Facebook Privacy-Protected Full URLs Data Set

Solomon Messing Christina DeGregorio Bennett Hillenbrand Gary King
Saurav Mahanti Zagreb Mukerjee Chaya Nayak Nathaniel Persily
Bogdan State Arjun Wilkins

April 12, 2022

This is version 9 of a dataset that results from an ongoing collaboration between Facebook and Social Science One. It describes the dataset's scope, structure, fields, and privacy-preserving characteristics. Version 1 was described at https://socialscience.one/blog/update-social-science-one.

Citation

Messing, Solomon; DeGregorio, Christina; Hillenbrand, Bennett; King, Gary; Mahanti, Saurav; Mukerjee, Zagreb; Nayak, Chaya; Persily, Nate; State, Bogdan; Wilkins, Arjun, 2021, "Facebook Privacy-Protected Full URLs Data Set", https://doi.org/10.7910/DVN/TDOAPG, Harvard Dataverse, V8.

Data Access

To obtain access to this data, see the Request for Proposals process at https://socialscience.one/rfps. No other means of access is allowed.

Summary

This document details a data set designed to allow researchers to study the distribution of URLs on Facebook and how users interacted with them. We've protected this data set using differential privacy, which adds enough noise to the data to provide precise guarantees that no significant additional information can be learned from the data about individuals (beyond what is already available from any external source). That means that while no one can learn anything significant about the individuals from the data set (including whether they are in the dataset or not), researchers can still use the data to uncover broad time series or group-level trends and relationships of interest.

The current version of this dataset summarizes the demographics of people who viewed, shared, and otherwise interacted with web pages (URLs) shared on Facebook starting January 1, 2017 up to and including December 31, 2021. The dataset is updated to add new months of data approximately every three months, so the date ranges listed as included are current as of the date listed at the top of the first page. URLs are included if shared (as an original post or reshare) with "public" privacy settings more than 100 times (plus Laplace(5) noise to minimize information leakage). The URLs have been canonicalized (standardized) and processed (as detailed below) to remove potentially private and/or sensitive data. Aggregate data on user actions on URLs is provided for URLs shared publicly and those shared under the "share to friends" privacy setting.

These dataset was collected by logging actions taken on Facebook. Logs were processed using a combination of Hive, Presto, and Spark using the Dataswarm job-execution framework. To construct this dataset, we processed approximately an exabyte of raw data from the Facebook platform – including more than 50 terabytes per day of interaction metrics and more than 1 petabyte per day of exposure data (views). The data set has about 63 million URLs, nearly 2.4 trillion rows, and over 55 trillion cell values.[1]

Data from users who have chosen to delete their accounts are not represented in our dataset due to legal constraints, which may have a larger impact on URLs that were shared further in the past (this release is aggregated to provide month-year breakdowns). Users who "deactivate" but do not delete their accounts remain in our dataset.

We have taken measures to remove URLs and associated engagement statistics that link to known child exploitative imagery from these data. We have also taken measures to remove URLs, "Title", and "Blurb" for known non-consensual intimate imagery, suicide and self-harm, although the associated engagement statistics with these links remain in the data.

This dataset includes posts from users that have been taken down due to Community Standards violations. To learn how Facebook defines and measures key issues, refer to the Community Standards Enforcement Report. The numbers cited in this report are not comparable to the data presented in this RFP as they reference different underlying data. For additional information, see: Community Standards, Enforcement Report Guide.

Infrastructure and Resources. Facebook is providing researchers with accepted proposals access to a system that provides an interface to query these data. The interface is a simple SQL layer that will provide access to the tables documented below.

Warnings

1. Privacy-protective procedures have been applied to this dataset (by adding noise in specific ways described below and recoding counts of actions to one when the same user took multiple actions on the same URL). From a statistical perspective, these adjustments induce measurement error, which biases statistical results.[2] Estimates that ignore the error may also induce incorrect coverage for standard errors and thus confidence intervals (in either direction). Researchers at Social Science One are developing unbiased statistical approaches for analyzing these data [see Evans et al., 2021a, Evans and King, 2021a,b, Evans et al., 2021b].

2. We do not adjust the privacy-protected data for consistency with other information, as for example the U.S. Census does in its invariants or its post-processing of its differentially private releases. As is true any time data contain noise, some values in DP-protected counts will be too large and some too small, and some will be negative, outside the range of the data. However, we are able to make all privacy protective procedures public, which means that researchers can correct for statistical biases. The data may also have other *apparent* inconsistencies, such as fewer views than clicks for some URLs. Appropriate statistical procedures should be used to analyze these data; simplistic approaches like censoring negative values to zero can severely bias statistical estimates.

3. With any large-scale data project, we expect to learn about issues related to data quality, validity, fidelity, etc. in the course of conducting analyses. This is especially the case here, as this is one of the

[1] We obtain the number of cell values by multiplying the number of rows times the number of action and URL data columns, which is 23. This includes 14 columns with differential privacy protections applied, and (1) Clean_url, (2) Parent_domain, (3) Full_domain, (4) first_post_time, (5) Share_title, (6) Share_main_blurb, (7) tpfc_rating, (8) tpfc_first_fact_check, (9) public_shares_top_country.

[2] The most common bivariate situation is wherein noise biases effects toward zero. However, depending on the quantity of interest, this error may bias estimates in either direction, and in some cases results in estimates with the incorrect sign.

largest social science research datasets ever constructed. Expect issues to arise and let us know what you learn by contacting researchtool-help@fb.com. We hope to respond with fixes to errors fast, where feasible.

4. These data are considerably larger than commonly used social science datasets, and so will not fit into system memory. Researchers should plan analyses in ways that efficiently use available memory, without exhausting the resources of our computing cluster. At present, this includes limiting system RAM to that of a modern server (e.g., with around 64GB RAM). SQL, or something like it, is required to access these data; Python, Linux, and R will be helpful.

5. The dataset includes 46 countries. If you need data for your research from a country not included here, please let us know. Once we get through the second round of grantee countries, Facebook is committed to adding countries to the dataset within one month. Be aware, however, that small countries, and those with a small number of Facebook users, are unlikely to generate informative datasets given the privacy-protective procedures described below.

Data

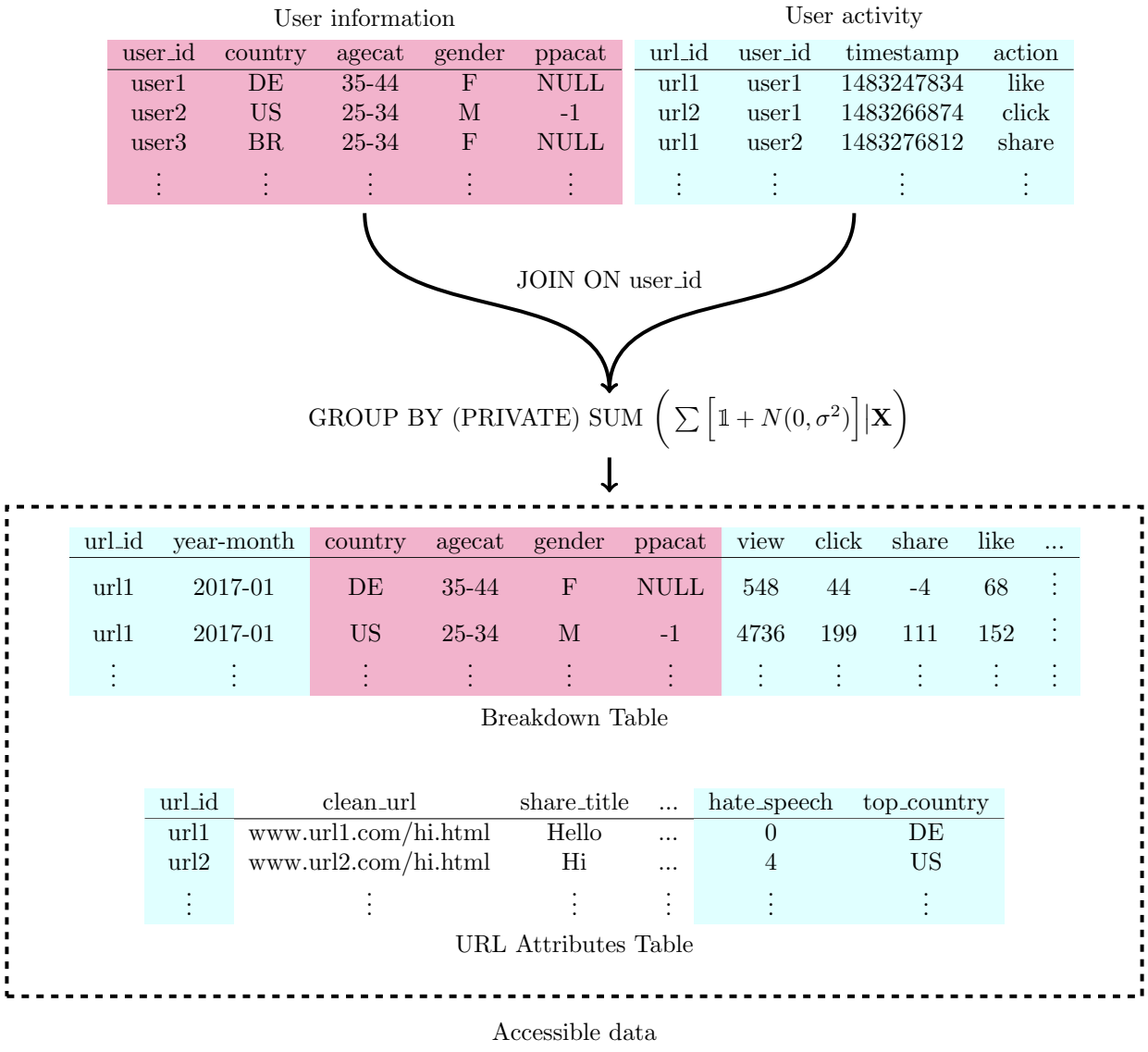
Data Collection and Structure. We include three tables in this release: the *URL Attributes Table*, *Breakdown Table*, and *User Reports Table*.

The URL Attributes Table contains the URL-level information that was included in the URLs light release: the URL, domain, timestamp, webpage title and blurb, spam, false news, and hate speech flags, and the country in which it was shared most frequently (after implementing privacy protective procedures). This table includes all URLs "shared publicly" by 100 users (+Laplace(5) random noise) somewhere in the world. "Shared publicly" means that each user chose the public option on their Facebook privacy settings. The table contains 63,574,836 URLs.

The Breakdown Table describes counts of users who take particular actions (e.g., views, likes, comments) within URL-year-month-country-age-gender "buckets," for all but the US. This includes 60 year-months, 45 countries, 7 age groups, and 3 gender groups, representing a number of buckets equal to 56,700 times the number of URLs — which is a total of approximately 2.1 trillion (more precisely, 2,115,561,215,565) rows. This number excludes structural zeros (URLs that haven't yet been shared in a given year-month). For the US, we have 60 year-months, 7 age groups, 3 gender groups, and also have a political page-affinity variable of 6 categories, and so the total number of buckets is 7,560 times the number of URLs — which is approximately 282 billion, or more precisely 282,074,828,742 rows. As more countries are added into the dataset, the numbers in this paragraph will change, but the numbers represented reflect the dataset with 46 countries included.

The 2,397,636,044,307 rows of the breakdown table are these buckets. Each unique user-URL-action is represented by the number 1, which is added to precisely one of the buckets. This means that if a user shares a URL more than once, only the first share is counted; this variable can be thought of as either the number of users who take a specific action or the total number of actions re-coded to one per user. This means that metrics related to unique user-URL-actions in these data will not match other data sources that provide counts of total shares, as some Facebook users can and do take the same action on a URL repeatedly.) We define a "user" throughout this codebook as a unique Facebook account (even though some people may have more than one Facebook account and some may share accounts).

This table was created by joining two tables of disaggregated data at Facebook: a user information table and a user-activity table (see figure below). The resulting user activity and demographic breakdowns table contains counts—to which Gaussian noise is added so as to provide differential privacy—of various actions broken down by country, age brackets, gender, and, in the U.S., political page affinity brackets (defined in additional detail below).



The User Reports table contains three variables recording the number of unique users who reported posts containing a particular URL as spam, false news, or hate speech. The data in the User Reports table is aggregated at the level of URL and year-month. These three user-reported variables were previously only aggregated at the level of URL (and included in the URL Attributes table), which complicated the process of adding more months of data with respect to privacy accounting. As of November 2020, these variables are now aggregated over both URL and year-month in this dedicated table. The URL Attributes table maintains the previously released values for the three user report variables (which cover a period from January 1, 2017 to July 31, 2019), but values of these variables in the Attributes table will not be updated in subsequent releases.

The "URL Attributes", "Breakdown", and "User Reports" tables can be joined using a "url_id" key. Aggregate counts have been protected by adding noise, as per the 'privacy protection' section below. Rows that contain 0 counts prior to adding noise have not be removed to ensure that we are not leaking information about rows containing 0 entries.

Variables Aggregate statistics in the breakdown table marked "DP" have noise added for differential privacy. An artificial example dataset with a few observations can be found at https://bit.ly/FullURLsEG; it may be helpful in understanding the fields described below.

URL Attributes Table

- **Url_rid** [**text**]: a unique URL id created specifically for this data set.
- **Clean_url** [**text**]: The webpage URL after processing. This is the full URL, not just the domain (e.g., https://www.nytimes.com/2018/07/09/world/asia/thailand-cave-rescue-live-updates.html). URLs that are no longer reachable will persist in the data. The URLs have been processed in an attempt to consolidate different web addresses that point to the same URL and to remove potentially private and/or sensitive data. Our post-processing procedure is explained below.
- **Parent_domain** [**text**]**:** parent domain name from the URL (eg. foxnews.com).
- **Full_domain** [**text**]**:** full domain name from the URL (eg. www.foxnews.com, video.foxnews.com, nation.foxnews.com, insider.foxnews.com).
- **First_post_time** [**timestamp**] - The date/time when URL was first posted by a user on Facebook. Date-times are truncated to 10 minute increments. The exact format is YYYY-MM-DD HH:MM:SS, for example: 2015-12-02 18:10:00.
- **First_post_time_unix** [**unix timestamp**] - The above field translated into unix time—the number of seconds since 1970-01-01 00:00:00, for example: 1449079800.
- **Share_title** [**text**]**:** Provided by the author of the URL's content, pulled from **og:title** field in original html if possible).
- **Share_main_blurb** [**text**]**:** Provided by the author of the URL's content (pulled from **og:description** field in original html if possible).
- **Tpfc_rating** [**text**]**:** If URL was sent to third-party fact-checkers (tpfc), did they rate it (NULL if not) and if so, how did they rate it? Category values include: 'True', 'False', 'Prank Generator' , 'False Headline or Mixture', 'Opinion', 'Satire', 'Not Eligible, 'Not Rated.' Definitions, and a list of fact checkers, are available here: https://www.facebook.com/help/publisher/182222309230722 and https://www.facebook.com/help/572838089565953?helpref=faq_content. More information on how news that may be false is selected can be found here: https://www.facebook.com/help/1952307158131536. Only available for some stories, only available in Argentina, Brazil, Cameroon, Canada, Colombia, Denmark, France, Germany, India, Indonesia, Ireland, Italy, Kenya, Mexico, Middle East and North Africa, Netherlands, Nigeria, Norway, Pakistan, Philippines, Senegal, South Africa, Sweden, Turkey, UK, US. When more than one rating is given to a story, we use Facebook's *precedence rules*, described below.
- **Tpfc_first_fact_check** [**timestamp**]**:** the date-time that article was first fact-checked, if at all. If the article has not been fact checked, this will be NULL. Date-times will be truncated to 10 minute increments. The exact format is YYYY-MM-DD HH:MM:SS, for example: 2015-12-02 18:10:00.
- **Tpfc_first_fact_check_unix** [**unix timestamp**] - The above field translated into unix time—the number of seconds since 1970-01-01 00:00:00, for example: 1449079800.
- **Spam_usr_feedback**** [**integer**]: the total number of unique users who reported posts containing the URL as spam over the period from January 1, 2017 to July 31, 2019. The User Reports Table contains monthly aggregations of the data in this column for months subsequent to July 2019.
- **False_news_usr_feedback**** [**integer**]: the total number of unique users who reported posts containing the URL as false news over the period from January 1, 2017 to July 31, 2019. The User Reports Table contains monthly aggregations of the data in this column for months subsequent to July 2019.

- **Hate_speech_usr_feedback**** **[integer]**: the total number of unique users who reported posts containing the URL as hate speech over the period from January 1, 2017 to July 31, 2019. The User Reports Table contains monthly aggregations of the data in this column for months subsequent to July 2019.
- **Public_shares_top_country** **[text]**: URL shares are tallied by country and the country with the most (differentially private) shares is provided as an ISO 3166-1 alpha-2 letter code. This field is *not* indicative of all locations where this article was posted—rather it is only the top country among users who shared it.

Breakdown Table

Three types of variables can be found here: the *keys* that define the row-units (as the Cartesian product of all keys); a *unique URL id*, which is a URL-level observation that uniquely identifies a URLs (and can be linked to from the URLs Attributes Table); and the *aggregate statistics*, which are summaries (e.g., sum or mean) within, i.e., conditioned on or grouped by each of the keys. The data contain aggregated counts of the number of users who have shared, viewed, clicked, liked (or otherwise reacted), or commented on each URL.

- **Url_rid** **[text]**: a unique URL id created specifically for this data set.
- **Keys**: These variables define the "buckets" which are the rows of this table.
  - **Year_month** **[string]:** year and month. Data will be partitioned on this variable.
  - **Country** **[text]**: the country in which the actions below occur. This variable is stored in a column called **c** in the actual dataset. Data will be partitioned on this variable and due to considerations related to the size of the data, for now we are prioritizing the release of all countries in researchers' proposals, though we are in the process of adding more. This release will contain data for countries needed to conduct analysis for research proposals already approved through Social Science One — Argentina, Australia, Brazil, Canada, Chile, Colombia, Hong Kong, India, Israel, Kenya, Mexico, Switzerland, Sierra Leone, Syria, Taiwan, the United States, the United Kingdom, Venezuela, Zimbabwe, and the 27 EU Member states[3] (Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Italy, Ireland, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, and Sweden).
  - **Age_bracket** **[text]:** Age brackets include: 18-24, 25-34, 35-44, 45-54, 55-64, 65+, NULL. Data from users' profiles.
  - **Gender** **[text]:** male, female, other. Data from users' profiles.
  - **Political_page_affinity** **[integer, ordered]:** These buckets [-2, -1, 0, 1, 2, NULL] take the pages people follow and scale them into components associated with political affinity, based on Barberá et al. [2015]. This variable is only available in US data. The "Political Page Affinity Model" section in this documentation includes more details about the estimation procedure for political page affinity.
- **Aggregate statistics:** these columns contain aggregated URL-user-actions, counts of the number of people who fall into the bucket corresponding to the cell of the cross-products of the key variables in question.
  - **Views** **[integer], DP:** Number of users who viewed a post containing the URL.

[3]There will be a European Union-level aggregation in the data, constructed by summing the country-counts across buckets after noise.

- **Clicks [integer], DP:** Number of users who clicked on the URL.
- **Shares [integer], DP:** Number of users who shared the URL in a post or reshared such a post.
- **Total_share_without_clicks, DP [integer]**: Number of users who shared a post containing the URL but did not actually click on the link. (Some users share articles without first clicking through to the actual content. Hence, this number may help identify articles that users are sharing without reading, or URLs used in organized campaigns to spread content.)
- **Comments [integer], DP:** Number of users who comment on posts containing the URL
- **Likes [integer], DP:** Number of users who liked posts containing the URL
- **Loves [integer], DP:** Number of users who 'love' posts containing the URL
- **Hahas [integer], DP:** Number of users who react 'haha' posts containing the URL
- **Wows [integer], DP:** Number of users who 'wow' posts containing the URL
- **Sorrys [integer], DP:** Number of users who 'sad' posts containing the URL. Note that the official name for this reaction is 'sad', but the column name in this dataset is 'sorrys.'[4]
- **Angers [integer], DP:** Number of users who 'angry' posts containing the URL

User Reports Table

- **Url_rid [text]**: a unique URL id created specifically for this data set.
- **Year_month [string]:** year and month. Data will be partitioned on this variable.
- **Spam_usr_feedback**** **[integer]**: the total number of unique users who reported posts containing the URL as spam.
- **False_news_usr_feedback**** **[integer]**: the total number of unique users who reported posts containing the URL as false news.
- **Hate_speech_usr_feedback**** **[integer]**: the total number of unique users who reported posts containing the URL as hate speech.

**Note about user feedback fields: these fields constitute information provided by users, which may not be indicative of actual violations of Facebook's Community Standards, and like any survey question or coding exercise, may not be a measure of the concept the researcher intends. For example, for the variable "total_hate_speech_usr_feedback", users may share URLs to endorse or oppose the content. Endorsements of hate speech violate Facebook's community standards policy, while opposing it does not. Users may also flag content as hate speech because they disagree with it (if they perceive the difference or can distinguish if they do), rather than to actually indicate hate speech, resulting in ambiguous or false positive reports of hate speech, if taken literally. Similar issues apply to other fields.

For "total_spam_usr_feedback", in contrast to URLs found to contain hate speech (which Facebook deliberately does not block due to the subtleties above), URLs containing content that violates spam policies are blocked from the platform for future sharing.

These data were originally collected or derived from operational information or data sources or otherwise — and not for research purposes. Features of the dataset may be inaccurate, incomplete, or have been collected in ways that are not compatible with research goals. Researchers need to adapt their methods, research designs, and quantities of interest to the data at hand. Please let us know if you see anything we might be able to adjust generically.

[4]For more details on Facebook Reactions, see the following post from the Facebook Newsroom.

Political Page Affinity Model

Following Barberá et al. [2015], our modeling approach assumes that users connections to political pages on Facebook are at least partially driven by perceived ideological affinity. Building on this assumption, it is possible to use scaling methods to estimate the political positions of political pages and users on a common latent space.

When computed using Twitter data, these estimates have been found to be highly predictive of partisan and ideological identities. Bond and Messing [2015] demonstrated that a similar scaling method applied to Facebook data yields equally valid estimates of political positions. This approach has also been extensively validated by the academic community, and estimates computed using this method have been used in multiple publications in top-tier journals (e.g. see Bail et al. [2018]; Brady et al. [2017]; Hassell et al. [2020]; Imai et al. [2016]; Tausanovitch and Warshaw [2017]).

Despite the high correlation between estimates derived using this method and self-reported political ideology from surveys, here we label our estimates "political page affinity." This reflects the fact that our approach measures people's connections to Pages with similar audiences as Pages representing politicians of known political affiliation/ideology, rather than a prediction of their self-reported political ideology, party affiliation, or policy preferences, for which an alternative model based on supervised learning methods might be more appropriate.

Our implementation as part of the "Facebook Privacy-Protected Full URLs Data Set" sought to replicate the original methodology from Barberá et al. [2015] – whose open-source code is available here – as closely as possible. It can be summarized in the following three steps:

1. Create an adjacency matrix with users as rows and political pages as columns.
   (a) Only monthly active users 18 years old and above who are based in the US are included in the dataset.
   (b) The full set of political pages we considered includes:
      i. All sitting Members of Congress
      ii. All parties in Congress (i.e. Democrats, GOP, HouseDemocrats, USSenateDemocrats, SenateGOP, HouseRepublicans)
      iii. Four leading candidates in the 2016 presidential primary election
      iv. The top ten candidates in the 2020 Democratic presidential primary election
      v. The 36 main news outlets in the United States, as selected by the Pew Research Center here
      vi. The hosts of all prime time cable news shows (from 8pm to 11pm). When multiple accounts existed, we selected the one with the most fans.
   (c) Each cell indicates whether the user in the row liked the page in the column. To follow the methodology described in Barberá et al. [2015] as closely as possible, here we do not consider additional signals, such as engagement with posts created by the pages or user-level covariates.
2. Run correspondence analysis [Greenacre, 1993] on a subset of this adjacency matrix.
   (a) As in Barberá et al. [2015], we split our estimation approach in two stages. In the first stage, we use correspondence analysis—a computationally efficient scaling method—on a random sample of one million users who follow at least 10 political pages.
   (b) If necessary, we reverse the direction of the first dimension estimated by the model (by multiplying by -1) so that Republican legislators are to the right of Democratic legislators. This is a change that affects only the direction of the scale and not the relative positions.
3. Using the estimates of pages political positions, we compute political page-affinity scores for all users.

(a) In the second stage of the model, we obtain estimates of political page-affinity for all users who follow at least one political page by taking the average political score from all the pages that they follow.

(b) As a final step, the resulting scores for all users are converted to percentiles. The five buckets in the final dataset correspond to the five quintiles of this distribution.

The model is re-run every time the URLs Data Set is updated. While the number of users for which an estimate is generated varies across model runs, it is generally around 25% of U.S. monthly active users who are 18 years of age or older. As detailed above, users who are younger than 18 years old or do not like any of the selected political Pages are excluded from the data.

Privacy Protection

We are releasing aggregates in these data using a privacy preserving technology called *differential privacy*, which allows researchers to uncover trends and patterns in the data without learning about the behavior of specific individuals. More precisely, differential privacy enables us to provide precise guarantees that no significant *additional* information about an "action" on Facebook (such as sharing a URL or liking it) taken by a person can be learned from the data beyond what is already available from external sources. This guarantee is quantified by a precise mathematical bound. Unlike other privacy protecting technologies, such as (attempts at) de-identification, differential privacy guarantees hold regardless of the auxiliary data and computing resources an adversary may possess.

Differential privacy provides *plausible deniability* to people whose information is included in the data set. In this case, it's impossible to determine—in a way that is significantly better than random guessing—whether a specific user took an action in these data, because differential privacy makes it impossible to isolate a specific row. That means it is impossible to determine whether or not information about the action exists in the dataset at all—again, in a way substantially better than random guessing, and where "substantially" is precisely quantified by a privacy parameter.

The privacy guarantees we are providing with this data release go another step and protect not only each action by user, but also all the actions by an individual user considered together (for 99% of users). This means it's not possible to determine whether or not all but the most active 1% of individuals are represented in the data at all. This guarantee may seem less important because such a high proportion of people have Facebook accounts, but even if all people have Facebook accounts being represented in our data requires having interacted with a URL shared at least approximately 100 times, and that fact is not public.

These privacy guarantees of both actions and users are generally operationalized by adding noise to data or the results of statistical procedures, or censoring large values to a fixed range. A non-technical introduction to differential privacy is available here: http://privacytools.seas.harvard.edu/files/privacytools/files/pedagogical-document-dp_0.pdf; a rigorous introduction can be found here: https://www.cis.upenn.edu/~aaroth/Papers/privacybook.pdf. Approaches to differential privacy that provides statistical guarantees for researchers can be found at Evans et al. [2021a], Evans and King [2021a,b], Evans et al. [2021b].

We now explain how we guarantee action-level differential privacy followed by how we guarantee user-level differential privacy.

Action-level zero-Concentrated Differential Privacy (zCDP). In this release, data aggregates that describe actions taken by a user on a URL are protected under a form of differential privacy called zero-concentrated differential privacy (zCDP, see Bun and Steinke [2016]).

The privacy protections are "action level" (rather than the more familiar "user-level" differential privacy) in

that the granularity of what is protected is not a single user, but rather a single action, or user-URL-action (e.g. a user sharing a specific URL).

We have made choices about the data that allow us to add significantly less noise compared with other formulations of differential privacy, essentially because it is easier to hide a great many modest numbers than a few large numbers. First, we define the unit of analysis as the *unique* user-URL-action tuple, which can occur in the data only once. This generally amounts to de-duplicating actions taken in the data. For example, if a user liked a post with the same URL more than once, we take the first instance and discard others.

This allows us to take advantage of assumptions underlying zCDP that achieve differential privacy guarantees that entail adding significantly less noise with minimal impact on the data. Because zCDP relies on the l_2 norm to formulate sensitivity (in this case the square root of the sum of squared indicator variables), we can add significantly lower levels of noise than if we instead allowed an arbitrary number of actions per user and relied on other variants of differential privacy with l_1 sensitivity formulations.

Formal definition of zCDP from Bun and Steinke [2016] Under zCDP, the key parameter governing privacy and thus noise is ρ. This parameter can be used to bound a user-level ε, the privacy parameter in the more standard parameterization of (ε, δ)-differential privacy. Below we provide a formal definition of zCDP including how the privacy parameter ρ relates to the Gaussian noise to be added to the data set, governed by σ. We then work backwards, setting ρ to attain an approximate user-level ε of 0.45 for each column in the data set.

A random mechanism, $M : \mathcal{X}^n \to \mathcal{Y}$ is (ρ)-zero-concentrated differentially private[5] if, for all $x, x' \in \mathcal{X}^n$ differing on a single entry and all $\alpha \in (1, \infty)$:

$$\mathrm{D}_\alpha \left(M(x) \| M\left(x'\right) \right) \leq \rho\alpha$$

where $\mathrm{D}_\alpha \left(M(x) \| M\left(x'\right) \right)$ is the α-Renyi divergence (see Van Erven and Harremos [2014] for a definition and comparison to KL divergence) between the distribution of $M(x)$ and the distribution of $M\left(x'\right)$.

Define a privacy loss function such that the privacy loss between two random variables Y and Y' is given by a new variable, $Z = Privloss\left(M(x) \| M\left(x'\right) \right)$. Z is then distributed according to $f(Y)$, where $f(y) = \log\left(\mathbb{P}[Y = y] / \mathbb{P}\left[Y' = y\right] \right)$, where all randomness in this distribution is due to the randomness in the mechanism, not a hypothetical data generating process.

The inequality above can then be re-written as a bound on the moment generating function of the privacy loss:

$$\mathbb{E}\left[e^{(\alpha-1)Z} \right] \leq e^{(\alpha-1)(\rho\alpha)}$$

The fact that zCDP entails a bound on the moment generating function of the privacy loss Z, $\mathbb{E}\left[e^{(\alpha-1)Z} \right]$ means that Z resembles a Gaussian distribution with mean ρ and variance 2ρ. This implies:

$$\mathbb{P}[Z > \lambda + \rho] \leq e^{-\lambda^2/4\rho}$$

for all $\lambda > 0$.

[5] We are using a special case of (ξ, ρ)-zero-concentrated differentially privacy by setting $\xi = 0$.

To define sensitivity: a function $q : \mathcal{X}^n \to \mathbb{R}$ is said to have sensitivity Δ if for all $x, x' \in \mathcal{X}^n$ differing on only a single entry, $|q(x) - q(x')| \leq \Delta$.

If $M(x)$ produces a sample from $N\left(q(x), \sigma^2\right)$, then M satisfies $\left(\Delta^2/2\sigma^2\right)$-zCDP.

The inequalities defining zCDP are *exactly tight* for the Gaussian mechanism for all values of α. For more details, see Bun and Steinke [2016].

Under zCDP, the parameter summarizing the privacy guarantee is ρ, which is achieved using the Gaussian mechanism. For the count queries here, the action-level sensitivity $\Delta = 1$ (user-level sensitivity varies as explained below). In other words, the mechanism entails adding $N(0, \sigma^2)$ noise to the data, where the relationship between σ and ρ follows (see also Bun and Steinke [2016]):

$$\rho = \frac{1}{2\sigma^2}$$

We can translate this to the more familiar epsilon-delta differential privacy framework for ease of interpretation. We use the following (Lemma 3.6, Bun and Steinke 2016):

$$\varepsilon = \rho + \sqrt{4\rho \log(\sqrt{\pi\rho}/\delta)}$$

or

$$\varepsilon = \frac{1}{2\sigma^2} + \sqrt{\frac{2}{\sigma^2} \log\left(\frac{\sqrt{\pi}}{\delta\sqrt{2\sigma^2}}\right)}$$

Because zCDP as defined here relies only on ρ, the translation to (ε, δ)-differential privacy is a two-dimensional surface, so ε depends on δ and vice-versa. If we know $\rho = 0.005$, we still must select a value for δ to get ε. So if $\rho = 0.005$, we can set $\delta = 1 \times 10^{-5}$, yielding $\varepsilon = 0.485$. Or, we can set $\delta = 1 \times 10^{-6}$, yielding $\varepsilon = 0.531$.

We can then think of users as "groups of actions" and examine how action-level privacy relates to user-level privacy, by relating the privacy guarantee on a single action to the privacy guarantee on a group of actions.

Relation to user-level differential privacy. We formulate an analogy to user level privacy based on this action-level definition. As users take more unique URL-actions (e.g., they click on different URLs), their total contribution to the data grows and more noise is required to provide plausible deniability that the user ever appeared in the data set—or in other words to provide user-level differential privacy. To be more precise, users are protected under a differential privacy guarantee similar to user-level privacy if they've taken at most k url-actions, which relies on the fact that a user-URL-action can occur only once.

We map action-level privacy to user-level privacy in two steps. First, for any one of the 11 possible actions (listed under aggregate statistics) a user may taken an arbitrary number of actions—for example, a user may have clicked on many hundreds of URLs in our dataset, a quantity we define as k. Second, we also wish to protect the fact that a user may take up to 11 of these possible actions on any one URL.

Using the composition properties of zCDP[6], we can compute the privacy guarantee for a group of k actions by a user. The l_2 sensitivity for any user who has taken at most k *unique* url-actions for count queries is

[6]We could obtain a weaker bound using the generic "group privacy" guarantee of zCDP, but in this instance, stronger bounds are possible using composition properties. This is because users can affect each count by at most 1, which results in a group of k actions having total l_2 sensitivity of $\sqrt{k}$, rather than k.

$\sqrt{\sum_1^k 1^2} = \sqrt{k}$ (see Dwork and Roth [2014] for a review of l_2 sensitivity and its relationship to the Gaussian mechanism versus l_1 sensitivity and the Laplacian mechanism). We can then offer a group-level ρ-zCDP guarantee for a group of size k by adding a sufficient amount of noise to satisfy:

$$\rho = \frac{k}{2\sigma^2}$$

thus satisfying group-level zCDP (see Proposition 1.9, Bun and Steinke [2016], page 7-9). We can combine this equation with Lemma 3.6, Bun and Steinke [2016] to provide a user-level privacy equivalent ε:

$$\varepsilon = \frac{k}{2\sigma^2} + \sqrt{\frac{2k}{\sigma^2}\log\left(\frac{1}{\delta}\cdot\sqrt{\frac{k\pi}{2\sigma^2}}\right)}$$

To solve for σ, we can begin by reformulating the equation above as:

$$\frac{\sigma^2\epsilon^2}{2k} + \frac{k}{8\sigma^2} + \log\sigma - \frac{\epsilon}{2} - \log\left(\frac{1}{\delta}\right) - \frac{1}{2}\log(k\pi) - \frac{1}{2}\log(2) = 0$$

We can solve for σ using Newton's method, which involves picking a candidate value for σ and iterating over the equation $\sigma_{i+1} = \sigma_i - \frac{f(\sigma_i)}{f'(\sigma_i)}$ until σ converges at the root. We define $f(\sigma)$ and $f'(\sigma)$ as follows:

$$\begin{aligned} f(\sigma) &= \frac{\sigma^2\epsilon^2}{2k} + \frac{k}{8\sigma^2} + \log(\sigma) - \frac{\epsilon}{2} - \log\left(\frac{1}{\delta}\right) - \frac{1}{2}\log(k\pi) - \frac{1}{2}\log(2) \\ f'(\sigma) &= \frac{\sigma\epsilon^2}{k} - \frac{k}{4\sigma^3} + \frac{1}{\sigma} \end{aligned}$$

For example, if we wish to protect users who have taken 100 unique [url-]actions ($k = 100$) with a formal (user-level) ε guarantee of 0.45, we can set delta to $\delta = 10^{-5}$, and add $N(0, \sigma = 98)$ noise.[7]

If a user actually made $k' > k$ [url-]actions, their ρ will be larger by a factor of k'/k. And to get their effective ε, we replace k' with k in the formula above. For example above wherein $k = 100$, if for a given person, $k' = 150$, this person's effective ε will not be approximately 0.5, but instead will be 0.75. If $k' = 200$, effective ε will be 1, etc (still assuming weve fixed δ at 10^{-5}). We've set k such that 99% of users are protected under the user-level differential privacy guarantee.

In the table below, we define how much noise is added to each variable. First, for each user, we count number of unique actions each user takes on each URL. For each action, we then set k at a value above the (differentially private) 99th percentile of unique url-actions taken.[8] [9] We then select ρ to ensure our final user-level $\varepsilon_{\text{user}}$ parameter is under .45. The noise parameter σ follows from the equations above, as does our [url-]action-level ε parameter, after setting $\delta = 10^{-5}$.

[7] Note that selecting different values of delta will change the translation—for example, we can set delta to $\delta = 10^{-6}$ with the same user-level $\varepsilon = 0.5$ and get $N(0, \sigma = 107)$; or add the same amount of noise, $N(0, \sigma = 98)$, and achieve a user-level $\varepsilon = 0.542$

[8] For this calculation we use noisy-min, (see Dwork and Roth [2014]). For each percentile p, we define $f(v)$ as the proportion of people who have less than v actions and compute $s_v = |f(v) - p| + \text{Lap}(2/(\varepsilon_{\text{user}} \times N))$ and take $\underset{v}{\text{argmin}} s_v$. We set $\varepsilon_{\text{user}}$ to 0.001. N here is Facebook's monthly active user base at the end of data collection, approximately 2.4 billion.

[9] For the view field, we estimated the 99th percentile by taking the percentile across 4 randomly selected weeks (2017-05-15 - 2017-05-23; 2018-04-09 - 2018-04-17; 2018-09-25 - 2018-10-02; 2019-04-30 - 2019-05-07) and multiplying by the ratio of days in our data to days in our sample, 33.82.

For example, for shares, we first compute the total number of URLs shared by each user. We then compute the differentially private 99th percentile and round up to the nearest positive integer to get k. We then plug k, $\varepsilon_{\text{user}} = 0.45$, and $\delta = 10^{-5}$ into the equations above to solve for ρ and thus σ.

Action	k	ρ	ρ_{user}	ε	$\varepsilon_{\text{user}}$	δ	σ	Pct users protected
views	51914	0.0000	0.0052	0.00	0.45	10^{-5}	2228	> 99
click	17	0.0003	0.0052	0.10	0.45	10^{-5}	40	> 99
share	2	0.0026	0.0052	0.31	0.45	10^{-5}	14	> 99
share_without_click	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
comment	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
like	5	0.0010	0.0052	0.19	0.45	10^{-5}	22	> 99
angry	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
haha	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
love	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
sad	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
wow	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
false_news_usr_feedback	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
hate_speech_usr_feedback	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99
spam_usr_feedback	1	0.0052	0.0052	0.45	0.45	10^{-5}	10	> 99

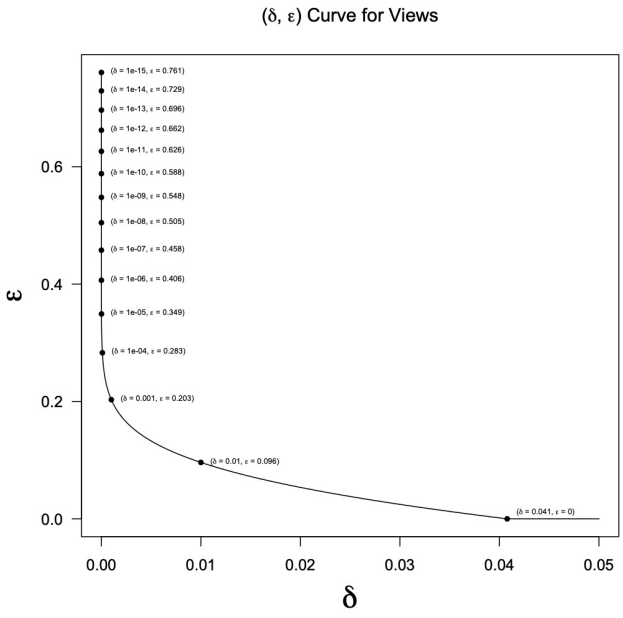
To provide a sense of how many users are protected under differential privacy at the $\varepsilon_{\text{user}} = 0.45$ level in the full table, we compute user-level multivariate histograms for all actions and compute the proportion of users whose url-action counts are all uniformly lower than all k for each action in the vector $\boldsymbol{k}$, from the table above. That number is 96.6 percent. Note that the total privacy consumption (setting $\delta = 10^{-5}$) is *not* simply $\varepsilon_{\text{user}} = 0.45 \cdot 14 = 6.3$, but rather

$$\varepsilon_{\text{user}} = \sum \boldsymbol{\rho} + \sqrt{4 \sum \boldsymbol{\rho} \cdot \log\left(\frac{1}{\delta}\sqrt{\sum \boldsymbol{\rho} * \pi}\right)} = 1.844$$

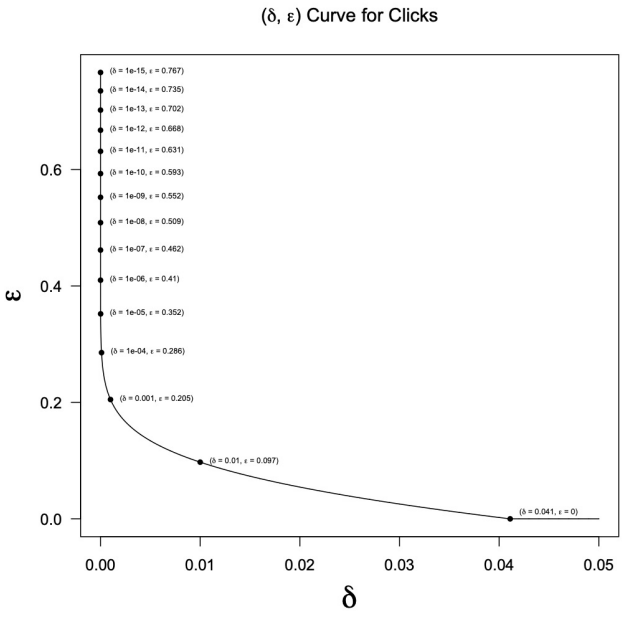
An update to the (ϵ, δ) privacy consumption. We have used the framework of zero-concentrated differential privacy (zCDP) to track the privacy consumption of this dataset. We made this decision because frameworks such as zCDP have superior compositional properties compared to the more traditional (ε, δ) differential privacy framework that allow us to much more accurately track the cumulative privacy consumption of this dataset across all action types and going into the future will allow us to more accurately track the cumulative privacy consumption as additional months of data are released. We provided a conversion from ρ-zCDP to (ε, δ)-DP in an earlier section, but drawing upon additional papers in the differential privacy literature, we are able to calculate the privacy consumption under (ε, δ)-DP using tighter bounds. We are able to state that given the noise added into the action columns in the dataset, our privacy consumption under (ε, δ)-DP was *lower* than we originally stated. It is equally important to note that our privacy bounds measured according to the zCDP metric remain tight and *unchanged*, but we can state tighter bounds in the more traditional (ε, δ) metric.

We used noise drawn from a normal distribution to implement zCDP and consequently, we are able to draw on the work of Balle and Wang [2018] and Dong et al. [Forthcoming] to calculate tighter bounds on the privacy consumption of the dataset under (ε, δ) differential privacy. For the purposes of this discussion, we will use $\Phi(\cdot)$ to denote the cumulative distribution function (CDF) of a standard normal distribution and $\phi(\cdot)$ to denote the probability distribution function (PDF) of a standard normal distribution.

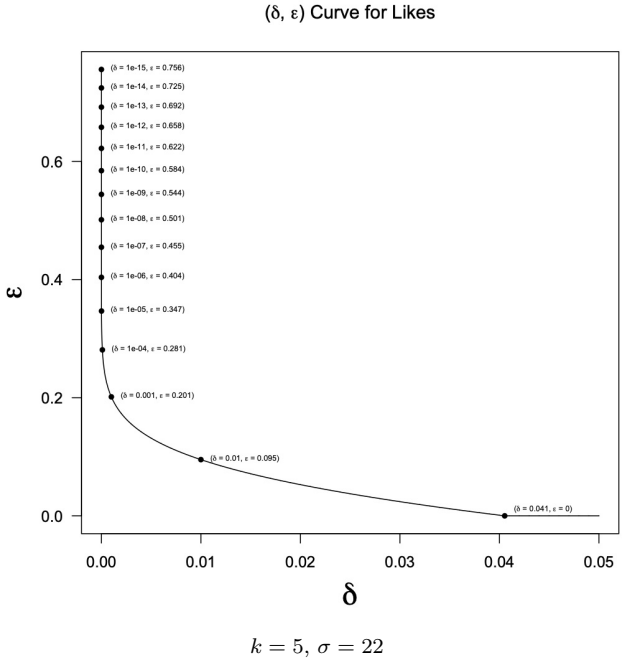
Theorem 8 in Balle and Wang [2018] states that a single Gaussian noise mechanism (with variance σ^2)



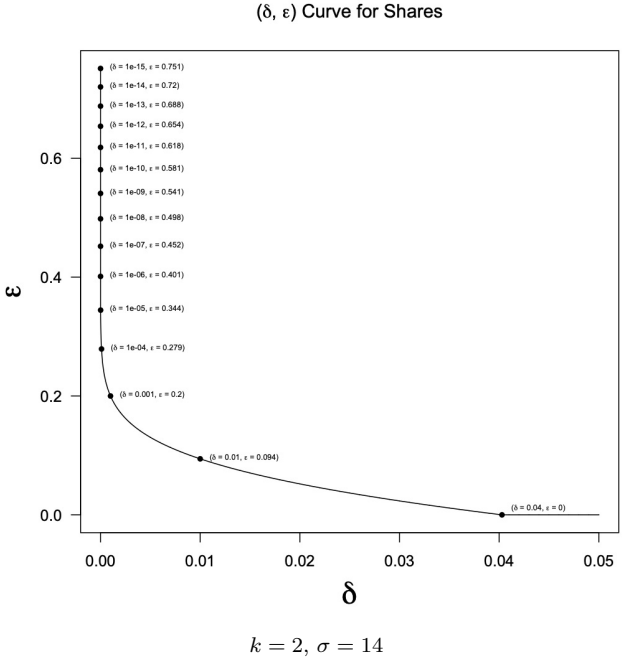
$k = 51914$, $\sigma = 2228$



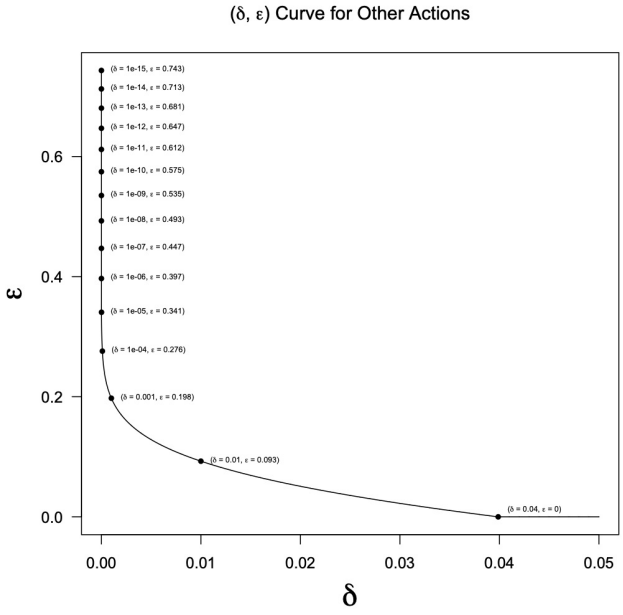
$k = 17$, $\sigma = 40$



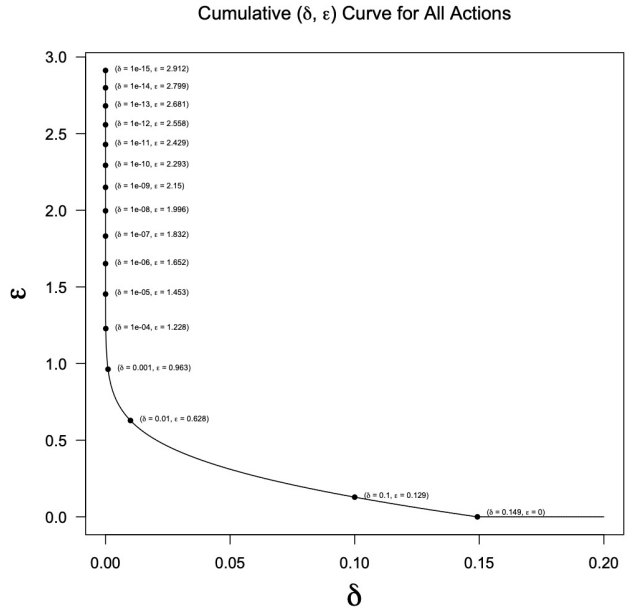
$k = 5$, $\sigma = 22$



$k = 2$, $\sigma = 14$



$k = 51914$, $\sigma = 2228$



$\mu = 0.376$

applied to a statistic with sensitivity Δ satisfies (ε, δ)-DP for any $\varepsilon \geq 0$ and $\delta \in [0, 1]$ if and only if:

$$\Phi\left(\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) - e^{\varepsilon}\Phi\left(-\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) \leq \delta$$

We have already fixed values for Δ, σ, and δ for action columns in the URLs dataset, so we can use **Theorem 8** to calculate tighter bounds on the user-level epsilons, $\varepsilon_{\text{user}}$. The L_2 sensitivity parameter Δ corresponds to $\sqrt{k}$ in our documentation where k denotes the 99th percentile of user actions.

Treating Δ, σ, and δ as constants, we can solve for the minimum ε that satisfies **Theorem 8**. To solve for ε, we can begin by reformulating **Theorem 8** as:

$$\Phi\left(\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) - e^{\varepsilon}\Phi\left(-\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) - \delta = 0$$

We can solve for ε using Newton's method, which involves picking a candidate value for ε and iterating over the equation $\varepsilon_{i+1} = \varepsilon_i - \frac{f(\varepsilon_i)}{f'(\varepsilon_i)}$ until ε converges at the root. We define $f(\varepsilon)$ and $f'(\varepsilon)$ as follows:

$$\begin{aligned} f(\varepsilon) &= \Phi\left(\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) - e^{\varepsilon}\Phi\left(-\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) - \delta \\ f'(\varepsilon) &= \phi\left(\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right)\left(-\frac{\sigma}{\Delta}\right) - e^{\varepsilon}\Phi\left(-\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) \\ &\quad -e^{\varepsilon}\phi\left(-\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right)\left(-\frac{\sigma}{\Delta}\right) \end{aligned}$$

To plot out the (ε, δ) curve, we can solve for ε across a range of values for δ (with σ and Δ fixed). The root of ε is the minimum ε (i.e. the tightest ε) that satisfies **Theorem 8**.

Theorem 8 allows us to calculate the user-level epsilons corresponding to each of the action columns, but we also need to calculate the overall $\varepsilon_{\text{user}}$ consumed across the entire set of action columns. We apply exact

computation of the privacy loss random variable under Gaussian noise to compute exact (ε, δ)- differential privacy parameters [Dong et al., Forthcoming]. The "Gaussian Differential Privacy" framework allows us to reason about composition as well. A Gaussian mechanism with standard deviation σ applied to a Δ-sensitive statistic is $\mu = \Delta/\sigma$ GDP (Gaussian Differentially Private). **Corollary 2.13** in Dong et al. [Forthcoming] states that a mechanism is μ-GDP if and only if it satisfies (ε, δ) differential privacy for all $\varepsilon \geq 0$ where:

$$\delta = \Phi\left(\frac{\mu}{2} - \frac{\varepsilon}{\mu}\right) - e^{\varepsilon}\Phi\left(-\frac{\mu}{2} - \frac{\varepsilon}{\mu}\right)$$

Note that if we set $\mu = \Delta/\sigma$, **Corollary 2.13** in Dong et al. [Forthcoming] reduces to **Theorem 8** in Balle and Wang [2018]. Composition of GDP mechanisms works as follows. Let us say that we have a collection, $\boldsymbol{\mu}$, consisting of n total GDP processes, each of which is μ_i-GDP, where $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_n)$. **Corollary 3.3** in Dong et al. [Forthcoming] states that the n-fold composition of a collection of $\boldsymbol{\mu}$-GDP mechanisms is $\|\boldsymbol{\mu}\|_2$-GDP (where $\|\cdot\|_2$ denotes the L_2 norm).

Based on **Corollary 3.3**, the total user-level privacy consumption of the dataset under μ-GDP is:

$$\|\boldsymbol{\mu}\|_2 = \sqrt{\sum_{i=1}^{n}\left(\frac{\Delta_i}{\sigma_i}\right)^2}$$

We can then calculate the total user-level privacy consumption under (ε, δ)-DP by plugging $\|\boldsymbol{\mu}\|_2$ into **Corollary 2.13** and solving for $\varepsilon_{\text{user}}$ across a range of values for δ.

$$\begin{aligned}\delta &= \Phi\left(\frac{\|\boldsymbol{\mu}\|_2}{2} - \frac{\varepsilon_{\text{user}}}{\|\boldsymbol{\mu}\|_2}\right) - e^{\varepsilon}\Phi\left(-\frac{\|\boldsymbol{\mu}\|_2}{2} - \frac{\varepsilon_{\text{user}}}{\|\boldsymbol{\mu}\|_2}\right)\\ \varepsilon_{\text{user}} &= 1.453 \text{ for } \delta = 10^{-5}\end{aligned}$$

At $\delta = 10^{-5}$, the $\varepsilon_{\text{user}}$ for each individual action type decreases from about 0.45 (as reported in an earlier section) to a range between 0.34 and 0.35. The total user-level privacy consumption across all action types under μ-GDP is about 0.376. The total user-level privacy consumption across all action types reduces from an epsilon of 1.844 (reported in an earlier section) to about 1.453. To reiterate, the privacy consumption under ρ-zCDP is unchanged: ρ_{user} remains at 0.0052 for each individual action column and remains at 0.0728 across the entire dataset.

Privacy Accounting for the User Reports Table. The first release of the "Facebook Privacy-Protected Full URLs Data Set" consisted of a URL Attributes table and a Breakdowns table, with data covering a period from January 1, 2017 to July 31, 2019. Our first update of this dataset added 5 additional months of data, so that the updated dataset covered a period from January 1, 2017 to December 31, 2019. Data in the Breakdowns table is partitioned by year-month, thus adding observations from additional year-months to this table results in no changes to the data in earlier releases, which cover year-months preceding those in the updated data. Privacy accounting in this case is straightforward: as additional year-months are added to the data, $\varepsilon_{\text{user}}$ increases as a function of column sensitivity, k. The three user report columns in the URL Attributes table (spam_usr_feedback, false_news_usr_feedback, and hate_speech_usr_feedback) presented a challenge as a far as how to add additional months of data. Data in these three columns **was not** subset by year-month, thus the values in every row recorded observations across the full period from January 1, 2017 to July 31, 2019.

One option for updating the user reports columns would have been to simply update the data in these three columns in the Attributes table so that they covered the period from January 1, 2017 to December 31, 2019. This would have required discarding the noise values used in the first release of the Attributes table and

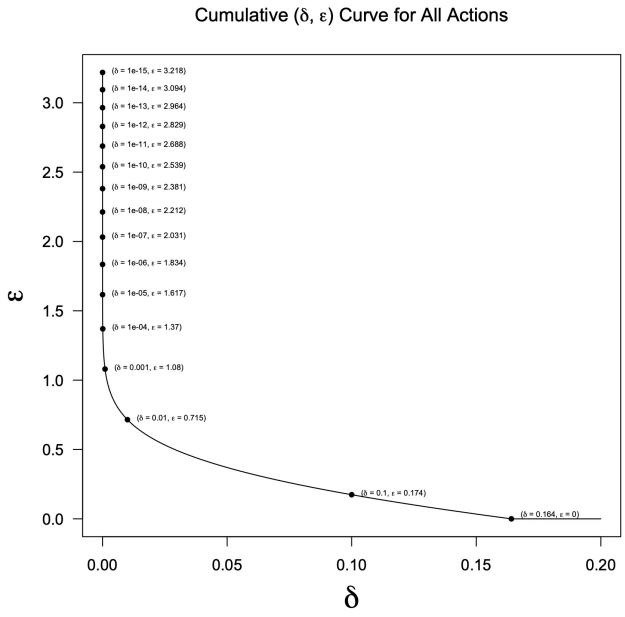
producing new draws from the Gaussian noise distribution in order to provide a differential privacy guarantee for the data in these columns. This would be necessary because if data were updated and prior noise values retained, someone comparing the first and second releases of the table could back out the exact values added into the updated data. The major downside to updating such running count variables is that it requires a considerable increase in privacy costs every time the data are updated. Because the periods covered in each update are non-mutually exclusive, each update to the dataset reduces uncertainty about values covered previously released periods, which will increase $\varepsilon_{\text{user}}$ with each update to the data, even assuming no increase in user-level sensitivity, k.

To illustrate the implications of such an approach, consider that for each of the user-report columns, $k = 1$ and $\sigma = 10$, corresponding to a privacy guarantee of $\mu = k/\sigma = 0.1$ under μ-GDP. $\mu = 0.1$ corresponds to $\varepsilon = 0.34, \delta = 10^{-5}$ under (ε, δ)-DP. Each update to a running count column requires a privacy expenditure of $\mu = 0.1$ because the data are non-mutually exclusive. Recall from the previous section that the total privacy consumed by N mechanisms with a privacy guarantee of μ_i under μ-GDP is $||\boldsymbol{\mu}||_2$ (where $\boldsymbol{\mu} = \{\mu_i\}$ and $||\cdot||_2$ corresponds to the L_2-norm). Thus the first update to the running count mechanism would increase privacy consumption from $\mu = 0.1$ to $\mu = 0.141$ (corresponds to $\varepsilon = 0.497, \delta = 10^{-5}$). Subsequent updates would increase the column privacy consumption to $\varepsilon \approx 0.62$, $\varepsilon = 0.726$, and $\varepsilon = 0.82$ (with $\delta = 10^{-5}$). This would present an unsustainable increase in privacy consumption, so we decided that the best approach to updating data in the user reports columns would be to change these columns to aggregate data by year-month, so that user reports data could be updated in the future without changing any values in the previously reported data.

Data in the User Reports table is aggregated by year-month and covers a period from January 1, 2017 to February 28, 2021. Noise is added to the original values in the privacy-protected columns, drawn from a Gaussian distribution with standard deviation $\sigma = 10$. The column-level sensitivity for the three user report columns is unchanged from the previous sensitivity calculation for these columns, because a particular user's reports filed against any particular URL are only counted once per column in the User Reports table. As an example, if a user files multiple reports on a particular URL for containing spam and the reports occur in different year-months, only the first report is counted. In future updates of the "Facebook Privacy-Protected Full URLs Data Set", we will be able to add data from subsequent year-months to the User Reports table, while rows representing year-months included in previous data releases will not change. Releasing the User Reports table incurs a one-time increase in the overall privacy consumption because the User Reports table has data covering year-months between January 2017 and July 2019, while the user reports columns in the previously released URL Attributes table cover the same period. The number of privacy protected columns increases from 14 to 17 and the three new privacy protected columns each have $k = 1$ and $\sigma = 10$.

The total privacy consumed by the "Facebook Full Privacy-Protected URLs Data Set" thus increases from $\mu = 0.376$ (see calculation in the previous section) to $\mu = 0.414$. Under (ε, δ)-DP, with $\delta = 10^{-5}$, $\varepsilon_{\text{user}}$ increases from 1.453 (see calculation in the previous section) to 1.617 after adding the User Reports table. The (ε, δ) curve in this section displays the set of values of ε which satisfy a range of potential values of δ for $\mu = 0.414$.

Replication Information for User Reports Data. The changes to the aggregation structure of spam_usr_feedback, false_news_usr_feedback, and hate_speech_usr_feedback (described in the preceding section) were implemented such that the originally released data (which was not aggregated by year-month) will remain available in the URL Attributes table. In the URL Attributes table, spam_usr_feedback, false_news_usr_feedback, and hate_speech_usr_feedback cover a period from January 1, 2017 to July 31, 2019, but the values in these columns will not be updated in the future. In the User Reports table, spam_usr_feedback, false_news_usr_feedback, and hate_speech_usr_feedback cover a period from January 1, 2017 to February 28, 2021, with the data aggregated by year-month. As subsequent months of data are released, the User Reports table will be updated to add new year-months that record user reports filed during that period.



$\mu = 0.414$

Implementation. We operationalize this protection simply by adding Gaussian noise to the aggregations (counts) based on the table above.

We generate this noise using the Yarrow-160 cryptographically secure pseudo-random number generator [Kelsey et al., 1999]. Our implementation relies on noise generated from the /dev/urandom device in the Linux kernel. The idea is to gather "environmental noise," including inter-keyboard timings, inter-interrupt timings from some interrupts, and other non-deterministic events that are difficult for an adversary to measure [Torvalds, 2014]. The device gathers randomness from these sources and adds them to an entropy pool, which it mixes using a function similar to a cyclic redundancy check.

Other privacy-preserving technology. We have applied other privacy-preserving technologies to these data in addition to differential privacy. First, access is limited to grantees and provided in a secure environment. What's more, all URLs included in these data have been shared at least 100 times $+Lap(5)$ noise by unique users with fully public privacy settings and we've taken steps to remove unintentional PII from URLs and ensure they contain only navigation-critical information as outlined below.

URL Sanitization

This section describes the URL-sanitization procedures used to clean the data set. The code used to execute steps 3-8 below can be found here: https://github.com/facebookresearch/URL-Sanitization.

1. Redirects are followed to the terminal URL, including URL shorteners.
2. If the terminal webpage has an "og:url" meta-tag, the associated URL becomes the consolidated URL—often referred to as the "canonical URL." If not, the rel = "canonical" tag is used. If neither tag is provided, the canonical URL is taken from the raw URL address. NOTE: the terminal webpage may differ from the "og:url" tag. For example: the "og:url" tag for https://www.dailymail.co.

uk/news/article-4367746/WikiLeaks-says-CIA-disguised-hacking-Russian-activity.html is actually http://www.dailymail.co.uk/~/article-4367746/index.html, which is how the URL is recorded in this data set.[10] Researchers can use Facebook's Object Debugger https://developers.facebook.com/tools/debug/og/object/ for information about any single URL. Furthermore, due to a number of prominent websites including the Washington Examiner and FoxNews.com making changes to their websites, the og:url tag will sometimes point to a different URL today than when it was originally shared on Facebook.[11] We provide the originally shared canonical URL.

3. The vast majority of obvious PII (personally identifiable information) contained in URLs is already removed by virtue of filtering URLs to those with on average 100 public shares, since less frequently shared URLs contain the bulk of PII.

4. For urls with query strings (∼21.8% of URLs above), special processing is applied. A query string in a URL passes data to the server when a client requests content, for example the "v=Ipi40cb_RsI" in https://www.youtube.com/watch?v=Ipi40cb_RsI. Sometimes query parameters provide navigation data, which tells the server what content to deliver to the client, as above. However, query parameters can also pass to the server data irrelevant to navigation, such as whether a URL was accessed from Twitter or Reddit, tracking data, and/or PII. We have attempted to remove query parameters unrelated to content navigation by iteratively removing each query parameter and testing the resulting content for differences with original page content (above and beyond the difference introduced by re-loading the page, which can occur due to ads, 'suggested content,' and/or randomized menu options). Note that for the vast majority of URLs, removing these parameters does not result in content that is different from the original. This is done at the domain level for 100 URLs (unless the domain has fewer than 100 URLs in the data) and repeated annually, so as to maintain accuracy in domain-level classification of query strings as necessary or unnecessary for navigation.

5. We keep query params that result in a different page title AND html content that differs by more than 2%, OR content that is > 95% different from original page. This measure is based on the difflib Python library and is defined as 2.0 * M/T, where M is the number of sequence matches and T is the number of elements in both sequences.

6. All URLs from domains that consistently fail to return a valid response within 120 seconds or consistently return a response under 100 characters are stripped of all query parameters.

7. Query parameter values that contain common phonenumber patterns are removed using the phonenumbers Python library.

8. Any email addresses that appear in any part of the URL string are removed using regular expressions.

Example URLs. Left raw, right processed. Non-essential query values have been altered to protect privacy.

[10] Thanks to Simon Hegelich for surfacing this example.

[11] We thank Juan Carlos Medina Serrano for pointing this out. One example includes the URL http://www.washingtonexaminer.com/a-hillary-clinton-donor-paid-500000-to-fund-women-who-would-accuse-trump-of-sexual-misconduct/article/2644747 which currently resolves to the following address https://www.washingtonexaminer.com/a-hillary-clinton-donor-paid-500-000-to-fund-women-who-would-accuse-trump-of-sexual-misconduct. However, previous versions of the web page resolved to the former address and used that address in the "og:url" meta-tag, as can be seen via the Wayback Machine: https://web.archive.org/web/20180104233644/http:/www.washingtonexaminer.com/a-hillary-clinton-donor-paid-500000-to-fund-women-who-would-accuse-trump-of-sexual-misconduct/article/2644747.

Raw URL	Processed URL
https://media1.tenor.co/images/da7eb8198618472aa82151e5d704f521/tenor.gif?itemid=5265827	https://media1.tenor.co/images/da7eb8198618472aa82151e5d704f521/tenor.gif
https://www.pivot.one/app/invite_login?inviteCode=csdfeddshkuyfckyc	https://www.pivot.one/app/invite_login
https://www.youtube.com/watch?v=oXWsoqesw7A&feature=youtu.be	https://www.youtube.com/watch?v=oXWsoqesw7A
https://www.youtube.com/watch?v=oX_fLP191-k&list=RDoX_fLP191-k	https://www.youtube.com/watch?v=oX_fLP191-k
https://news.google.com/newspapers?nid=2478&dat=10260530&id=xFc1AAAAIBAJ&sjid=iiUMAAdFJSIBAJ&pg=1558%2C27085012&hl=en	https://news.google.com/newspapers?id=xFc1AAAAIBAJ&pg=1558%2C27085012

Third Party Fact-Checker Ratings and Precedence Rules Explained:

Based on a single fact-check, Facebook can reduce the distribution of a specific piece of false content. Facebook also uses similarity detection methods to identify duplicates of debunked stories and reduce their distribution as well. Facebook can use this as a signal to reduce the overall distribution of Pages and web sites that repeatedly share things found to be false by fact-checkers. Facebook is able to get useful signals about false content that we can then feed back into its machine learning model, helping it more effectively detect potentially false items in the future.

Occasionally, multiple fact-checkers apply different ratings to the same piece of content. In these cases, the more definitive rating takes precedence, e.g. 'False' or 'True' trumps 'Mixture'. In very rare cases where the two most definitive ratings, 'True' and 'False', are applied to the same piece of content, 'True' takes precedence since we refrain from demoting content rated 'True' by a fact-checking partner. Our tpfc_rating incorporates the below precedence rules when deciding how to handle multiple fact checker ratings for the same URL. It is very rare for multiple fact-checkers to rate the same URL.

For third-party fact-checked content, a fact-checker in a country other than the top public shares country may have rated content if it circulated broadly within their country. For a complete list of our third party fact checkers, please visit this website and this one.

Instance	Example	Rule
Same Fact Checker, Multiple Ratings	A publisher appeals to the fact checker or the publisher updates the content, causing the fact checker to change its rating of the content	Use the rating with the latest timestamp
Many Fact Checkers, one rating per fact checker	Multiple partners fact check the same claim	Use the rating that wins the following precedence rule: True > False or Prank Generator > False Headline or Mixture > Not Eligible or Satire or Opinion > Not Rated
Many Fact Checkers, more than one rating per fact checker	Multiple partners have fact checked the same claim and some or all have revised their initial rating of the content	First take latest rating for each Fact Checker, then decide according to the same precedence rule as above using the latest ratings only: True > False or Prank Generator > False Headline or Mixture > Not Eligible or Satire or Opinion > Not Rated

Errata

10/5/21: The URL Breakdown Table for the U.S. (`country = 'US'`) did not include aggregate statistics for users with `NULL` page-affinity scores (`political_page_affinity IS NULL`). This has been corrected in versions of the data starting with v8. U.S. users with non-`NULL` political page-affinity scores represent 24.7% of U.S. monthly active Facebook users.

Acknowledgements

We thank Aaron Roth and Ilya Mironov for providing extensive guidance on differential privacy. We also thank Danfeng Zhang, Salil Vadhan, Abhradeep Guha Thakurta, Thomas Steinke, Julia Lane, Daniel Goroff, Daniel Kifer, and Cynthia Dwork for feedback and helpful discussions.

Appendix 1: Modified Report Noisy Max

Below, we reproduce an analysis based on a extension of the "report noisy max" differential privacy mechanism from Dwork and Roth (2014) to arbitrary sensitivity Δ score functions. This algorithm is based on correspondence with Aaron Roth and reproduced here with his permission.

Let $\mathcal{X}^n$ be an arbitrary data domain and let $\mathcal{O}$ be an arbitrary finite outcome space of cardinality K. Let $f : \mathcal{X}^n \times \mathcal{O} \to \mathbb{R}$ be an arbitrary sensitivity Δ function in its first argument (i.e. $f(\cdot, o)$ is a sensitivity Δ function for all $o \in \mathcal{O}$). Define *report noisy max* as the algorithm that first samples $Z_o \sim \text{Lap}(2\Delta/\varepsilon)$, and then outputs $RNM(D) = \arg\max_{o\in\mathcal{O}}(f(D, o) + Z_o)$.

Theorem 1. *The Report Noisy Max algorithm satisfies ε-differential privacy.*

Proof. To simplify notation, throughout the argument, we assume that $\arg\max_{o\in\mathcal{O}}(f(D, o) + Z_o)$ is unique. This is true with probability 1 over the randomness of Z, and hence does not affect the claim of differential privacy. Given a noise vector $Z \in \mathbb{R}^K$, write $o(D, Z) = \arg\max_{o\in\mathcal{O}}(f(D, o) + Z_o)$ to denote the element output by report noisy max, given that the noise is realized as Z. For each $D \in \mathcal{X}^n$ and $o \in \mathcal{O}$, Let $\mathcal{E}(D, o) = \{Z : o(D, Z) = o\}$ be the set of noise vectors that result in o being output. Fixing any output $o \in \mathcal{O}$ and noise vector $Z \in \mathbb{R}^K$, let $\tilde{Z}$ be the vector such that $\tilde{Z}_o = Z_o + 2\Delta$, and $\tilde{Z}_{o'} = Z_{o'}$ for all $o' \neq o$.

First, observe that by inspection of the pdf of the Laplace distribution, when each coordinate is sampled independently $Z_o \sim \text{Lap}(2\Delta/\varepsilon)$ we have that (abusing notation to write $\Pr[Z]$ for the probability density of the vector Z) $\Pr[Z] \leq e^{\varepsilon} \Pr[\tilde{Z}]$.

The crux of the argument follows from the following lemma:

Lemma 1. *For every pair of neighboring $D, D' \in \mathcal{X}^n$, $Z \in \mathbb{R}^K$, and $o \in \mathcal{O}$, if $Z \in \mathcal{E}(D, o)$ then $\tilde{Z} \in \mathcal{E}(D', o)$. In particular:*

$$\mathbb{1}[Z \in \mathcal{E}(D, o)] \leq \mathbb{1}[\tilde{Z} \in \mathcal{E}(D', o)]$$

Proof. Observe that for every $o' \neq o$, we have:

$$\begin{aligned} f(D', o) + \tilde{Z}_o &\geq f(D, o) - \Delta + \tilde{Z}_o \\ &= f(D, o) + Z_o + \Delta \\ &> f(D, o') + Z_{o'} + \Delta \\ &\geq f(D', o') + Z_{o'} \\ &= f(D', o') + \tilde{Z}_{o'} \end{aligned}$$

Here the first and last inequalities follow from the fact that f is a sensitivity Δ function in its second argument, and the third inequality follows from the fact that o is the unique maximizer of $f(D, o) + Z_o$. Hence we can conclude that o is also the unique maximizer of $f(D', o) + \tilde{Z}_o$. □

Again abusing notation by writing $\Pr[Z]$ to refer to the probability density on the vector Z, we can calculate:

$$\begin{aligned}
\Pr[RNM(D) = o] &= \int_{\mathbb{R}^K} \Pr[Z] \cdot \mathbb{1}[Z \in \mathcal{E}(D, o)] dZ \\
&\leq \int_{\mathbb{R}^K} e^{\varepsilon} \Pr[\tilde{Z}] \cdot \mathbb{1}[Z \in \mathcal{E}(D, o)] dZ \\
&\leq \int_{\mathbb{R}^K} e^{\varepsilon} \Pr[\tilde{Z}] \cdot \mathbb{1}[\tilde{Z} \in \mathcal{E}(D', o)] dZ \\
&= \int_{\mathbb{R}^K} e^{\varepsilon} \Pr[Z] \cdot \mathbb{1}[Z \in \mathcal{E}(D', o)] dZ \\
&= e^{\varepsilon} \Pr[RNM(D') = o]
\end{aligned}$$

Here the third line follows from Lemma 1 and the fourth line follows from a change of variables. This establishes ε-differential privacy. □

Appendix 2: Distribution of URL-interactions across countries and variables.

In order to provide more insights into the distribution of the number of unique URL actions taken by country, we calculated differentially private percentiles, using the noisy-min method discussed in footnote 6 and Appendix 1. These are available on the Facebook Research Tool.

Appendix 3: Demonstrating tightness of ε given Δ, σ, δ

In our earlier discussion, we noted that root of ε is the minimum ε (i.e. the tightest ε) that satisfies **Theorem 8** in Balle and Wang [2018]. We can demonstrate this by noting that we can simplify $f'(\varepsilon)$ further because $\phi\left(\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right)\left(-\frac{\sigma}{\Delta}\right) = e^{\varepsilon}\phi\left(-\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right)\left(-\frac{\sigma}{\Delta}\right)$. This gives us the following simplified expression for $f'(\varepsilon)$:

$$f'(\varepsilon) = -e^{\varepsilon}\Phi\left(-\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right)$$

Given that $\varepsilon \geq 0$, $f'(\varepsilon) < 0$ over all possible values of ε, meaning that $f(\epsilon)$ is a monotonically decreasing function. Consequently, values of ε greater than or equal to the root will satisfy the inequality in **Theorem 8**, but the root is the tightest ε that satisfies this theorem. To provide a more concrete example, a privacy mechanism that satisfies $(\varepsilon = 0.45, \delta = 10^{-5})$ differential privacy also satisfies $(\varepsilon = 0.5, \delta = 10^{-5})$ differential privacy, but the reverse is not necessarily true.

We simplified $f'(\varepsilon)$ by noting that the first and the third terms will cancel out. Let us consider the first term of $f'(\varepsilon)$:

$$\begin{aligned}
\phi\left(\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right)\left(-\frac{\sigma}{\Delta}\right) &= \left(-\frac{\sigma}{\Delta}\right)\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right)^2\right] \\
&= \left(-\frac{\sigma}{\Delta}\right)\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{\Delta^2}{4\sigma^2} - \varepsilon + \frac{\varepsilon^2\sigma^2}{\Delta^2}\right)\right] \\
&= \left(-\frac{\sigma}{\Delta}\right)\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{\Delta^2}{8\sigma^2} + \frac{\varepsilon}{2} - \frac{\varepsilon^2\sigma^2}{2\Delta^2}\right]
\end{aligned}$$

Now let us consider the third term of $f'(\varepsilon)$:

$$\begin{aligned}
-e^{\varepsilon}\phi\left(-\frac{\Delta}{2\sigma}-\frac{\varepsilon\sigma}{\Delta}\right)\left(-\frac{\sigma}{\Delta}\right) &= \left(\frac{\sigma}{\Delta}\right)\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(-\frac{\Delta}{2\sigma}-\frac{\varepsilon\sigma}{\Delta}\right)^2\right]\exp(\varepsilon) \\
&= \left(\frac{\sigma}{\Delta}\right)\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{\Delta^2}{4\sigma^2}+\varepsilon+\frac{\varepsilon^2\sigma^2}{\Delta^2}\right)\right]\exp(\varepsilon) \\
&= \left(\frac{\sigma}{\Delta}\right)\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{\Delta^2}{8\sigma^2}-\frac{\varepsilon}{2}-\frac{\varepsilon^2\sigma^2}{2\Delta^2}\right]\exp(\varepsilon) \\
&= \left(\frac{\sigma}{\Delta}\right)\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{\Delta^2}{8\sigma^2}+\frac{\varepsilon}{2}-\frac{\varepsilon^2\sigma^2}{2\Delta^2}\right]
\end{aligned}$$

References

Christopher A. Bail, Lisa P. Argyle, Taylor W. Brown, John P. Bumpus, Haohan Chen, M. B. Fallin Hunzaker, Jaemin Lee, Marcus Mann, , Friedolin Merhout, and Alexander Volfovsky. Exposure to opposing views on social media can increase political polarization. *Proceedings of the National Academy of Sciences*, 115(37):9216–9221, 2018.

Borja Balle and Yu-Xiang Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *35th International Conference on Machine Learning (ICML)*, 2018. URL https://arxiv.org/abs/1805.06530.

Pablo Barberá, John T Jost, Jonathan Nagler, Joshua A Tucker, and Richard Bonneau. Tweeting from left to right: Is online political communication more than an echo chamber? *Psychological science*, 26(10): 1531–1542, 2015. URL https://journals.sagepub.com/doi/abs/10.1177/0956797615594620.

Robert Bond and Solomon Messing. Quantifying social medias political space: Estimating ideology from publicly revealed preferences on facebook. *American Political Science Review*, 109(1):62–78, 2015.

William J. Brady, Julian A. Wills, John T. Jost, Joshua A. Tucker, and Jay J. Van Bavel. Emotion shapes the diffusion of moralized content in social networks. *Proceedings of the National Academy of Sciences*, 114(28):7313–7318, 2017.

Mark Bun and Thomas Steinke. *Concentrated differential privacy: Simplifications, extensions, and lower bounds*, pages 635–658. 2016.

Jinshuo Dong, Aaron Roth, and Weijie J. Su. Gaussian differential privacy. *Journal of the Royal Statistical Society, Series B*, Forthcoming. URL https://arxiv.org/abs/1905.02383.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Georgina Evans and Gary King. Statistically valid inferences from differentially private data releases, with application to the facebook urls dataset. *Political Analysis*, 2021a. URL GaryKing.org/dpd.

Georgina Evans and Gary King. Statistically valid inferences from differentially private data releases ii: Extensions to nonlinear transformations. 2021b. URL GaryKing.org/dpd2.

Georgina Evans, Gary King, Margaret Schwenzfeier, and Abhradeep Thakurta. Statistically valid inferences from privacy protected data. 2021a. URL GaryKing.org/dp.

Georgina Evans, Gary King, Adam D. Smith, and Abhradeep Thakurta. Differentially private survey research. *Working Paper*, 2021b. URL GaryKing.org/DPsurvey.

Michael J. Greenacre. *Correspondence analysis in practice*. Academic Press: London, England, 1993.

Hans J. G. Hassell, John B. Holbein, and Matthew R. Miles. There is no liberal media bias in which news stories political journalists choose to cover. *Science Advances*, 6(14):eaay9344, 2020.

Kosuke Imai, James Lo, and Jonathan Olmsted. Fast estimation of ideal points with massive data. *American Political Science Review*, 110(4):631–656, 2016.

John Kelsey, Bruce Schneier, and Niels Ferguson. *Yarrow-160: Notes on the design and analysis of the yarrow cryptographic pseudorandom number generator*, pages 13–33. 1999.

Chris Tausanovitch and Christopher Warshaw. Estimating candidates political orientation in a polarized congress. *Political Analysis*, 25(2):167–187, 2017.

Linus Torvalds. *Linux Kernel drivers/char/random.c comment documentation.* 2014.

Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.