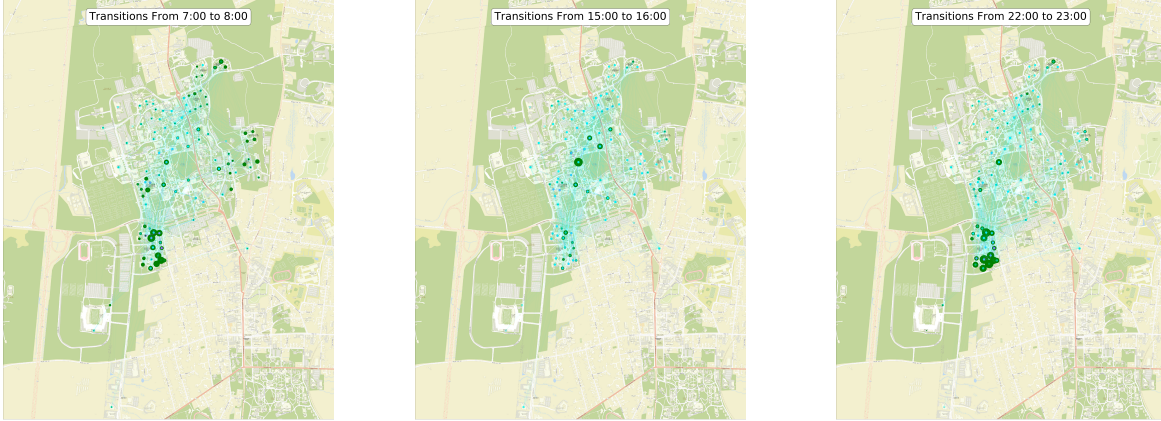


Figure 12: Visualizations of all building transitions from (a) 07:00-08:00, (b) 15:00-16:00, and (c) 22:00-23:00. Blue lines represent single-hop building transitions, red lines represent *non-existent* single-hop building transitions. Note how the campus moves from the dorms (at the bottom), to academic buildings and dining halls (at the middle), and then back to the dorms.

3.6.1 Transition Probability Matrix

With these caveats in mind, following [9], we constructed a transition probability matrix from our building-level transition (Figure 14) in a couple of variants to try to better understand the data. We see that most of the transitions are self-transitions (Figure 14 a,c) and that most buildings do not have a high-probability next-hop (Figure 14 d). In fact, as shown in Figure 14 b, there is a reasonably consistent ‘floor’ of transition counts across a wide range of buildings.

3.7 Privatizing Trajectory Queries

Given a large collection of trajectories, due to its high dimensionality and combinatorial complexity of trajectories, there is no immediately obvious way of privatizing them using typical count-based methods. So the goal of privatizing trajectory queries becomes a question of what intermediate data-structure should be used to represent them and to privatize. In other words, we want to first transform the unstructured set of trajectories into a arrangement which allows easier privatization, privatize this arrangement, and then transform back into an unstructured set of *privatized* trajectories which can then be released.

There are many examples in the literature of different intermediate data-structures designed for differentially private mobility analysis however many of them focus on privatizing GPS location data (e.g. [21]). For GPS mobility, one of the biggest difficulties is the continuous nature of GPS coordinates and how GPS trajectories can span many (hundreds of) miles. This is markedly different than campus network mobility



Figure 13: Visualizations of all building transitions originating from four buildings on campus. The green circles represent self-transitions (within a single building to different AP's) while the blue lines represent single-hop building transitions. The size of the blue circle and opacity of the blue lines signify the relative number of transitions. Note that there are very few non-existent single-hop transitions (red lines), even across large geographic distance. This is consistent with the majority of other buildings on campus as well.

analysis because we are constrained to a small number of buildings (131) and are only focused on a dense set of trajectories in a small area. In our research, we focused on intermediate representations based on variable-length n -grams.

3.7.1 NGRAM: Variable-Length N-Grams Trajectory Privatization

We investigated the use of Chen *et al.*'s NGRAM differentially private algorithm ([8]) on our data-set. It is designed for use with *sequential* data-sets (i.e. ordered lists of discrete items), which is applicable to our trajectories of discrete buildings. Note that our data-set is based on time-series mobility, so it encapsulates both the ordering and time of each association. A sequential intermediate representation purposefully ignores the temporal aspect of the data-set in order to better agglomerate similar transitions which occurred at different time-points. If we wanted to also privatize the temporal aspect of our data, a common approach is to model the temporal data separately by looking at overall temporal statistics such as mean sojourn time and mean transition frequency (hops per unit time) (e.g. [9]).

To give a high-level description of this algorithm, the NGRAM algorithm is based on n -grams, a probabilistic model designed for sequential data-sets which has found wide application in a variety of computational tasks. An n -gram model collects ordered lists of sequential items and analyzes them with the so-called ' $(n-1)$ Markov independence assumption', which states that the probability of each item occurring depends only on the previous $n - 1$ items. The advantage of such a simplifying assumption is it gives well-understood and tunable trade-offs on the size of look-back parameter n , as larger n gives higher accuracy but lower efficiency.

The main idea of NGRAM is to convert a sequential data-set into a set of n -grams, collate these grams into high-count sub-sequences of grams, and privatize the counts of these sub-sequences (using the Laplacian mechanism). Then these privatized grams can be published directly or post-processed into longer sequences.

In particular, NGRAM focuses on optimizing the length (number of hops) of these sub-grams. As the sub-grams get longer, they contain more sequential information but they also become combinatorially more rare. So, there will be fewer counts of longer sub-gram in the set of all the trajectories, and when we go to privatize this sub-gram count, the effect of the Laplacian noise on the relative error will be higher. *In other words, there is a fundamental trade-off where longer sub-grams carry more sequential information but have larger privatization error.* NGRAM tries to optimize for this trade-off by using *variable length* n -grams and not fixing the length of each sub-gram. Rather it uses an exploration tree, a variant of a prefix tree, to combine all trajectories into a variable-depth tree. In this tree, the root of the tree is an empty gram, the

first level is the counts of length-1 grams (i.e. the number of occurrences of each *single* building in all the trajectories), the second level is the counts of length-2 grams (i.e. the number of occurrences of each *pair* of buildings in all the trajectories), and so on.

In order to privatize such a tree, we allocate a portion of the privacy budget ε to privatize the counts at each successive level of the tree. So if we try to privatize the counts for a rare long gram, which is will be towards the bottom of the tree, then the noise will overwhelm its small count. Instead, in order to privatize a rare prefix, we want to limit its length to be shorter and therefore higher up in the tree. Then we can allocate more privacy budget towards its rare count and therefore add less noise, preserving more accuracy. Similarly, for popular prefixes, its larger count can ‘withstand’ a smaller privacy budget without losing too much accuracy, and so we can expand its length further down the tree. This has the added benefit of adding more sequential information for the more popular trajectories, which are the most useful for mobility models.

The way that NGRAM adaptively limits the length of grams, to follow this insight, is to stop traversing a subtree of exploration tree when its nodes are below some threshold. By doing so the tree will be jagged (have ‘variable-length root-leaf paths’) as more popular prefixes will lead to sub-trees which will take longer to get below the cutoff count.

3.8 NGRAM Trajectory Query Results

To compare to the ground-truth transitions, we first compared the one-hop transition probability matrices from ground-truth data in Figure 14 to the privatized data in Figure 15. Looking at the log plots (part (b)), increased number of black cells and sparser rows shows how NGRAM pruned many of the less-popular trajectories. But the interspersed conserved rows show how it also focused on keeping the trajectories of certain highly used buildings entirely intact. Additionally, comparing the non-identity percent transitions (part (d)), we see that there is a drastic reduction in outlier trajectories between the ground-truth and privatized data-sets.

We also wanted to understand and quantify how NGRAM reconstructed (counts of) entire trajectories, not just next-hop transitions. This is in itself difficult because similarity metrics for comparing millions of trajectories are very computationally expensive and can be a challenge to interpret. Instead, we focused on quantifying whether NGRAM successfully reconstructed the *existence* of paths and did not compare the relative count of these trajectories. This allowed us to generate Venn diagram visualizations like Figure 16 which looks at sequences of length one to four and whether they exist *only* in our ground-truth data (so

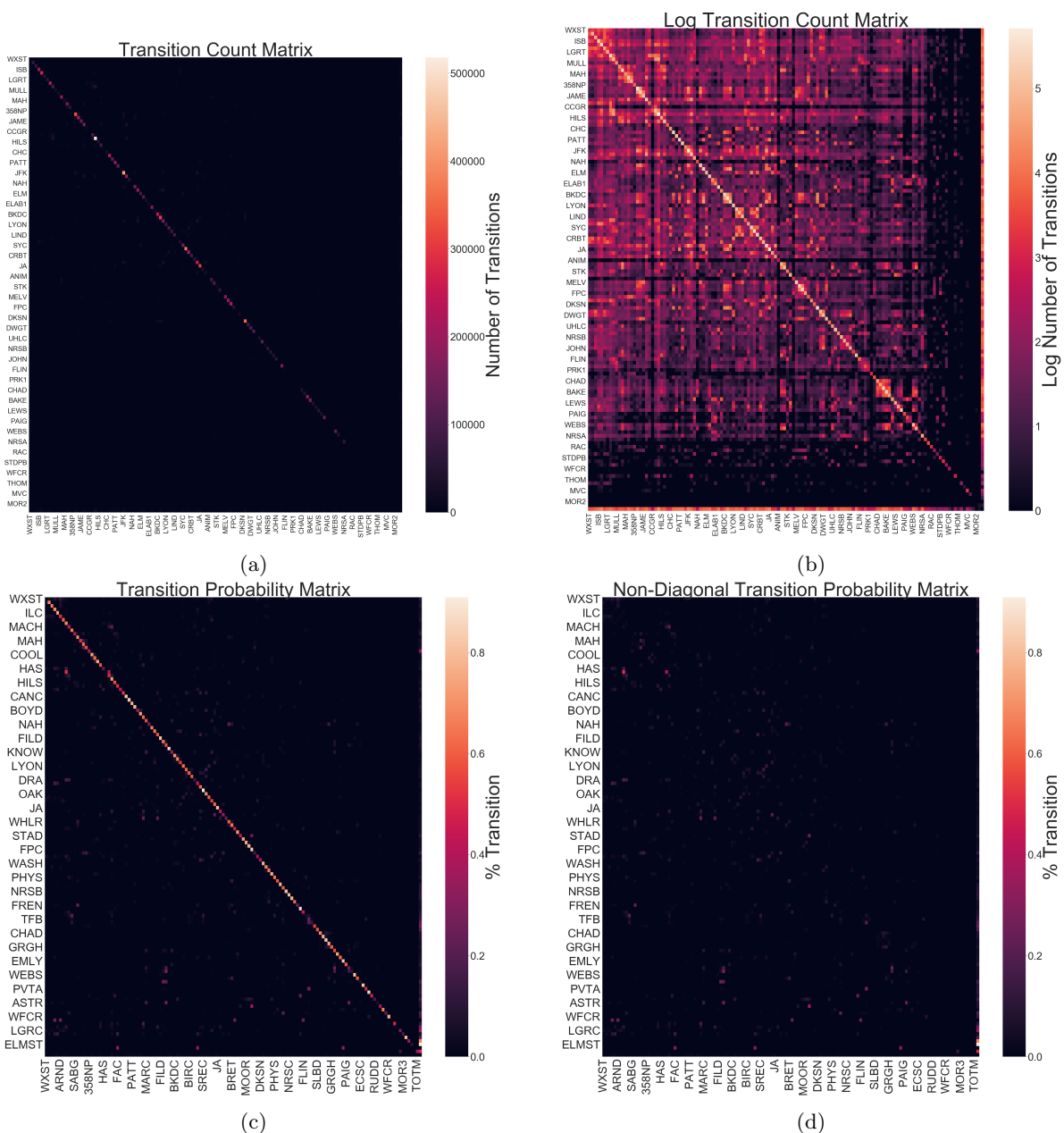


Figure 14: Visualizations of the one-hop building transition matrix. Part (a) gives the raw counts of one-hop transitions, (b) is the log transform of (a). Part (c) normalizes each row by the total row-count and part (d) is the same normalization but ignoring the contribution of the diagonal self-transitions. Note in the log-plot (b), how there is a ‘floor’ of low-count transitions among the majority of the buildings on campus. Also note in (d) how some buildings have strong next-hop associations with a few other buildings but that the majority of transitions are self-transitions (c).

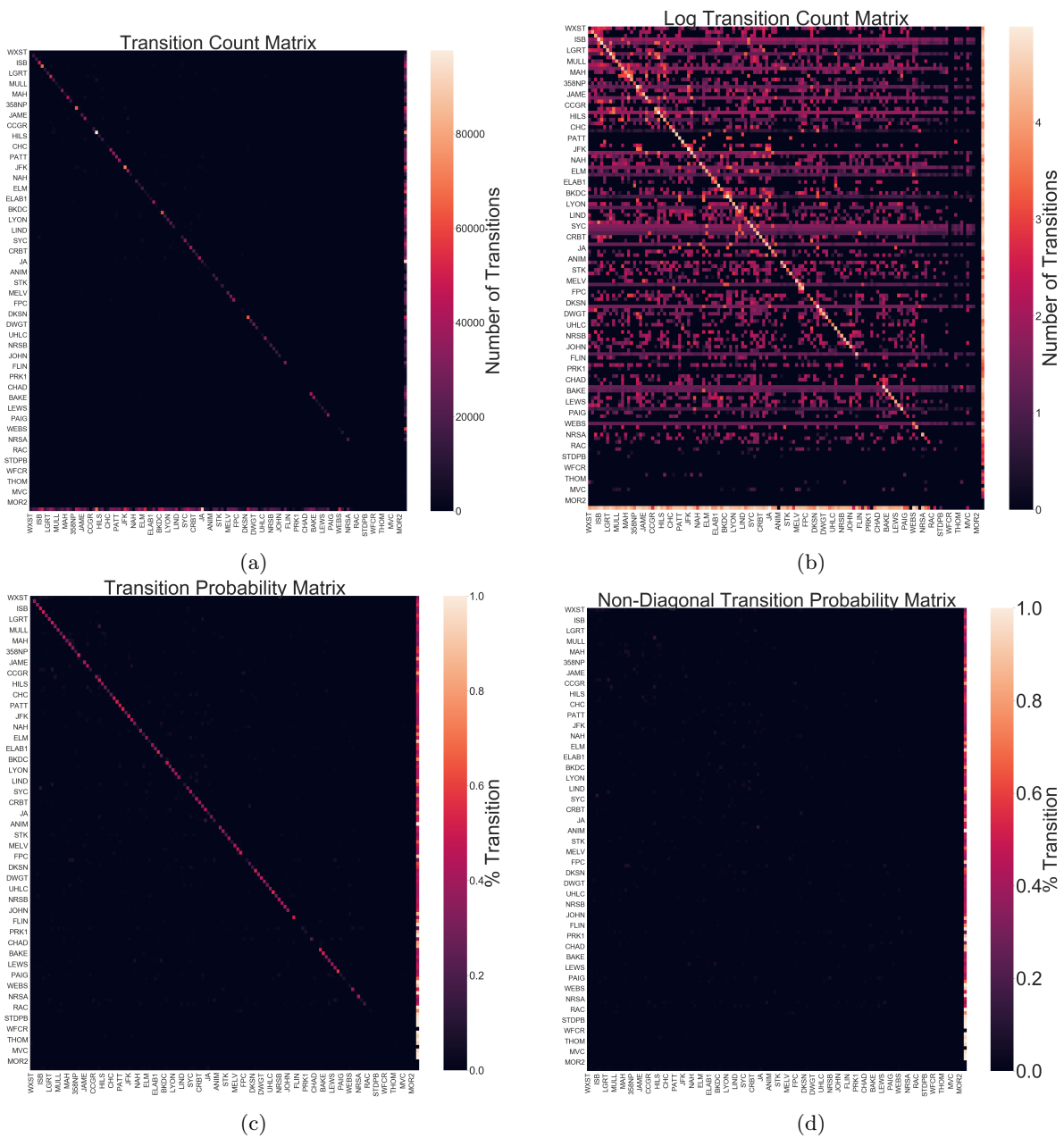


Figure 15: Visualizations of the *privatized* one-hop building transition matrix. We used the NGRAM privatization algorithm with parameters $n_{\max} = 5$, $\ell_{\max} = 10$, and $\varepsilon = 0.1$. See [8].

Part (a) gives the raw counts of one-hop transitions, (b) is the log transform of (a). Part (c) normalizes each row by the total row-count and part (d) is the same normalization but ignoring the contribution of the diagonal self-transitions. Compared to Figure 14, note the gaps in low-count next-hop transitions (b) and the elimination of many of the outlier non-identity transitions (d).

were not privatized), *only* in our privatized data (so are fake paths falsely generated), or if they are shared by both ground-truth and privatized data (so were successfully privatized). We see that there is relatively good agreement at 2-hop trajectories but even at modest 4-hop trajectories, NGRAM still has many fake and non-privatized paths, across different privacy budgets ϵ .

We were also curious at the effect of different privacy budget on the trajectories generated by NGRAM. We see that a larger ϵ , leads to less fake paths generated but does not give a greater number of real paths privatized. This could be because it allows the exploration tree to be explored further before it reaches the count cutoff, allowing greater coverage of the combinatorial possibilities, but also allows more errant non-existent paths to be manufactured as well. In summary found the n -gram approach with increasing trajectory length creates noisy trajectories suffers from significant false-positives, or fake trajectories. We postulate that this is because, n -gram makes constraints about the length of the trajectory it privatizes over. We also don't have DP algorithms that provide bounds for trajectory data, making it hard to determine how close to "optimal" it is.



Figure 16: Venn diagram comparing the existence of sequences from one to four hop length in the ground-truth vs. NGRAM privatized trajectories for privacy budget $\epsilon = 0.1, 1$. Note how longer sequences have worse coverage by the privatized queries and have more false-positive fake paths. Also note how increased privacy budget leads to fewer false-positive paths but not necessarily more real paths correctly privatized.

3.8.1 Trajectory Query Discussion

Privatizing trajectory queries is a greater challenge compared to count queries. For one, count queries are better studied and have an identical histogram data structure, which allows for easier empirical analysis of many algorithms. Additionally, there are well-defined error metrics which allow easy quantification of error for different algorithms. Using differential privacy to privatize trajectories is less well studied (especially for mobility data with a discrete set of locations) and its complexity makes algorithm design and error analysis more difficult. We examined the use of the NGRAM differentially private algorithm [8], which is designed for discrete sequential data and which uses adaptive n -grams in order to better collate sub-trajectories. We found that its privatization preserves the somewhat next-hop transition matrix, although we definite artifacts from dropping low-count trajectories. Then, when examining the existence of un-privatized trajectories, false-positive fake trajectories, and correctly privatized trajectories, we found rapidly decreasing success at even moderately-sized (≈ 4 hops) trajectories, across multiple ϵ parameters. Because of the exponential number of possible trajectories at a given length, this was to be expected. One feature of the data-set we had hoped to capitalize on was that the local neighborhood of next-hop transitions would be geographically constrained around each building, which would allow us to locally limit the exponential nature of trajectories. However, the GIS data showed that there was no such ‘local neighborhood’ as the majority of buildings had a next-hop trajectories to almost all other buildings.

Finally, it is worth noting that neither of these error metrics capture the full richness of time-series data; they don’t include the temporal aspect at all and don’t compare the relative similarity of trajectories. Further work into comparing similarity of large trajectory data-sets in a feasible amount of time would allow for more in-depth analysis of longer trajectories and a greater ability to compare multiple candidate algorithms.

4 Conclusion

Our goal was to take a large-scale Wi-Fi access point dataset from raw system logs and apply privatization techniques so that it could be used by the broader network research community with reasonable utility and quantifiable privacy guarantees. As the Internet becomes both more essential for every-day life, its use is exacerbated by unanticipated mobility and awe-inspiring scale. One promising way to cut this Gordian knot of complications is to use data-driven approaches derived from real-world data. These approaches can help network researchers better understand how the network is being used and design novel solutions. But, using

real-world data is a potential achilles' heel because the data can encode private information, unsuitable for public release.

Our solution is to use differential privacy, a cutting-edge privatization technique which offers tunable guarantees for quantifiable privacy vs. utility trade-offs, and is actively being researched and adopted. It has many theoretical advantages and a wide variety of conforming algorithms but it is not obvious *a priori* which algorithm works best with our data-set. We decided to split our empirical analysis of how different DP algorithms performed on our data-set into two sets of queries: count queries and trajectory queries. Count queries are useful to gain a snapshot understanding of well-known summary statistics for average network mobility while trajectory queries are useful for giving more flexible, and insights into causal relationships and campus mobility over time.

We found that our privatization of count queries successfully captured fine-grained detail for two example count query workloads: service time and session-start time histograms. Two data-independent algorithms, Identity and GreedyH, had the least squared error against ground-truth for both workloads across a range of the privacy budgets ϵ . However there was still a good deal of complexity in understanding the contributions of different factors (e.g. the ϵ parameter, workload, and data-set) on the algorithms' performance. To maintain as much privacy as possible, future data-holders should not empirically test algorithms on their ground-truth data-set because the algorithm choice itself could release unwanted private information. Our results give one avenue for understanding what algorithms work for mobility time-series data, specifically, but further research is necessary to translate theoretical analysis of DP algorithms into practical *a priori* decisions for which algorithm to use.

For trajectory queries, achieving a good trade-off between privacy and accuracy was more difficult due to the inherent high dimensionality and combinatorial complexity of trajectories. We evaluated one algorithm, NGRAM [8], which is designed for sequential data-sets like mobility data and found moderate success along with many directions for future study. It captured many of the more prominent single-hop transitions within the trajectories but had some difficulty accurately reconstructing longer trajectories, even with relaxed privacy budgets. While these metrics are two examples of methods to compare ground-truth against privatized trajectories, neither captures the full complexity inherent in these trajectories. Future work into better quantifying the trajectory query error would allow easier comparison between trajectory privatization methods.

Likewise, our work also leads to a rich set of future directions. Further work into sequential data-set privatization, in general, and time-series mobility data, in particular, could help decrease error for trajectory

queries. Time-series mobility data is, by its very nature, particularly difficult to privatize because both the temporal and sequential aspects of its data are highly unique. The vast exponential possibilities of unique trajectories (even at modest lengths) means that the counts of a given trajectories will be small, anathema to differential privacy’s goal of reducing the uniqueness of any individual row, which leads to high error rates. Similarly, the monotonic nature of timestamps means that time-stamped trajectories must be aggregated in order to be non-unique, but doing so in an accurate way remains an open question. The repetitive schedule of college campus schedules offers one possible avenue of aggregation but more general aggregation methods are necessary for other less-regimented mobility settings.

Finally, even after choosing an algorithm, using a DP algorithm to publicly release data requires choosing an ‘appropriate’ value for the privacy budget ϵ . The 2020 U.S. Census’ high profile use of differential privacy to privatize their data has been a major instigator for differential privacy adoption and more and more businesses are starting to adopt DP for use with their sensitive data (e.g. [5, 18]). The benefit of differential privacy is that it ultimately becomes a quantifiable question of societal mores and the public’s expectations for what is a good balance between social good and personal privacy.

5 Acknowledgments

I want to thank Vasanta Chaganti for being the quintessential mentor – encouraging, supporting, challenging, and kind. This research was everything I hoped it would be.

I also want to thank my research partner, Gregory Lee, for calmly looking into the heart of a problem, and solving it with panache. It turns out elegant solutions actually are better (sometimes).

Finally, I want to thank my family and friends, who nodded thoughtfully at my twisting meandering ramblings and understood what I actually meant, even if it was all a muddle at the time.

References

- [1] John M Abowd. “Staring-Down the Database Reconstruction Theorem”. In: *Joint Statistical Meetings, Vancouver, BC*. 2018.

- [2] GUNES Acar. *Four cents to deanonymize: Companies reverse hashed email addresses*. <https://freedom-to-tinker.com/2018/04/09/four-cents-to-deanonymize-companies-reverse-hashed-email-addresses/>. 2019.
- [3] Daily Free Press Admin. *BU protester fined, could face jail time*. <https://dailyfreepress.com/2007/12/06/bu-protester-fined-could-face-jail-time/>. 2007.
- [4] M. Afanasyev et al. “Usage Patterns in an Urban WiFi Network”. In: *IEEE/ACM Transactions on Networking* 18.5 (2010), pp. 1359–1372.
- [5] Apple. *Differential Privacy*. https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf. 2019.
- [6] UN General Assembly. “Universal declaration of human rights”. In: *UN General Assembly* 302.2 (1948).
- [7] Louis Brandeis and Samuel Warren. “The right to privacy”. In: *Harvard law review* 4.5 (1890), pp. 193–220.
- [8] Rui Chen, Gergely Acs, and Claude Castelluccia. “Differentially private sequential data publication via variable-length n-grams”. In: *Proceedings of the 2012 ACM conference on Computer and communications security*. 2012, pp. 638–649.
- [9] Yung-Chih Chen, Jim Kurose, and Don Towsley. “A mixed queueing network model of mobility in a campus wireless network”. In: *2012 Proceedings IEEE INFOCOM*. IEEE. 2012, pp. 2656–2660.
- [10] Cisco. *Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper*. <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html>. 2019.
- [11] David Clark. “The design philosophy of the DARPA Internet protocols”. In: *Symposium proceedings on Communications architectures and protocols*. 1988, pp. 106–114.
- [12] Yves-Alexandre De Montjoye et al. “Unique in the crowd: The privacy bounds of human mobility”. In: *Scientific reports* 3 (2013), p. 1376.
- [13] Irit Dinur and Kobbi Nissim. “Revealing information while preserving privacy”. In: *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 2003, pp. 202–210.
- [14] Gautam Divgi and Edward Chlebus. “Characterization of user activity and traffic in a commercial nationwide Wi-Fi hotspot network: global and individual metrics”. In: *Wireless networks* 19.7 (2013), pp. 1783–1805.

- [15] Cynthia Dwork, Aaron Roth, et al. “The algorithmic foundations of differential privacy”. In: *Foundations and Trends in Theoretical Computer Science* 9.3–4 (2014), pp. 211–407.
- [16] Cynthia Dwork et al. “Calibrating noise to sensitivity in private data analysis”. In: *Theory of cryptography conference*. Springer. 2006, pp. 265–284.
- [17] Nick Feamster, Jennifer Rexford, and Ellen Zegura. “The road to SDN: an intellectual history of programmable networks”. In: *ACM SIGCOMM Computer Communication Review* 44.2 (2014), pp. 87–98.
- [18] Google. *Differential Privacy*. <https://github.com/google/differential-privacy>. 2019.
- [19] Moritz Hardt, Katrina Ligett, and Frank McSherry. “A simple and practical algorithm for differentially private data release”. In: *Advances in Neural Information Processing Systems*. 2012, pp. 2339–2347.
- [20] Michael Hay et al. “Exploring privacy-accuracy tradeoffs using dpcomp”. In: *Proceedings of the 2016 International Conference on Management of Data*. 2016, pp. 2101–2104.
- [21] Xi He et al. “DPT: differentially private trajectory synthesis using hierarchical reference systems”. In: *Proceedings of the VLDB Endowment* 8.11 (2015), pp. 1154–1165.
- [22] U.S. Department of Health & Human Services. *Guidance Regarding Methods for De-identification of Protected Health Information in Accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule*. <https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html>. 2015.
- [23] Tristan Henderson, David Kotz, and Ilya Abyzov. “The changing usage of a mature campus-wide wireless network”. In: *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM. 2004, pp. 187–201.
- [24] Troy Hunt. *Have I Been Pwned?* <https://haveibeenpwned.com/>. 2020.
- [25] David Kotz and Kobby Essien. “Analysis of a campus-wide wireless network”. In: *Wireless Networks* 11.1-2 (2005), pp. 115–133.
- [26] David Kotz et al. *CRAWDAD dataset dartmouth/campus (v. 2009-09-09)*. Downloaded from <https://crawdad.org/dartmouth/campus/20090909>. Sept. 2009. DOI: 10.15783/C7F59T.
- [27] Chao Li et al. “A data-and workload-aware algorithm for range queries under differential privacy”. In: *Proceedings of the VLDB Endowment* 7.5 (2014), pp. 341–352.
- [28] Ashwin Machanavajjhala, Xi He, and Michael Hay. “Differential privacy in the wild: A tutorial on current practices & open challenges”. In: *Proceedings of the 2017 ACM International Conference on Management of Data*. 2017, pp. 1727–1730.

- [29] University of Massachusetts-Amherst. *University of Massachusetts-Amherst GIS Data*. https://www.umass.edu/oir/sites/default/files/publications/glance/FS_gla_01.pdf. 2020.
- [30] Frank D McSherry. “Privacy integrated queries: an extensible platform for privacy-preserving data analysis”. In: *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. 2009, pp. 19–30.
- [31] University of Michigan. *First-day WiFi traffic triples*. <https://michigan.it.umich.edu/news/2017/09/20/wifi-traffic-triples/>. 2017.
- [32] Ilya Mironov. “On significance of the least significant bits for differential privacy”. In: *Proceedings of the 2012 ACM conference on Computer and communications security*. 2012, pp. 650–661.
- [33] Paul Ohm. “Broken promises of privacy: Responding to the surprising failure of anonymization”. In: *UCLA l. Rev.* 57 (2009), p. 1701.
- [34] Ljubica Pajevic, Gunnar Karlsson, and Viktoria Fodor. *CRAWDAD dataset kth/campus (v. 2019-07-01)*. Downloaded from <https://crawdad.org/kth/campus/20190701>. July 2019. DOI: 10.15783/c7-5r6x-4b46.
- [35] Pittsburgh Post-Gazette. *New Pennsylvania law allows school districts to record student conversations on buses*. <https://www.post-gazette.com/news/education/2014/06/12/New-Pennsylvania-law-allows-school-districts-to-record-student-conversations-on-buses/stories/201406120127>. 2014.
- [36] Wahbeh Qardaji, Weining Yang, and Ninghui Li. “Understanding hierarchical methods for differentially private histograms”. In: *Proceedings of the VLDB Endowment* 6.14 (2013), pp. 1954–1965.
- [37] Philipp Richter et al. “A multi-perspective analysis of carrier-grade NAT deployment”. In: *Proceedings of the 2016 Internet Measurement Conference*. 2016, pp. 215–229.
- [38] Libo Song et al. “Predictability of WLAN mobility and its effects on bandwidth provisioning”. In: (2006).
- [39] Jennie Steshenko, Vasanta G Chaganti, and James Kurose. “Mobility in a large-scale WiFi network: from syslog events to mobile user sessions”. In: *Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*. 2014, pp. 331–334.
- [40] Latanya Sweeney. “Simple demographics often identify people uniquely”. In: *Health (San Francisco)* 671 (2000), pp. 1–34.
- [41] Amherst University of Massachusetts. *University of Massachusetts, Amherst – At a Glance*. https://www.umass.edu/oir/sites/default/files/publications/glance/FS_gla_01.pdf. 2019.

- [42] *World Street Map*. http://server.arcgisonline.com/ArcGIS/rest/services/World_Street_Map/MapServer/export?bbox=-8075783.77909,5187611.83338,-8071776.27742,5192717.29657&bboxSR=3395&imageSR=3395&size=2000,2547&dpi=96&format=png32&transparent=true&f=image. 2020.
- [43] Jungkeun Yoon et al. “Building realistic mobility models from coarse-grained traces”. In: *Proceedings of the 4th international conference on Mobile systems, applications and services*. 2006, pp. 177–190.
- [44] Dan Zhang et al. “Ektelo: A framework for defining differentially-private computations”. In: *Proceedings of the 2018 International Conference on Management of Data*. 2018, pp. 115–130.