Federated and differentially private estimation of KL divergence

Sayan Biswas
EPFL
Lausanne, Switzerland
sayan.biswas@epfl.ch

Graham Cormode
University of Oxford
Oxford, UK
graham.cormode@cs.ox.ac.uk

Carsten Maple
University of Warwick
Coventry, UK
cm@warwick.ac.uk

Mary Scott
Infectious Diseases Data Observatory
Oxford, UK
mary.scott@ndm.ox.ac.uk

arXiv:2411.16478v3 [cs.LG] 21 Aug 2026

***Abstract*—Measuring distribution drifts is a key task in managing distributed, sensitive data, as it underpins a wide range of federated learning and analytics applications. In many practical settings, however, directly sharing such information is either undesirable (*e.g.*, due to privacy concerns) or infeasible (*e.g.*, due to high communication costs). In this work, we present FedPriKL, a novel method for estimating the KL divergence of data across federated computational models under differential privacy (DP) guarantees. We establish its theoretical properties, showing that FedPriKL is unbiased with low and bounded sensitivity and variance, thereby ensuring strong utility under DP. In addition, we present an empirical study that explores parameter choices to optimize accuracy while minimizing communication overhead. Our experiments demonstrate that FedPriKL achieves accuracy comparable to a similar sampling-based non-private estimator, while delivering greater stability and accuracy than a baseline variant in which users perturb their data locally to obtain privacy guarantees.**

***Index Terms*—KL divergence, differential privacy, federated computing.**

I. Introduction

Modern applications in data analysis and machine learning work with high-dimensional data to support inferences and provide recommendations [1], [2]. Increasingly, the data to support these tasks comes from individuals who hold their data on personal devices such as smartphones and wearables. In the federated model of computation [3], [4], this data remains on the users' devices, which collaborate and cooperate to build accurate models by performing computations and aggregations on their locally held information (*e.g.*, training and fine-tuning small-scale models).

A key primitive needed is the ability to compare the distribution of data held by these clients with a reference distribution. For instance, a platform or a service provider would like to know whether the overall behavior of the data is consistent over time for deploying the best-fitting and most relevant model. In cases where the data distribution has changed, it may be necessary to trigger model rebuilding or fine-tuning, whereas if there is no change the current model can continue to be used. Hence, we seek ways to measure the similarity of distributions within the federated setting, where strong privacy and security guarantees must be maintained [5], [6], while also ensuring that the computation and communication overheads are bounded.

Among ways to compare probability distributions, the Kullback-Leibler (KL) divergence [7] is a standard information-theoretic measure widely used in statistics and machine learning. Given two distributions P_1 and P_2 defined over the same domain, the KL divergence between them captures the information-theoretic difference of P_1 from P_2, sometimes expressed as the amount of "surprise" or "information gain" in seeing samples from P_1 when expecting P_2. Its asymmetry is useful in our setting: we will estimate $D_{KL}(\Pi\|P)$, measuring departure of the private population P from a fixed public reference Π. As usual for this direction of KL, finite divergence requires $P(x) > 0$ wherever $\Pi(x) > 0$. If we have a complete representation of P_1 and P_2 to work with, then the KL divergence can be computed directly in its closed form. However, in federated settings catering to modern-day applications, the test distribution is often defined implicitly by the empirical frequencies of samples held across a distributed set of clients. Typically, the domains of these client-side distributions are too large and the data too sensitive to permit a full-sharing approach. A trivial solution is for a central server to collect samples from all clients together in a single location, and to evaluate the KL divergence itself. Although this method gives an accurate answer, it fails to consider privacy or efficiency. Furthermore, the parameter space of the clients' data can be extremely large, particularly in the setting of modern-day machine learning. Thus, this initial *full-sharing* method fails to meet the practical requirements of the scenario. The cost overhead will be high: communicating all the samples or their empirical frequencies can consume significant bandwidth, since the number of data points or the domain size can be very large. Moreover, this approach raises privacy concerns for the participants since it reveals the samples held by each client directly to the server. Variations on this paradigm which have clients add noise to the histogram of their item frequencies also fail due to the high domain size involved. Instead, we seek solutions that reduce the overhead, and provide provable privacy guarantees under various models of trust.

In this paper, we study the problem of estimating the KL divergence between a public reference distribution and an empirical distribution held by federated clients. Our solution satisfies formal differential privacy constraints and avoids communicating full histograms over the data domain. Our approach

is to reduce the total communication cost by sampling a subset of clients and items to participate in building a randomized estimator, and to ensure that differential privacy (DP) is satisfied on the output. Sampling alone is insufficient to solve our problem since it still reveals the sensitive information of the clients included in the sample [8]. In fact, via the *shift invariance theorem*, Guha et al. [9] showed that it is not possible to obtain even a non-private sketch of KL divergence between two unknown distributions in space smaller than that of the domain of the distributions. Our core technical ingredient is an unbiased Monte Carlo estimator for the KL divergence, based on sampled likelihood ratios $(P(x)/\Pi(x))$, which allows us to build the first private estimator for this task. We present FedPriKL, a sampling-based protocol that queries only a small number of points drawn from the reference distribution and uses secure aggregation to estimate the corresponding probability masses in the private federated distribution. This sidesteps the impossibility result by restricting attention to the practically relevant case of comparing a private population to a known benchmark.

Specifically, in this work, we consider a horizontal federated learning environment consisting of a set of clients that interact with a coordinating server to compute a function of interest. The primary research question we solve is to design an estimator of the KL divergence between the distribution of the clients' private data and a reference distribution representing public knowledge in a fully federated and privacy-preserving manner without needing to share the probability mass for each element in the domain. For simplicity, we assume that all clients independently and honestly follow the protocol (*i.e.*, there are no adversarial clients and no collusion). To this end, we first propose a sampling-based unbiased estimator of KL divergence, establish its key utility-related properties, and then leverage it to develop a communication-efficient method for *FED*erated and *PRI*vate estimation of *KL* divergence (FedPriKL). To our knowledge, FedPriKL is the first communication-efficient federated estimator of $D_{KL}(\Pi\|P)$ between a public reference and a private empirical population that provides formal example-level DP. In our trust model, the trusted aggregator observes only securely aggregated statistics for the sampled points; confidentiality with respect to the aggregator is supplied by the trust assumption, or by its TEE/SMC realization, rather than by differential privacy

Our key contributions are as follows:

- We formalize the problem of federated computation of KL divergence with privacy.
- We propose FedPriKL, a novel probabilistic estimator for KL divergence based on sampling, and provide a comprehensive theoretical analysis that formally characterizes its accuracy. In addition, by analyzing the sensitivity of the proposed estimator, we demonstrate how example-level differential privacy guarantees can be achieved through carefully designed noise addition within the federated model, and we provide a theoretical characterization of the optimal parameters to maximize the utility of FedPriKL.
- We present an experimental study of our methods applied to real data, and compare the impact of varying different parameters such as the rate of client sampling and the level of privacy provided. We see that FedPriKL can obtain similar accuracy to a non-private estimator, while providing a strong formal privacy guarantee.

II. Related Work

The federated computation model was introduced in its current form by the machine learning (ML) community, with a focus on training ML models over distributed data [10]. Since then, research has largely remained focused on this federated learning (FL) paradigm [3], while gradually expanding to encompass a wider range of tasks, under the banner of "federated analytics" or "federated computation" [4].

In this paper, we focus on designing randomized algorithms that can operate in the federated setting to estimate the similarity of datasets. This is a task in support of FL, where identifying when a distribution has changed can be important in determining when to improve an existing model via retraining or fine-tuning. Our approach is informed by prior work which has sought to build compact "sketches" for information divergences. We develop a mechanism to privately compare different sets of data across a period of time. It is not possible to assume both distributions are unknown, because the Shift Invariance Theorem [9] means that the distance between them is not estimable in this sketching model. We navigate this issue by instead comparing one publicly known benchmark distribution with a private one, described by a large group of clients involved in an FL environment.

Our model of privacy adopts differential privacy (DP), a widely-used statistical approach that introduces carefully calibrated randomness into query answers to mask the footprint of any individual in the data [11], [8]. DP has been applied to a wide range of tasks, from gathering basic statistics such as counts and histograms [8] to the optimization of high-dimensional ML models [5]. For distance estimation, results on the privacy properties of the Johnson-Lindenstrauss lemma mean that Euclidean distance can be estimated accurately [12], [13]. However, apart from this, the study of DP for distance estimation in the federated model has been limited.

There are numerous metrics in the literature for (non-private) distance estimation. The Fréchet distance [14] has been used in the generative adversarial network (GAN) literature to evaluate the distance between real datasets, but it can only be written in closed form if both distributions are Gaussian. The Hölder divergence [15] can only be proper (zero for two identical distributions) given a specific property between conjugate exponents. The Earth-Mover distance [16] produces accurate representations in all spaces except ℓ_1, causing problems for Laplace distributions. The Kullback-Leibler (KL) divergence [7] provides a standard information-theoretic comparison that is particularly natural in our setting because one distribution is a fixed public reference. In what follows, we make the KL divergence the focus of our study.

Symbol	Meaning
D_{KL}	KL Divergence (Defn III.1)
ε, δ	Differential privacy parameters
$\Delta(f)$	Sensitivity of function f (Defn III.5)
$\psi(h)$	Mapping from histogram to probability distribution
$\eta_{\varepsilon,\delta}$	(Gaussian) noise added for differential privacy
n	Number of participating clients
m	Number of samples used to make estimator
T	Number of batches for estimator
ϕ	KL divergence estimation function (Section III-D)
P, Π	Data distributions: public (Π) and private (P)
$\pi_{\min}$	Smallest observed probability
$r(x_i)$	Odds ratio of element x_i, as $P(x_i)/\Pi(x_i)$
τ	Clipping threshold for small/zero probabilities

TABLE I: Summary of important notation

Prior work has studied general approaches to private estimation using Monte Carlo samplers in order to draw independent and identically distributed (i.i.d.) samples from the unknown distribution [17]; a good estimator is unbiased and has low variance. However, our focus is on defining a sampler that can be applied for KL divergence in the federated model.

Other research has considered different models of secure and private computation, trading off the level of trust in different distributed participants, and removing reliance on a single centralized trusted entity. For instance, work has sought to aggregate data for learning without compromising on privacy, via secure multi-party computation (SMC) combined with Gaussian noise scaling to protect all data [18]. Similarly, in the DPrio algorithm [19], clients generate noise and secret-share it with the servers, who then select a small number of clients' noise to add. Additional steps to protect from malicious clients include zero-knowledge proofs, secret-shared non-interactive proofs [20] and Boolean secret sharing [21].

III. Preliminaries and Background

We begin with key concepts from statistics and privacy. Table I lists the main notation needed.

A. Probability Distributions & Divergences

We consider probability distributions defined empirically by observations of data. Assume that each data point is drawn from a discrete finite domain $\mathcal{X}$, and write $\mathcal{P}_{\mathcal{X}}$ to be the space of probability distributions on $\mathcal{X}$. Given a finite (multi)set of data points D, write $D(x)$ for the frequency of x in D. Let P_D be the associated empirical probability distribution, *i.e.*, $P_D(x) = \frac{D(x)}{|D|}$. Given two probability distributions, P_1 and P_2, it is natural to want to measure the similarity between them. There are many commonly used statistical divergence measures. In this paper, our focus will be on the KL divergence between P_1 and P_2.

Definition III.1. For discrete probability distributions $P_1, P_2 \in \mathcal{P}_{\mathcal{X}}$, the *Kullback-Leibler (KL) divergence* between P_1 and P_2, denoted by $D_{\mathrm{KL}}(P_1 \| P_2)$, is given by:

$$D_{\mathrm{KL}}(P_1 \| P_2) = \sum_{x \in \mathcal{X}} P_1(x) \ln\left(\frac{P_1(x)}{P_2(x)}\right).$$

B. Privacy standards

Definition III.2 (Differential privacy [11], [8])**.** For any input space $\mathcal{X}_{\mathrm{in}}$ and output space $\mathcal{X}_{\mathrm{out}}$, a randomized algorithm $\mathcal{M}$ that maps any dataset whose entries belong to $\mathcal{X}_{\mathrm{in}}$ to an element in $\mathcal{X}_{\mathrm{out}}$ satisfies (ε, δ)-*differential privacy (DP)* if, for any two adjacent datasets D and D' (*i.e.*, datasets differing in at most one element), and for any measurable $O \subseteq \mathcal{X}_{\mathrm{out}}$, we have $\Pr[\mathcal{M}(D) \in O] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in O] + \delta$.

When the adjacency relation is defined element-wise on $\mathcal{X}_{\mathrm{in}}$, we obtain the notion of "example-level DP" [22]. We will apply the DP definition to outputs that are histograms. For a finite domain $\mathcal{X}$, let $\mathcal{H}_{\mathcal{X}} = \{h \mid h\colon \mathcal{X} \mapsto \mathbb{Z}_{\geq 0}\}$ be the space of all datasets (represented as histograms) whose entries belong to $\mathcal{X}$, and let $\mathcal{P}_{\mathcal{X}}$ be the space of all probability distributions on $\mathcal{X}$. Let the *cardinality* of any $h \in \mathcal{H}_{\mathcal{X}}$ be defined as $|h| = \sum_{x \in \mathcal{X}} h(x)$, and let $\psi(h)$ be the empirical distribution on $\mathcal{X}$ that we obtain from h. That is, $\psi\colon \mathcal{H}_{\mathcal{X}} \mapsto [0,1]^{|\mathcal{X}|}$ so that

$$\psi(h) = \{\psi(h)_x\colon \psi(h)_x = h(x)/|h| \quad \forall x \in \mathcal{X}\}.$$

Definition III.3 (Adjacent histograms)**.** We call histograms $h_1, h_2 \in \mathcal{H}_{\mathcal{X}}$ *adjacent*, denoted $h_1 \sim h_2$, if $|h_1|$ and $|h_2|$ are finite, and $\exists x_1, x_2 \in \mathcal{X}$, $x_1 \neq x_2$ s.t.:
(1) $h_1(x_1) - h_2(x_1) = 1$,
(2) $h_1(x_2) - h_2(x_2) = -1$, and
(3) $h_1(x) = h_2(x) \, \forall \, x \in \mathcal{X} \setminus \{x_1, x_2\}$.

Definition III.3 describes dataset pairs that have the same cardinality and differ in the count of exactly one attribute by one. In other words, we call two histograms h_1 and h_2 adjacent if they correspond to a pair of datasets with exactly one item substituted for another, but are otherwise identical. This gives the notion of "example-level differential privacy".

Definition III.4 (Adjacent distributions)**.** We call distributions $P_1, P_2 \in \mathcal{P}_{\mathcal{X}}$ *adjacent*, denoted $P_1 \sim P_2$, if there exist $h_1, h_2 \in \mathcal{H}_{\mathcal{X}}$ such that $h_1 \sim h_2$ with $P_1 = \psi(h_1)$ and $P_2 = \psi(h_2)$.

Definition III.5 (Sensitivity)**.** For any query $f\colon \mathcal{P}_{\mathcal{X}} \mapsto \mathbb{R}$ on the space of distributions, the *sensitivity of* f is:

$$\Delta(f) = \max_{\substack{P_1, P_2 \in \mathcal{P}_{\mathcal{X}} \\ P_1 \sim P_2}} |f(P_1) - f(P_2)|.$$

A standard way to obtain DP for a numeric query f is to add noise scaled by the sensitivity of f.

Corollary III.1 (Gaussian noise [8]). For any $\varepsilon, \delta \geq 0$, sampling Gaussian noise $\eta \sim \mathcal{N}\left(0, \frac{2 \ln \frac{1.25}{\delta} (\Delta(f))^2}{\varepsilon^2}\right)$ and adding this to $f(P)$ provides (ε, δ)-DP for f.

More generally, for any $\mathcal{P}' \subseteq \mathcal{P}(\mathcal{X})$, if $f\colon \mathcal{P}' \mapsto \mathbb{R}$ is a query on a subspace of $\mathcal{P}_{\mathcal{X}}$, the *sensitivity of* η *restricted to the subspace* $\mathcal{P}'$ is defined as:

$$\Delta(f|\mathcal{P}') = \max_{\substack{P_1, P_2 \in \mathcal{P}' \\ P_1 \sim P_2}} |\eta(P_1) - \eta(P_2)|.$$

C. Federated Computation

The federated model of computation has recently emerged to capture cases where multiple clients holding sensitive data, under the guidance of a coordinating server, collaboratively perform a computation over their data [4]. This covers a wide range of scenarios, from those where there are only a few powerful clients (*e.g.*, hospitals), to where there are millions or billions of clients (*e.g.*, mobile devices). We will focus on the cases where there is a moderate to large number of relatively weak clients, representing mobile devices like phones or wearables, that are constrained in terms of computation or memory. Within recent research advancements in federated computation, significant attention has focused on FL, which aims to build a ML model using clients' data. FL can be broadly classified into two categories: *(i) horizontal* or *cross-device* FL, where data is partitioned across clients, and all data points held by the clients share the same set of features (*e.g.*, photos stored locally on clients' mobile phones), and *(ii) vertical* or *cross-silo* FL, where data is partitioned by features, making it relevant when different entities hold different attributes for the same set of users (*e.g.*, hospitals in the same region specializing in different aspects of healthcare, each holding different pieces of health data for overlapping sets of patients). In typical FL use cases, where clients' data contain sensitive information (*e.g.*, locations, health records, photos, text messages, etc.), it is natural to consider the associated data distribution of the clients as private.

Definition III.6 (Federated Model)**.** In this paper, we consider a horizontally federated environment consisting of a set of clients $C = \{c_1, \ldots, c_n\}$ for some $n \in \mathbb{N}$ such that all clients independently and honestly comply with the protocol (*i.e.*, there do not exist any Byzantine clients or any adversarial collusion between them). The clients interact with a coordinating server S to compute a function of interest. The goal is to enable computation of functions of interest while minimizing the amount of sensitive information that is visible to the server.

In order to achieve a good tradeoff between accuracy and trust, we adopt a "trusted aggregator" paradigm. Our protocols involve an aggregator which securely combines the sampled client frequencies, computes the exact global estimate, and only then applies the requisite DP noise. That is, the aggregator obtains an already aggregated view of the client data, and merely completes the last step to turn it into the required estimator. Moreover, adopting a trusted aggregator does not strictly necessitate a centralized single server. In practice, this role can be decentralized and instantiated dynamically; for example, by randomly electing a trusted user node in each communication round, or by utilizing Secure Multiparty Computation (SMC) and Trusted Execution Environments (TEEs) to act as a virtual aggregator that guarantees code integrity and secure privacy noise injection.

We denote by `SecAgg` a secure aggregation, where the sum of data from a set of mutually distrustful sources is computed without revealing the value of any individual inputs. Following Definition III.6, all clients are assumed to follow the protocol honestly, while the server is honest-but-curious. Implementations of `SecAgg` range from a cluster-based model [23] to a privacy-preserving protocol [6] that uses Shamir's t-out-of-n secret sharing scheme [24] to account for dropout clients. In Section IV, `SecAgg` will be used to securely aggregate the local datasets of the clients involved under different trust models. This ensures that even the trusted aggregator only gets to see the combined information from all clients, rather than the raw data of any individual client.

Threat model. Concretely, our threat model is as follows: clients hold their own data, never revealing it directly, and participating honestly in the protocol; a `SecAgg` protocol is used to collect statistics from subsets of clients, and reveals only the summation of its inputs; a trusted aggregator sees only the output of `SecAgg` and performs some small mathematical manipulation to produce a noisy version; the server sees only this noised output, and produces the final estimator for downstream use. All participants are honest-but-curious.

D. Monte Carlo subsampling

As probability distributions P_1 and P_2 are defined via observations of data, we are unable to compute the exact expression of their KL divergence as in Definition III.1. Instead, we will use Monte Carlo subsampling [25]: build an estimator that takes in subsamples $x_1, x_2, \ldots \sim P_2$ and produces an estimator of $D_{\mathrm{KL}}(P_1 \| P_2)$. Since KL divergence is a Bregman divergence [26], it can be measured as the shortest distance between a convex function and its tangent. Setting $r_i = P_1(x_i)/P_2(x_i)$ for $x_i \sim P_2$, the estimator $\Phi(r_i) = (r_i - 1) - \ln r_i$ measures the vertical distance between $\ln r_i$ and its tangent. $\Phi(r_i)$ is the core of the estimator of KL divergence built in this paper.

IV. DP estimator of KL divergence

Given a collection of n federated clients holding data from the space $\mathcal{X}$ which collectively defines a data set D, we wish to compute an accurate estimate of the KL divergence between this data distribution and a reference distribution with formal DP guarantees. That is, each client i holds a local dataset D_i, so that $D(x) = \sum_{i=1}^{n} D_i(x)$. The global dataset D then defines a probability distribution P. The server holds a public reference distribution Π, also over $\mathcal{X}$. For instance, Π could be derived from data collected in the past or in a controlled fashion using voluntary participation of users. Hence, the goal is to provide an estimation of $D_{\mathrm{KL}}(\Pi \| P)$ under privacy guarantees.

FedPriKL in a nutshell. Our proposed approach, FedPriKL, operates in three primary phases designed to securely and efficiently estimate distribution shifts. First, participating clients utilize a shared, public reference distribution (Π) to independently calculate the empirical frequencies of their local, private datasets over the reference support. Second, these localized statistics are securely transmitted to the trusted aggregator, which sums them to form a global empirical distribution

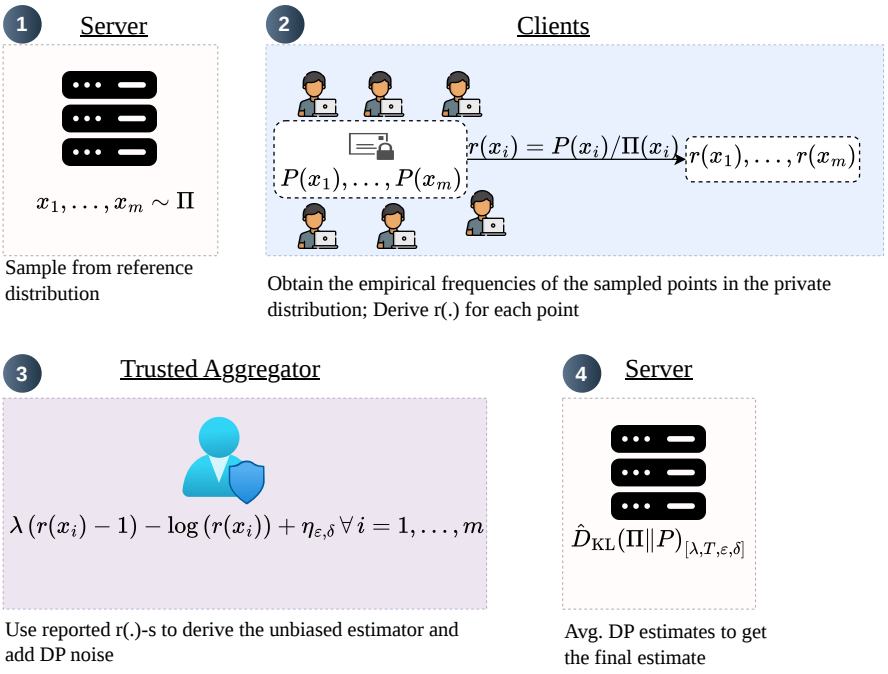


Fig. 1: The workflow of FedPriKL with a trusted aggregator

representing the entire federation. Finally, the aggregator constructs an unbiased estimate of the KL divergence between this aggregated private distribution and the public reference. Utilizing rigorously analyzed and bounded sensitivity metrics, the aggregator then calculates and injects the necessary amount of calibrated differential privacy noise to privatize the estimate. This resulting privatized estimate of KL divergence is then shared back to the clients or published, providing an accurate, privacy-preserving measure of systemic data drift without ever exposing an individual user's data distribution. Figure 1 illustrates the high-level workflow of FedPriKL with a trusted aggregator.

As a build-up to the design of FedPriKL, we start by proving the following facts that are necessary to obtain a communication-efficient and unbiased estimator of the KL divergence.

Fact IV.1. If $\Pi \in \mathcal{P}_{\mathcal{X}}$ is a full-support distribution (*i.e.*, it excludes events with zero probability), for any random variable $X \sim \Pi$ and $P \in \mathcal{P}_{\mathcal{X}}$, we have $\mathbb{E}\,[r(X)] = 1$, where $r(X) = \frac{P(X)}{\Pi(X)}$.

Proof. $\mathbb{E}[r(X)]$ can be expanded as

$$\sum_{x \in \mathcal{X}} r(x)\Pi(x) = \sum_{x \in \mathcal{X}} \frac{P(x)}{\Pi(x)}\Pi(x) = \sum_{x \in \mathcal{X}} P(x) = 1.$$

□

Fact IV.2. For a random variable X and a function r on X, let $\Phi_{\Pi,P}(r(X), \lambda) = \lambda\,(r(X) - 1) - \ln\,(r(X))$. If $r(X)$ is as defined in Fact IV.1 w.r.t. Π and P, then, under the assumption of Fact IV.1, for any $\lambda \in \mathbb{R}$, we have $\mathbb{E}_{\Pi}\,[\Phi_{\Pi,P}(r(X), \lambda)] = D_{\mathrm{KL}(\Pi \| P)}$.

Proof. As Fact IV.1 implies $\mathbb{E}[r(X) - 1] = 0$, we get

$$\mathbb{E}_{\Pi}\,[\Phi_{\Pi,P}(r(X), \lambda)] = \mathbb{E}_{\Pi}\,[\lambda\,(r(X) - 1) - \ln\,(r(X))]$$
$$= -\mathbb{E}_{\Pi}[\ln r(X)] = -\mathbb{E}_{\Pi}\left[\ln \frac{P(x)}{\Pi(x)}\right] = D_{\mathrm{KL}}(\Pi \| P).$$

□

Remark 1. As Fact IV.2 establishes that $\Phi((r(X), \lambda)$ is an unbiased estimator of $D_{\mathrm{KL}}(\Pi \| P)$, it immediately follows that for any arbitrary independent samples of $X_1, \ldots, X_M$ for some $M \in \mathbb{N}$, $\frac{1}{M}\sum_{i=1}^{M} \Phi((r(X_i), \lambda)$ is also an unbiased estimator of $D_{\mathrm{KL}}(\Pi \| P)$ with a factor M decrease in its variance.

Remark 1 enables the estimation of the KL divergence between Π and P by asking a subset of the clients to share the probability masses of a few points sampled by the server according to the reference distribution. To make our estimation of KL divergence both communication-efficient and privacy-preserving, we propose a sampling-based approach that leverages this.

In the subsequent sections concerining the theoretical analysis of FedPriKL, while the short proofs have been included in the main body of the paper, for brevity, the details of all the longer proofs are deferred to the Appendix.

A. Baseline: pre-aggregation perturbation

We first present a canonical baseline that places no trust in the aggregator, by directly adapting the estimator derived in

Algorithm 1: FEDPRIKL: Baseline

for *batch* $t = 1, \ldots, T$ **do**
  Sample (disjoint) set of participating clients C_t
  $x_{1_t}, \ldots, x_{m_t} \overset{i.i.d.}{\sim} \Pi$ and shared with C_t
  for $i = 1, \ldots, m$ **do**
    for $c \in C_t$ **do**
      `// Client side:`
      c finds frequency of $D_c(x_{i_t})$
      $P(x_{i_t}) = \texttt{SecAgg}\left(\frac{\sum_{c \in C_t} D_c(x_{i_t})}{N_t|} + \eta_{\varepsilon,\delta}\right)$, where $\eta_{\varepsilon,\delta} \sim \mathcal{N}\left(\frac{2\ln(1.25/\delta)}{(N_t|\varepsilon)^2}\right)$
    `// Server side:`
    $P'(x_{i_t}) = \max\{P(x_{i_t}), \tau\}$ where $\tau > 0$ is small constant
    $r'(x_{i_t}) = \frac{P'(x_{i_t})}{\Pi(x_{i_t})}$
$\hat{D}_{\mathrm{KL}}(\Pi \| P)_{[\lambda,T,\varepsilon,\delta]} = \frac{1}{mT} \sum_{t=1}^{T} \sum_{i=1}^{m} \Phi_{\Pi,P}(r'(x_{i_t}), \lambda)$

Remark 1. In order to have DP guarantees, the clients add noise to the aggregation of their individual records before the secure aggregation takes place to share the sum with the server. In parallel, T subsets of clients C_t are selected to participate in each batch t and are provided with i.i.d. points $x_{1_t}, \ldots, x_{m_t} \in \mathcal{X}$ sampled by the server from the reference distribution Π. The clients then securely aggregate the frequencies of the x_i computed over their local datasets and normalize them to obtain the empirical probability masses of $x_{1_t}, \ldots, x_{m_t}$ in the subpopulation distribution, *i.e.*, $\Pr_{c \in C_t}[X = x_{1_t}], \ldots, \Pr_{c \in C_t}[X = x_{m_t}]$. The clients add independent Gaussian noise $\eta_{\varepsilon,\delta}$ to each reported value. The scale of the noise is determined by ε, δ, and the number of items in C_t. For a participating client c, let $D_c(x)$ denote the number of records at c equal to x, and let $N_t = \sum_{c \in C_t} |D_c|$ denote the total number of records held by the clients participating in batch t. The empirical probability mass of x in that batch is therefore $P_t(x) = \sum_{c \in C_t} D_c(x)/N_t$. The distributed noise added by each client, summed over all clients, has sufficient variance to provide the desired (ε, δ)-DP guarantee. The server receives the aggregated value, P, and applies clipping to handle very small or negative values, obtaining $P'(x_{i_t}) = \max\{P(X = x_{i_t}) + \eta_{\varepsilon,\delta}, \tau\}$, where $\tau > 0$ is a small constant. This ensures that the logarithm computed by the server is well-defined, which is necessary for applying the KL divergence estimator from Remark 1. The privacy properties of this baseline, presented in Algorithm 1, follow by considering the sensitivity of the aggregated data reported, and scaling the Gaussian noise accordingly.

Theorem IV.3. The baseline version of FEDPRIKL (Algorithm 1) satisfies (ε, δ)-DP.

Proof. Under our notion of adjacency, which corresponds to example-level DP, making one change to the input can change $P(x_i)$ by at most $1/N$, where N is the number of samples used to estimate $P(x_i)$. Moreover, looking at $P(x_i)$ for a collection of samples x_i can be considered as a histogram query, so we can add total noise with sensitivity $2/N$ to each sampled item. Consequently, we obtain an (ε, δ)-DP guarantee by adding the Gaussian noise as specified in Algorithm 1. □

This straightforward approach, where clients add local noise and send a locally perturbed version of the aggregated probability masses of points sampled from the reference distribution, is useful as a baseline for evaluating the performance of FEDPRIKL under canonical DP guarantees. However, it does not yield an unbiased estimator of $D_{\mathrm{KL}}(\Pi \| P)$. The bias arises from the correction factor τ, introduced to avoid negative values in the estimated ratio $r'(x_{i_t})$. Consequently, the error reduction offered via Remark 1 is not applicable. This renders the estimator undesirable from a utility perspective, since the bias introduced by τ cannot be corrected by the server without violating the clients' DP guarantees. Algorithm 1 will serve as an empirical baseline for comparison against our improved estimator.

B. FEDPRIKL with trusted aggregator

In light of the shortcomings of the baseline approach described in Section IV-A, we now present the main version of FEDPRIKL in Algorithm 2. The overall structure of the estimation algorithm remains similar; however, the tasks of frequency aggregation and noise injection (required to ensure DP guarantees) are now distributed between the clients and a trusted aggregator. FEDPRIKL makes an explicit and narrowly scoped trust trade-off: securely aggregated sampled frequencies are processed by the trusted aggregator before release to the server. This permits the non-linear transformation and calibrated DP noise addition to occur after secure aggregation, avoiding the bias introduced by perturbing the individual probability masses before the transformation. The server then performs the final aggregation step and derives the estimate of $D_{\mathrm{KL}}(\Pi \| P)$. This trust assumption can be discharged by performing the aggregation on secure hardware, such as via a Trusted Execution Environment (TEE) [27], [28]. Alternatively, it can be implemented in the secure multiparty computation model (SMC) [29], where the main technical hurdle is to approximately compute the logarithm function [30]. This is consistent with the threat model of Section III-C, whose central objective is to keep clients' raw data and any statistics derived from it in the clear hidden from the server. This narrow trust assumption is highly practical; the aggregator only evaluates a single logarithm and performs basic arithmetic on securely aggregated scalars. This point is discussed further in Remark 3 at the end of Section IV.

Privacy analysis. To support the privacy analysis of our algorithm, we first bound the sensitivity of the estimator that is at the heart of Algorithm 2. In what follows, let $\hat{D}_{\mathrm{KL}}(\Pi \| P)_{[\lambda,T,\varepsilon,\delta]}$ denote the estimate of the true value of $D_{\mathrm{KL}}(\Pi \| P)$ derived by FEDPRIKL. In the end, the accuracy and privacy guarantees of the estimation are affected by the *precision parameter* $T \in \mathbb{N}$, the *variance parameter* $\lambda \in \mathbb{R}$, and the ε and δ parameters for achieving (ε, δ)-DP.

Algorithm 2: FEDPRIKL with trusted aggregator

for *batch* $t = 1, \ldots, T$ **do**
  Sample (disjoint) set of participating clients C_t
  $x_{1_t}, \ldots, x_{m_t} \overset{i.i.d.}{\sim} \Pi$ and shared with C_t
  for $c \in C_t$ **do**
    // Client side:
    for $i = 1, \ldots, m$ **do**
      c finds frequency of $D_c(x_{i_t})$
      $P(x_{i_t}) = \texttt{SecAgg}\left(\frac{\sum_{c \in C_t} D_c(x_{i_t})}{N_t}\right)$
      $r(x_{i_t}) = \frac{P(x_{i_t})}{\Pi(x_{i_t})}$
      Shared with the trusted aggregator
// Trusted aggregator side:
for $t = 1, \ldots, T$ **do**
  for $i = 1, \ldots, m$ **do**
    $\Phi'_{\Pi,P}(r(x_{i_t}), \lambda) = \Phi_{\Pi,P}(r(x_{i_t}), \lambda) + \eta_{\varepsilon,\delta}$, s.t.
    $\eta_{\varepsilon,\delta} \sim \mathcal{N}\left(0, \frac{2\ln(1.25/\delta)\Delta\left(\hat{D}_{\mathrm{KL}}(\Pi\|P)_{[\lambda,T]}\right)^2}{(\varepsilon T)^2}\right)$
    Shared with the server
// Server side:
$\hat{D}_{\mathrm{KL}}\left(\Pi\|P\right)_{[\lambda,T,\varepsilon,\delta]} = \frac{1}{mT}\sum_{t=1}^{T}\sum_{i_t=1}^{m}\Phi'_{\Pi,P}(r(x_{i_t}), \lambda)$

For convenience, let $f(P) = D_{\mathrm{KL}}(\Pi\|P)$ and $\hat{f}(P) = \hat{D}_{\mathrm{KL}}\left(\Pi\|P\right)_{[\lambda,T]}$, where $\hat{D}_{\mathrm{KL}}(\Pi\|P)_{[\lambda,T]}$ is the estimate of $D_{\mathrm{KL}}(\Pi\|P)$ without the DP noise. We now analyze the sensitivity of f and $\hat{f}$ as a precursor to appropriately add noise to obtain the final version of our estimator, $\hat{D}_{\mathrm{KL}}(\Pi\|P)_{[\lambda,T,\varepsilon,\delta]}$, with (ε, δ)-DP guarantee. Let $\mathcal{H}_{\mathcal{X}}^{>1}$ be the space of datasets where each value from $\mathcal{X}$ occurs more than once. Correspondingly, let $\mathcal{P}^{>1}$ be the set of probability distributions such that $\mathcal{P}^{>1} = \left\{P \in \mathcal{P}_{\mathcal{X}} \colon \psi^{-1}(P) \in \mathcal{H}_{\mathcal{X}}^{>1}\right\}$. This restriction is needed for the proof to avoid the vacuous case of an unbounded sensitivity value. Accordingly, we derive a constant upper bound on the sensitivity of f as follows.

Theorem IV.4. For $\mathcal{H}_{\mathcal{X}}^{>1}$ with a fixed cardinality, $\Delta\left(f|\mathcal{P}^{>1}\right) < 0.7$.

The analysis proceeds by upper bounding the change in KL divergence between neighboring distributions, by considering the worst-case values of the probability distribution. The finite-sensitivity result assumes positive support with aggregate counts exceeding one on the effective estimation domain. This is the standard support issue inherent in $\hat{D}_{\mathrm{KL}}(\Pi\|P)$: if $\Pi(x) > 0$ while $P(x) = 0$, the divergence itself is infinite. In settings with rare categories, the protocol can therefore be applied to a prespecified smoothed/effective distribution, in which case the target is the KL divergence of that distribution. FEDPRIKL's estimation and privacy analysis then apply unchanged. Any singletons in the aggregated histogram can be eliminated via standard Laplace smoothing at a negligible cost to accuracy, without altering the core FEDPRIKL protocol. Next, we bound the sensitivity of the estimated KL divergence with DP guarantee, as given by Algorithm 2.

Theorem IV.5. For $\mathcal{H}_{\mathcal{X}}^{>1}$ with a fixed cardinality N, we have $\Delta\left(\hat{f}|\mathcal{P}^{>1}\right) \leq \frac{|\lambda|}{N\pi_{\min}} + \ln 2$, where λ is the variance parameter used in the estimation and $\pi_{\min} = \min_{x \in \mathcal{X}} \Pi(x)$.

The next theorem follows since the algorithm adds noise as a function of the computed sensitivity.

Theorem IV.6. The output of FEDPRIKL with a trusted aggregator (Algorithm 2) satisfies (ε, δ)-DP.

Remark 2. The sensitivity analyses of Theorems IV.4 and IV.5 are carried out over $\mathcal{P}^{>1}$, *i.e.*, over empirical distributions in which every element of the domain is observed more than once. This restriction is what keeps the sensitivity bounded: if an adjacent substitution could reduce the count of some element from one to zero, the corresponding probability mass would vanish, the log-ratio terms in the divergence would diverge, and no finite noise scale could be calibrated to guarantee DP. The restriction should be interpreted as a condition on the effective distribution whose divergence is being estimated. If the application domain contains cells with zero or singleton counts, a fixed smoothing or support-selection rule can be specified before applying FEDPRIKL. The resulting mechanism then estimates $D_{KL}(\Pi\|P')$ for that resulting empirical distribution P'. This support treatment is separate from the randomized estimator itself, but should be fixed independently of the private release mechanism.

Utility analysis. Next, we establish that the estimate given by Algorithm 2 is indeed unbiased and bound the behaviour of its variance as a function of the parameter λ.

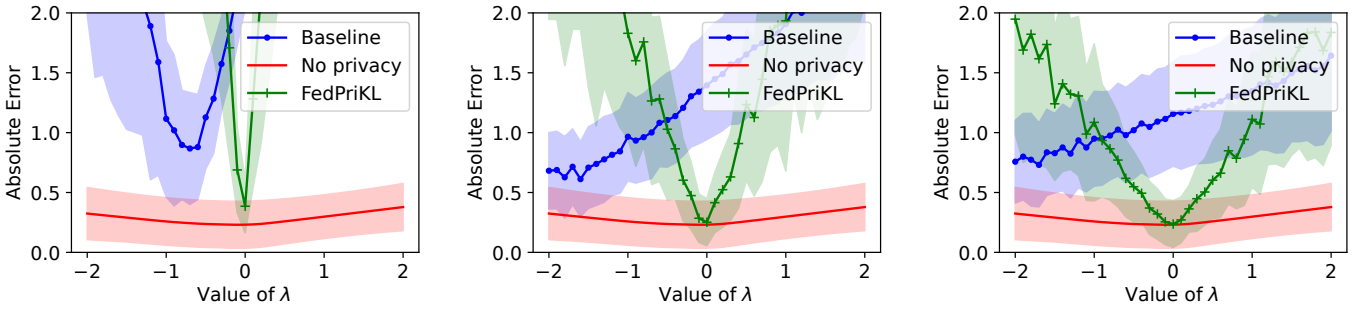
Theorem IV.7. FEDPRIKL with a trusted aggregator gives an unbiased estimator of $D_{\mathrm{KL}}(\Pi\|P)$.

Theorem IV.8. If P and Π are full-support distributions, let N and $\pi_{\min}$ be as in Theorem IV.5, and $\alpha = \max_{x \in \mathcal{X}} \frac{P(X)}{\Pi(x)}$ and $\beta = \max_{x \in \mathcal{X}} \frac{\Pi(X)}{P(x)}$. Then the estimator given by FEDPRIKL in Algorithm 2 ensures:

$$\begin{aligned}
&\mathrm{Var}\left[\hat{D}_{\mathrm{KL}}\left(\Pi\|P\right)_{[\lambda,T,\varepsilon,\delta]}\right] \\
&\leq \frac{1}{mT}\left(\lambda^2(\alpha - 1) + \max\{\alpha, \beta^2\} - 1 - D_{\mathrm{KL}}(\Pi\|P)^2\right. \\
&\left.-2\lambda\left(D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)\right)\right) \\
&+ \frac{1}{mT^3\varepsilon^2}\left(2\ln(1.25/\delta)\left((N\pi_{\min})^{-1}|\lambda| + \ln 2\right)^2\right).
\end{aligned}$$

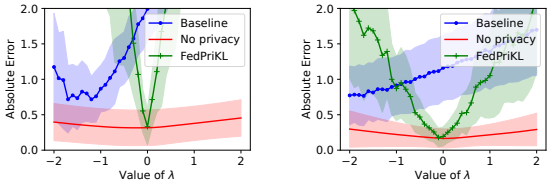
The subsequent section provides a detailed analysis of the optimal choice of the variance parameter λ that minimizes the variance of the estimator produced by FEDPRIKL.

Complexity analysis. We sample all n clients partitioned in T disjoint batches. In each of the T batches, m instances of $X \sim \Pi$ are sampled, which the clients participating in the corresponding round operate with. The complexity associated with deriving and performing a secure-aggregation-based averaging on the frequencies for each of the m sampled points is taken to be linear. Hence, the client-side computational complexity of Algorithm 2 is $\mathcal{O}(nm)$. On the trusted aggregator, the complexity of adding the Gaussian noise independently to each of the mT points and taking the average is $\mathcal{O}(mT)$.



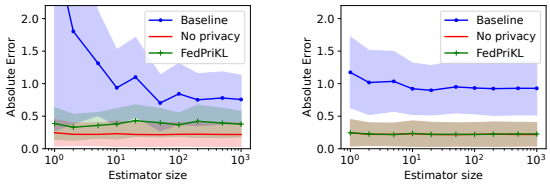
(a) $\varepsilon = 0.1$ (b) $\varepsilon = 0.5$ (c) $\varepsilon = 1.0$

Fig. 2: Absolute error of different estimators as λ varies for $\epsilon = 0.1, 0.5, 1.0$



(a) $|C_t| = 71$ ($\sim 2\%$ of clients) (b) $|C_t| = 358$ ($\sim 10\%$ of clients)

Fig. 3: Absolute error for different batch sizes $|C_t|$.



(a) $\varepsilon = 0.1$ (b) $\varepsilon = 1.0$

Fig. 4: Absolute error varying number of samples m

Thus, the overall algorithmic complexity of FedPriKL in Algorithm 2 is $\mathcal{O}(m(n+T))$.

Complementing the discussion on the assumption on the occurrence frequency of each item in the aggregated population distribution P presented in Remark 2, we briefly discuss another core technical aspect assumed on the algorithmic level of FedPriKL that is crucial for its privacy and accuracy guarantees, namely the role of the trusted aggregator.

Remark 3. We emphasize that the trust placed in the aggregator is narrow in both scope and exposure. Functionally, the aggregator performs exactly one step of Algorithm 2 (line 14): applying Φ to the securely aggregated ratios and perturbing the result with Gaussian noise calibrated to the sensitivity bound of Theorem IV.5. It never accesses raw client data and, aligned with the threat model of Section III-C, `SecAgg` ensures that the aggregator's entire view is limited to the empirical masses of the m points sampled from Π in each batch, already summed over the participating clients of that batch, *i.e.*, mT scalars in total, as opposed to individual datasets or full histograms over $\mathcal{X}$. Moreover, since the delegated task amounts to a few arithmetic operations and one logarithm evaluation, the aggregator need not exist as a distinguished physical party; it can be instantiated inside a TEE [27], [28], emulated via SMC [29], for which protocols to securely evaluate the logarithm are well studied [30], or realized under fully homomorphic encryption (FHE) [31] based techniques. It is precisely the simplicity of the aggregator's task that makes such instantiations practical, and optimizing it with advanced and secure machinery remains orthogonal to the core functionality of FedPriKL.

V. Analysis of the optimal variance parameter

We note that the bounds of the variance of our estimator depend on the choice of the parameter λ. Therefore, we seek a choice of λ that minimizes the variance as follows.

Definition V.1 (Optimal variance parameter)**.** We call a variance parameter $\lambda_{0,P}$ *optimal* for P under FedPriKL with a trusted aggregator if, for every $\lambda \in \mathbb{R}$,

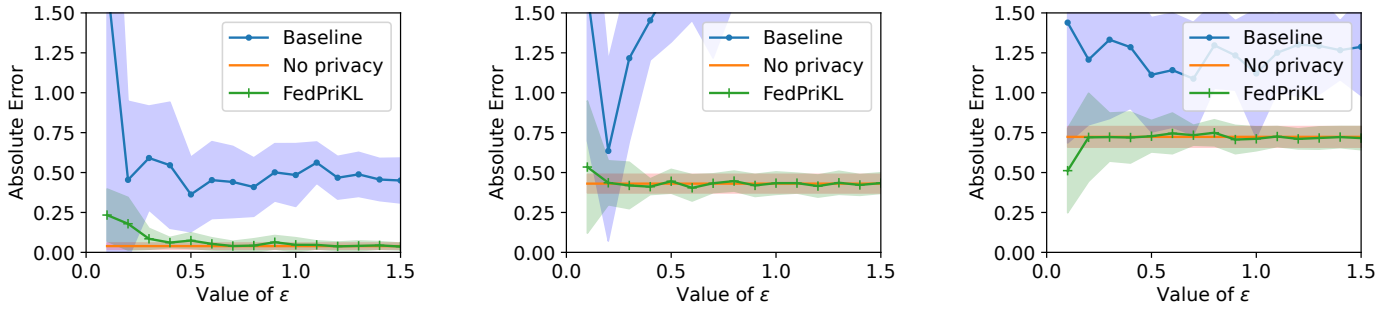
$$\mathrm{Var}\left[D_{\mathrm{KL}}\left(\Pi \| P\right)_{[\lambda_{0,P},T,\varepsilon,\delta]}\right] \leq \mathrm{Var}\left[D_{\mathrm{KL}}\left(\Pi \| P\right)_{[\lambda,T,\varepsilon,\delta]}\right].$$

Now we present a theoretical analysis to provide a foundational characterization and understanding on the optimal variance parameter λ which, in turn, optimizes the estimate obtained with FedPriKL. For notational convenience, we omit P from the subscript of $\lambda_{0,P}$ and refer to the optimal variance parameter simply as λ_0, as the clients' data distribution P is fixed in this context. In what follows, all expectations and variances are taken over $X \sim \Pi$, unless otherwise specified. Hence, the subscript $X \sim \Pi$ can be dropped from $\mathbb{E}_{X\sim\Pi}[.]$ and $\mathrm{Var}_{X\sim\Pi}[.]$ and we simply write $\mathbb{E}[.]$ and $\mathrm{Var}[.]$ when $X \sim \Pi$ is clear in the context.

Theorem V.1. Under the assumptions of Theorem IV.8, if *no* DP noise was introduced in FedPriKL, we have

$$\lambda_0 = \frac{D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)}{\mathrm{Var}_{X\sim\Pi}[r(X)]},$$

where $r(X)$ is as defined in Fact IV.1.



(a) $KL(3,8)$ – closest pair (b) $KL(4,1)$ – median pair (c) $KL(9,0)$ – furthest pair

Fig. 5: Absolute error of different estimators for different digit pairs

Corollary V.2. Under the assumptions of Theorem V.1, we have

$$\lambda_0 \geq \frac{D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)}{\alpha - 1},$$

where α is as defined in Theorem IV.8.

Proof. We recall the definition of α from Theorem IV.8 which ensures that $\mathbb{E}_{X\sim\Pi}[r(X)^2] \leq \alpha$, or $\mathrm{Var}_{X\sim\Pi}[r(X)] = \mathbb{E}_{X\sim\Pi}[r(X)^2] - 1 \leq \alpha - 1$, or $\frac{D_{\mathrm{KL}}(P\|\Pi)+D_{\mathrm{KL}}(\Pi\|P)}{\mathrm{Var}_{X\sim\Pi}[r(X)]} \geq \frac{D_{\mathrm{KL}}(P\|\Pi)+D_{\mathrm{KL}}(\Pi\|P)}{\alpha-1}$. Combining this with Theorem V.1, we get the required result. □

Theorem V.3. Under the assumptions of Theorem IV.8, for FedPriKL with a trusted aggregator, we have the following:

1) If $D_{\mathrm{KL}}(P\|\Pi)+D_{\mathrm{KL}}(\Pi\|P) \geq \frac{\ln(1.25/\delta)\ln 4}{(T\varepsilon)^2 N\pi_{\min}}$, the optimal $\lambda_0 \in \mathbb{R}_{\geq 0}$ is s.t.

$$\lambda_0 = \frac{D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P) - \frac{\ln(1.25/\delta)\ln 4}{(T\varepsilon)^2 N\pi_{\min}}}{\mathrm{Var}[r(X)] + \frac{2\ln(1.25/\delta)}{(T\varepsilon N\pi_{\min})^2}}.$$

2) If $\mathrm{Var}_{X\sim\Pi}[r(X)] < \frac{2\ln(1.25/\delta)}{(T\varepsilon N\pi_{\min})^2}$, the optimal $\lambda_0 \in \mathbb{R}_{<0}$ is s.t.

$$\lambda_0 = \frac{D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P) + \frac{\ln(1.25/\delta)\ln 4}{(T\varepsilon)^2 N\pi_{\min}}}{\mathrm{Var}_{X\sim\Pi}[r(X)] - \frac{2\ln(1.25/\delta)}{(T\varepsilon N\pi_{\min})^2}}.$$

Corollary V.4. $|\lambda_0|$ for FedPriKL with a trusted aggregator with DP guarantee is less than $|\lambda_0|$ for FedPriKL when there is no DP noise involved.

VI. Experimental evaluation

We implemented FedPriKL and the baseline technique from Section IV. We also compare against the results of computing the estimator Φ (*cf.*, Section III-D) via `SecAgg` without adding any DP noise. We refer to this as "No privacy" to understand the effect of privacy noise distortion. We performed our evaluation on multiple distributions drawn from the FEMNIST dataset, a benchmark from the FL community [32]. The theoretical guarantees of FedPriKL are distribution-independent subject to the stated support conditions. We therefore use FEMNIST as a controlled federated benchmark to test the finite-sample behavior predicted by the analysis: dependence on the privacy budget ε, estimator parameter λ, sample count m, client-batch size, and divergence magnitude. The objective of these experiments is thus validation of the estimator's theoretical and systems trade-offs. The data is a collection of images of handwritten digits 0-9, where the image data is partitioned based on the writer of the digit. In all experiments, clients are partitioned into disjoint batches, as specified in Algorithm 2. To generate a meaningful distribution for each digit, each 28×28 image is partitioned into sixteen 7×7 sub-images, with the mean intensity of each sub-image computed and mapped to a binary level, and the probabilities of encountering each of the 2^{16} possibilities listed. In total there are over 400,000 digits from over 3,500 writers. We primarily evaluate the accuracy of the result using the (mean) absolute error (AE) between the ground truth KL divergence and that found by the different estimators. The mean is taken over different repetitions of the test with different random choices, and over different distributions induced by pairs of digits. By default, we set $\varepsilon = 1.0$ and $\delta = 10^{-6}$. Experiments are run in a Jupyter notebook on a commodity laptop with Intel i7-1260P processor and 16GB memory. See https://figshare.com/s/cbb3d58ad7e42785405d for code.

A. Choice of λ

Our first experiments study the impact of the parameter λ on the performance of FedPriKL, complementing the theoretical analysis of the optimal λ presented above. Recall that λ is used to control the variance of the estimator, and we seek a choice of λ that will reliably achieve a small error across a range of scenarios. The plots in Figure 2 show results across a choice of values of privacy parameter ε while sampling clients into batches of 5% of the total population (error bands show the standard deviation over 10 repetitions). We see that the estimator with no privacy is able to obtain a consistently good average error close to 0.2. The values of KL divergence on this data range between 0.25 to 2.0, meaning that this accuracy is sufficient to accurately distinguish small from large values.

We use this experiment to determine how best to set the parameter λ. For FedPriKL, the best choice of λ is always close to 0, consistent with our analysis in Section V. At this

choice of λ, the observed empirical absolute error is close to that of the estimator without privacy. Meanwhile, the baseline approach of adding noise directly to the data is optimized by a value of λ below 0, and achieves a lower level of accuracy.

A similar pattern is seen in Figure 3, where we vary λ, but this time hold ε constant at 0.5, and instead vary the number of clients in each batch, from 2% to 10%. For FEDPRIKL, the best result is consistently obtained at $\lambda = 0$, where it is close to that with no privacy noise. The main effect of varying the batch size is in how quickly the accuracy decreases as λ is varied away from 0. Meanwhile, the baseline approach achieves weaker accuracy and is highly dependent on the choice of λ. Hereafter, we use a default value of $\lambda = 0$ for FEDPRIKL, since this appears to be a preferred choice, and $\lambda = -1$ for the baseline approach.

B. Performance of FEDPRIKL models

Next, we vary the number of samples m used to build each individual estimator. Figure 4 shows results for different choices of ε. We see that there is relatively weak dependency on this parameter: although error decreases slowly as m increases, choosing $m = 10$ is sufficient to obtain close to the best accuracy. Thus, the application-level contribution from each client consists only of counts for a small number of sampled indices. For $m = 10$, this payload is below 1 KB in our representation, making FEDPRIKL suitable for lightweight clients.

Finally, we consider how the accuracy performs for different inputs. We consider the KL divergences induced by three pairs of digits: the pair with the lowest KL divergence (3 and 8) of 0.24; the median KL divergence (4 and 1) of 0.76; and the maximum divergence (9 and 0) of 2.0. The results for these are shown in Figure 5 as we vary ε from 0.1 to 1.5.

Across all these cases, we see that FEDPRIKL obtains accuracy that is close to that of the non-private estimator when ε is moderate in value (0.5 to 1), whereas the baseline approach only becomes competitive for the largest KL divergences when ε is large. We also observe that the error is lower for smaller divergences. In the case of the closest pair, the absolute error is close to zero. This means that the estimators can accurately distinguish between small and large divergences, with error that appears to scale with the magnitude. Taken together, these experiments test FEDPRIKL across orthogonal sources of difficulty (privacy level, sampling budget, client participation, and distribution separation) within a fixed benchmark, allowing the effects predicted by the analysis to be isolated.

Experimental summary. The empirical study confirms that FEDPRIKL achieves a practical tradeoff between privacy and accuracy, and is comparable to its non-private version. Concretely, we recommend setting the variance control parameter $\lambda = 0$ and number of samples to $m = 10$, since these choices are seen to be robust in practice.

VII. Concluding Remarks

We have proposed FEDPRIKL, to our knowledge the first method for estimating the KL divergence between a public reference distribution and a private, federated data distribution under formal differential privacy guarantees. Our estimator is unbiased with provably bounded sensitivity and variance, and its per-client payload is independent of the domain size, remaining below 1 KB even for domains with 2^{16} elements. Experiments on FEMNIST show that FEDPRIKL matches the accuracy of its non-private counterpart at moderate privacy budgets ($\varepsilon \in [0.5, 1]$) and substantially outperforms a baseline in which clients perturb their reports locally, providing accurate estimates to inform distribution-drift and retraining decision. Future directions include removing the trusted aggregator in favor of fully distributed computation, extending the estimator to broader divergence families, supporting continual drift monitoring over data streams, and making the core protocol robust against malicious participants.

References

[1] J. F. McCarthy, K. A. Marx, P. E. Hoffman, A. G. Gee, P. O'Neil, M. L. Ujwal, and J. Hotchkiss, "Applications of machine learning and high-dimensional visualization in cancer detection, diagnosis, and management," *Ann. NY Acad. Sci.*, vol. 1020, no. 1, pp. 239–262, 2004.

[2] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J. Big Data*, vol. 2, pp. 1–21, 2015.

[3] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in ML*, vol. 14, no. 1-2, pp. 1–210, 2021.

[4] A. Bharadwaj and G. Cormode, "An introduction to federated computation," in *ACM SIGMOD*, 2022, pp. 2448–2451.

[5] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *ACM CCS*, 2016.

[6] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *ACM CCS*, 2017.

[7] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[8] C. Dwork and A. Roth, "Algorithmic foundations of differential privacy." *Foundations and Trends in TCS*, vol. 9, no. 3-4, pp. 211–407, 2014.

[9] S. Guha, P. Indyk, and A. McGregor, "Sketching information divergences," in *COLT*, 2007, pp. 424–438.

[10] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[11] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, 2006.

[12] K. Kenthapadi, A. Korolova, I. Mironov, and N. Mishra, "Privacy via the Johnson-Lindenstrauss transform," *J. Priv. Confidentiality*, vol. 5, no. 1, pp. 39–71, 2013.

[13] J. Blocki, A. Blum, A. Datta, and O. Sheffet, "The Johnson-Lindenstrauss transform itself preserves differential privacy," in *IEEE FOCS*, 2012.

[14] C. Hou, H. Zhan, A. Shrivastava, S. Wang, A. Livshits, G. Fanti, and D. Lazar, "Privately customizing prefinetuning to better match user data in federated learning," arXiv:2302.09042, 2023.

[15] F. Nielsen, K. Sun, and S. Marchand-Maillet, "On Hölder projective divergences," *Entropy*, vol. 19, no. 3, p. 122, 2017.

[16] A. Andoni, K. D. Ba, P. Indyk, and D. Woodruff, "Efficient sketches for earth-mover distance, with applications," in *IEEE FOCS*, 2009.

[17] M. Heikkilä, J. Jälkö, O. Dikmen, and A. Honkela, "Differentially private markov chain monte carlo," *NeurIPS*, vol. 32, pp. 1–11, 2019.

[18] M. Heikkilä, E. Lagerspetz, S. Kaski, K. Shimizu, S. Tarkoma, and A. Honkela, "Differentially private Bayesian learning on distributed data," *NeurIPS*, vol. 30, pp. 1–10, 2017.

[19] D. Keeler, C. Komlo, E. Lepert, S. Veitch, and X. He, "DPrio: Efficient differential privacy with high utility for Prio," in *PoPETS*, 2023.

[20] H. Corrigan-Gibbs and D. Boneh, "Prio: Private, robust, and scalable computation of aggregate statistics," in *NSDI*, 2017, pp. 259–282.

[21] S. Addanki, K. Garbe, E. Jaffe, R. Ostrovsky, and A. Polychroniadou, "Prio+: Privacy preserving aggregate statistics via boolean shares," in *Security and Cryptography for Networks*, 2022, pp. 516–539.
[22] K. Bonawitz, P. Kairouz, B. McMahan, and D. R. Ramage, "Federated learning and privacy," *ACM Queue*, vol. 19, 2021.
[23] J. Kim, G. Park, M. Kim, and S. Park, "Cluster-based secure aggregation for federated learning," *Electronics*, vol. 12, no. 4, p. 870, 2023.
[24] A. Shamir, "How to share a secret," *CACM*, vol. 22, no. 11, pp. 612–613, 1979.
[25] A. Shapiro, "Monte carlo sampling methods," *Handbooks in Operations Research and Management Science*, vol. 10, pp. 353–425, 2003.
[26] S.-I. Amari, "α-divergence is unique, belonging to both f-divergence and Bregman divergence classes," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 4925–4931, 2009.
[27] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: What it is, and what it is not," in *IEEE TRUSTCOM*, 2015.
[28] C. Shepherd and K. Markantonakis, *Trusted Execution Environments*, 2024.
[29] R. Cramer, I. Damgård, and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing*, 2015.
[30] A. Aly and N. P. Smart, "Benchmarking privacy preserving scientific operations," in *Applied Cryptography and Network Security*, 2019.
[31] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *ACM STOC*, 2009, p. 169–178.
[32] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," arXiv:1812.01097, 2018.

ADDITIONAL PROOFS

This section of provides detailed proofs of the results omitted from the main paper.

A. Proof of Theorem IV.4

For distributions $P_1, P_2 \in \mathcal{P}^{>1}$ such that $P_1 \sim P_2$, there must be histograms $h_1, h_2 \in \mathcal{H}_{\mathcal{X}}^{\geq 1}$ with $h_1 \sim h_2$ satisfying $P_1 = \psi(h_1)$ and $P_2 = \psi(h_2)$. Thus, there must be some $x_1, x_2 \in \mathcal{X}$ such that $h_1(x_1) - h_2(x_1) = 1$, $h_1(x_2) - h_2(x_2) = -1$ and $h_1(x) = h_2(x) \, \forall \, x \in \mathcal{X} \setminus \{x_1, x_2\}$. Let $N = |h_1| = |h_2|$, $n_1 = h_1(x_1)$, and $n_2 = h_1(x_2)$. Because $h_1(x) = h_2(x)$ for $x \in \mathcal{X} \setminus \{x_1, x_2\}$, let the value of $h_1(x_1) + h_1(x_2) = h_2(x_1) + h_2(x_2)$ be denoted by k. Note that $P_1, P_2 \in \mathcal{P}^{>1}$ implies that $k \geq 4$ (because each value from $\mathcal{X}$ occurs at least twice). Therefore, we get

$$
\begin{aligned}
|f(P_1) - f(P_2)| &= \left| \sum_x \Pi(x) \ln \frac{\Pi(x)}{P_1(x)} - \sum_x \Pi(x) \ln \frac{\Pi(x)}{P_2(x)} \right| \\
&= \left| \sum_x \Pi(x) \ln \frac{P_2(x)}{P_1(x)} \right| \\
&= \left| \Pi(x_1) \ln \frac{h_2(x_1)}{h_1(x_1)} + \Pi(x_2) \ln \frac{h_2(x_2)}{h_1(x_2)} \right| \\
&= \left| \Pi(x_1) \ln \frac{n_1}{n_1 - 1} + \Pi(x_2) \ln \frac{n_2 + 1}{n_2} \right| \\
&= \left| \Pi(x_1) \ln \left(1 - \frac{1}{n_1}\right) + \Pi(x_2) \ln \left(1 + \frac{1}{k - n_1}\right) \right| \\
&\leq \left| \Pi(x_1) \ln \left(1 - \frac{1}{n_1}\right) \right| + \left| \Pi(x_2) \ln \left(1 + \frac{1}{k - n_1}\right) \right| \\
&= \Pi(x_1) \ln \left(1 + \frac{1}{n_1 - 1}\right) + \Pi(x_2) \ln \left(1 + \frac{1}{k - n_1}\right) \\
&\leq \Pi(x_1) \ln \left(1 + \frac{1}{n_1 - 1}\right) \\
&\quad + \Pi(x_2) \ln \left(1 + \frac{1}{(n_1 + 2) - n_1}\right) \\
&\quad \text{[as } k \geq n_1 + 2 \text{ \& } \ln \left(1 + \frac{1}{k - n_1}\right) \text{ decreases with } k\text{]} \\
&\leq \Pi(x_1) \ln \left(1 + \frac{1}{n_1 - 1}\right) + \Pi(x_2) \ln \frac{3}{2} \\
&\leq \Pi(x_1) \ln \left(1 + \frac{1}{2 - 1}\right) + \Pi(x_2) \ln \frac{3}{2} \\
&\quad \text{[as } n_1 \geq 2 \text{ \& } \ln \left(1 + \frac{1}{n_1 - 1}\right) \text{ decreases with } n_1\text{]} \\
&\leq \ln 2 (\Pi(x_1) + \Pi(x_2)) \leq \ln 2 < 0.7.
\end{aligned}
$$

□

B. Proof of Theorem IV.5

In order to prove this result, we derive the proof of a more generic result where the estimation of the KL divergence between Π and P_i uses the variance parameter λ_{P_i} for $i = 1, 2$ such that λ_{P_1} and λ_{P_2} may or may not be the same. In the special case where $\lambda_{P_1} = \lambda_{P_2} = \lambda$, we recover the setting specified in the theorem.

For $P_1, P_2 \in \mathcal{P}^{>1}$ with $P_1 \sim P_2$, we have $h_1, h_2 \in \mathcal{H}_{\mathcal{X}}^{\geq 1}$ with $h_1 \sim h_2$ satisfying $P_1 = \psi(h_1)$ and $P_2 = \psi(h_2)$. Thus, there are $x_1, x_2 \in \mathcal{X}$ such that $h_1(x_1) - h_2(x_1) = 1$, $h_1(x_2) - h_2(x_2) = -1$ and $h_1(x) = h_2(x) \, \forall \, x \in \mathcal{X} \setminus \{x_1, x_2\}$. Let $N = |h_1| = |h_2|$, $n_1 = h_1(x_1)$, and $n_2 = h_1(x_2)$. Note that, as $P_1, P_2 \in \mathcal{P}^{>1}$, $n_1, n_2 \geq 2$. Let $X = x \in \mathcal{X}$ be some realization of $X \sim \Pi$, and, for notational convenience, let $r_i = \frac{P_i(x)}{\Pi(x)}$ and $\lambda_i = \lambda_{P_i}$ for $i = 1, 2$ in the context of this proof. We obtain $|\hat{f}(P_1) - \hat{f}(P_2)|$

$$
\begin{aligned}
&= |(\lambda_1(r_1 - 1) - \ln r_1) - (\lambda_2(r_2 - 1) - \ln r_2)| \\
&= \left| \frac{\lambda_1 P_1(x) - \lambda_2 P_2(x)}{\Pi(x)} + (\lambda_1 - \lambda_2) + \ln \frac{P_2(x)}{P_1(x)} \right| \\
&\leq \left| \frac{\lambda_1 P_1(x) - \lambda_2 P_2(x)}{\Pi(x)} \right| + |\lambda_1 - \lambda_2| + \left| \ln \frac{P_2(x)}{P_1(x)} \right| \\
&\quad \text{[by triangle-inequality]} \\
&= \left| \frac{\lambda_1 P_1(x) - \lambda_2 P_2(x)}{\Pi(x)} \right| + |\lambda_1 - \lambda_2| \\
&\quad + \left| \ln \frac{h_2(x)}{h_1(x)} \right| \left[\text{as } P_i(x) = \frac{h_i(x)}{N} \text{ for } i = 1, 2 \right] \\
&\leq \left| \frac{\lambda_1 P_1(x) - \lambda_2 P_2(x)}{\Pi(x)} \right| + |\lambda_1 - \lambda_2| \\
&\quad + \max \left\{ \left| \ln \frac{n_1}{n_1 - 1} \right|, \left| \ln \frac{n_2 + 1}{n_2} \right| \right\} \\
&\quad \left[\text{as } h_1(x) = h_2(x) \forall x \neq x_1, x_2, \text{ or } \ln \frac{h_2(x)}{h_1(x)} = 0 \right] \\
&\leq \max \left\{ \frac{p|\lambda_1 - \lambda_2|}{\Pi(x)}, \frac{|\lambda_1 n_1 - \lambda_2(n_1 - 1)|}{\Pi(x_1)}, \frac{|\lambda_1 n_2 - \lambda_2(n_2 + 1)|}{N \Pi(x_2)} \right\} \\
&\quad + |\lambda_1 - \lambda_2| + \max \left\{ \ln \frac{n_1}{n_1 - 1}, \ln \frac{n_2 + 1}{n_2} \right\} \\
&\quad [\text{ where } p = h_1(x)/N = h_2(x)/N \forall x \in \mathcal{X} \setminus \{x_1, x_2\}] \\
&\leq \max \left\{ \hat{\alpha} |\lambda_1 - \lambda_2|, \frac{|n_1(\lambda_1 - \lambda_2) + \lambda_2|}{N \Pi(x_1)}, \frac{|n_2(\lambda_1 - \lambda_2) - \lambda_2|}{N \Pi(x_2)} \right\} \\
&\quad + |\lambda_1 - \lambda_2| + \max \left\{ \ln \frac{n_1}{n_1 - 1}, \ln \frac{n_2 + 1}{n_2} \right\} \\
&\leq \max \left\{ \hat{\alpha} |\lambda_1 - \lambda_2|, \frac{|n_1(\lambda_1 - \lambda_2)| + |\lambda_2|}{N \Pi(x_1)}, \frac{|n_2(\lambda_1 - \lambda_2)| + |\lambda_2|}{N \Pi(x_2)} \right\} \\
&\quad + |\lambda_1 - \lambda_2| + \max \left\{ \ln \frac{n_1}{n_1 - 1}, \ln \frac{n_2 + 1}{n_2} \right\} \\
&\leq \max \left\{ \hat{\alpha} |\lambda_1 - \lambda_2|, \hat{\alpha} |\lambda_1 - \lambda_2| + \frac{|\lambda_2|}{N \Pi(x_1)}, \right. \\
&\quad \left. \hat{\alpha} |\lambda_1 - \lambda_2| + \frac{|\lambda_2|}{N \Pi(x_2)} \right\} + |\lambda_1 - \lambda_2| \\
&\quad + \max \left\{ \ln \frac{n_1}{n_1 - 1}, \ln \frac{n_2 + 1}{n_2} \right\} \\
&= (\hat{\alpha} + 1) |\lambda_1 - \lambda_2| + \frac{|\lambda_2|}{N} \max \left\{ \frac{1}{\Pi(x_1)}, \frac{1}{\Pi(x_2)} \right\} \\
&\quad + \max \left\{ \ln \frac{n_1}{n_1 - 1}, \ln \frac{n_2 + 1}{n_2} \right\} \\
&\leq (\hat{\alpha} + 1) |\lambda_1 - \lambda_2| + \frac{|\lambda_2|}{N \pi_{\min}}
\end{aligned}
$$

$$
\begin{aligned}
&+\max\left\{\ln\frac{n_1}{n_1-1},\ln\frac{n_2+1}{n_2}\right\}\\
&\le(\hat{\alpha}+1)|\lambda_1-\lambda_2|+\frac{|\lambda_2|}{N\pi_{\min}}\\
&+\max\left\{\ln\left(1+\frac{1}{n_1-1}\right),\ln\left(1+\frac{1}{n_2}\right)\right\}\\
&\le(\hat{\alpha}+1)|\lambda_1-\lambda_2|+\frac{|\lambda_2|}{N\pi_{\min}}\\
&+\max\left\{\ln\left(1+\frac{1}{2-1}\right),\ln\left(1+\frac{1}{2}\right)\right\}\\
&=(\hat{\alpha}+1)|\lambda_1-\lambda_2|+\frac{|\lambda_2|}{N\pi_{\min}}+\ln 2.
\end{aligned}
$$

Substituting λ for λ_1 and λ_2 gives us the desired result. □

C. Proof of Theorem IV.6

By Theorem IV.5, for any $x\in\mathcal{X}$ and a chosen λ, the sensitivity of $\Phi_{\Pi,P}(r(x),\lambda)$, is bounded above by $\frac{|\lambda|}{\pi_{\min}m}+\ln 2$. As this upper bound is independent of T, let us denote this term as κ, a constant (w.r.t. T). Therefore, for *each* set of clients C_t in any batch t, for m instances of x sampled from Π, the total sensitivity of $\sum_{i_t=1}^{m}\Phi_{\Pi,P}(r(x_{i_t}),\lambda)$ is independently bounded above by $m\kappa$. Moreover, as $C_1,\ldots,C_T$ are all disjoint, $\sum_{t=1}^{T}\sum_{i_t=1}^{m}\Phi_{\Pi,P}(r(x_{i_t}),\lambda)$ is formally equivalent to computing $\sum_{i_t=1}^{m}\Phi_{\Pi,P}(r(x_{i_t}),\lambda)$ in T independent and parallel rounds over the disjoint sets of clients $C_1,\ldots,C_T$ and obtaining the final sum over $i_1,\ldots,i_T$. Hence, by the parallel compositionality property of DP, the total sensitivity of $\sum_{t=1}^{T}\sum_{i_t=1}^{m}\Phi_{\Pi,P}(r(x_{i_t}),\lambda)$ is bounded above by $m\kappa$. Thus, in order to show that the final estimator given by Algorithm 2 satisfies (ε,δ)-DP, by the definition of DP with additive Gaussian noise, it suffices to show $\eta_{\epsilon,\delta}\sim\mathcal{N}\left(0,\frac{\ln(1.25/\delta)\Delta\left(\frac{1}{mT}\sum_{t=1}^{T}\sum_{i_t=1}^{m}\Phi_{\Pi,P}(r(x_{i_t}),\lambda)\right)^2}{\epsilon^2}\right)$. But we obtain

$$
\begin{aligned}
&\Delta\left(\frac{1}{mT}\sum_{t=1}^{T}\sum_{i_t=1}^{m}\Phi_{\Pi,P}(r(x_{i_t}),\lambda)\right)\\
&=\frac{1}{mT}\Delta\left(\sum_{t=1}^{T}\sum_{i_t=1}^{m}\Phi_{\Pi,P}(r(x_{i_t}),\lambda)\right)\\
&=\frac{1}{mT}m\kappa=\frac{1}{T}\Delta\left(\hat{D}_{\mathrm{KL}}(\Pi\|P)_{[\lambda,T]}\right). \qquad (1)
\end{aligned}
$$

Hence, recalling that the noise $\eta_{\varepsilon,\delta}\sim\mathcal{N}\left(0,\frac{2\ln(1.25/\delta)\Delta\left(\hat{D}_{\mathrm{KL}}(\Pi\|P)_{[\lambda,T]}\right)^2}{(\varepsilon T)^2}\right)$, and using (1) in this expression, we obtain $\eta_{\epsilon,\delta}\sim\mathcal{N}\left(0,\frac{\ln(1.25/\delta)\Delta\left(\frac{1}{mT}\sum_{t=1}^{T}\sum_{i_t=1}^{m}\Phi_{\Pi,P}(r(x_{i_t}),\lambda)\right)^2}{\epsilon^2}\right)$, as needed. □

D. Proof of Theorem IV.7

$$
\begin{aligned}
&\mathbb{E}\left[\hat{D}_{\mathrm{KL}}\left(\Pi\|P\right)_{[\lambda,T,\varepsilon,\delta]}\right]\\
&=\mathbb{E}\left[\frac{1}{mT}\sum_{t=1}^{T}\sum_{i_t=1}^{m}\Phi'_{\Pi,P}(r(x_{i_t}),\lambda)\right]\\
&=\mathbb{E}\left[\frac{1}{mT}\sum_{t=1}^{T}\sum_{i_t=1}^{m}\Phi_{\Pi,P}(r(x_{i_t}),\lambda)\right]\text{[since }\mathbb{E}\left[\eta_{\varepsilon,\delta}\right]=0]\\
&\frac{1}{mT}\sum_{t=1}^{T}\sum_{i_t=1}^{m}\mathbb{E}\left[\Phi_{\Pi,P}(r(x_{i_t}),\lambda)\right]\\
&=\mathbb{E}_{X\sim\Pi}\left[\Phi_{\Pi,P}(r(X),\lambda)\right]=D_{\mathrm{KL}}(\Pi\|P)\\
&\text{[as }x_{i_t}\text{s are i.i.d. and using Facts IV.1 and IV.2].}
\end{aligned}
$$

□

E. Proof of Theorem IV.8

$$
\begin{aligned}
&\mathrm{Var}\left[\hat{D}_{\mathrm{KL}}\left(\Pi\|P\right)_{[\lambda,T,\varepsilon,\delta]}\right]=\mathrm{Var}\left[\frac{1}{mT}\sum_{t=1}^{T}\sum_{i=1}^{m}\Phi'_{\Pi,P}(r(x_i),\lambda)\right]\\
&=\mathrm{Var}\left[\frac{1}{mT}\sum_{t=1}^{T}\sum_{i=1}^{m}\Phi_{\Pi,P}(r(x_i),\lambda)\right]+\frac{1}{(mT)^2}mT\,\mathrm{Var}\left[\eta_{\varepsilon,\delta}\right]\\
&=\frac{1}{(mT)^2}\sum_{t=1}^{T}\sum_{i=1}^{m}\mathrm{Var}\left[\Phi_{\Pi,P}(r(x_i),\lambda)\right]+\frac{1}{mT}\mathrm{Var}\left[\eta_{\varepsilon,\delta}\right]\\
&=\frac{1}{(mT)^2}(mT)\,\mathrm{Var}_{X\sim\Pi}\left[\Phi_{\Pi,P}(r(X),\lambda)\right]+\frac{1}{mT}\mathrm{Var}\left[\eta_{\varepsilon,\delta}\right]\\
&[\text{as }x_1,\ldots x_m\overset{\text{i.i.d.}}{\sim}\Pi]\\
&=\frac{1}{mT}\mathrm{Var}_{X\sim\Pi}\left[\Phi_{\Pi,P}(r(X),\lambda)\right]+\frac{1}{mT}\mathrm{Var}\left[\eta_{\varepsilon,\delta}\right]\\
&=\frac{1}{mT}\left(\lambda^2\,\mathrm{Var}[r(X)-1]+\mathrm{Var}[\ln r(X)]\right.\\
&\left.-2\lambda\left(\mathrm{Cov}\left[(r(X)-1),\ln r(X)\right]\right)\right)+\frac{1}{mT}\mathrm{Var}\left[\eta_{\varepsilon,\delta}\right]\\
&=\frac{1}{mT}\left(\lambda^2\,\mathrm{Var}[r(X)]+\mathrm{Var}[\ln r(X)]\right.\\
&\left.-2\lambda\left(\mathbb{E}\left[(r(X)-1)\ln r(X)\right]-\mathbb{E}[r(X)-1]\mathbb{E}[\ln r(X)]\right)\right)\\
&+\frac{1}{mT}\mathrm{Var}\left[\eta_{\varepsilon,\delta}\right]\\
&=\frac{1}{mT}\left(\lambda^2\,\mathrm{Var}[r(X)]+\mathrm{Var}[\ln r(X)]\right.\\
&\left.-2\lambda\left(\mathbb{E}\left[(r(X)-1)\ln r(X)\right]\right)\right)+\frac{1}{mT}\mathrm{Var}\left[\eta_{\varepsilon,\delta}\right]\\
&[\text{as }\mathbb{E}_{X\sim\Pi}[r(X)-1]=0]\\
&=\frac{1}{mT}\left(\lambda^2\,\mathrm{Var}[r(X)]+\mathrm{Var}[\ln r(X)]\right.\\
&\left.-2\lambda\left(\mathbb{E}\left[(r(X)\ln r(X)\right]\right)+2\lambda\mathbb{E}\left[\ln r(X)\right]\right)\\
&+\frac{1}{mT}\mathrm{Var}\left[\eta_{\varepsilon,\delta}\right]\\
&=\frac{1}{mT}\left(\lambda^2\,\mathrm{Var}[r(X)]+\mathrm{Var}[\ln r(X)]\right.\\
&\left.-2\lambda\sum_{x\in\mathcal{X}}\frac{P(x)}{\Pi(x)}\ln\frac{P(x)}{\Pi(x)}\Pi(x)+2\lambda\sum_{x\in\mathcal{X}}\ln\frac{P(x)}{\Pi(x)}\Pi(x)\right)\\
&+\frac{1}{mT}\mathrm{Var}\left[\eta_{\varepsilon,\delta}\right]\\
&=\frac{1}{mT}\left(\lambda^2\,\mathrm{Var}[r(X)]+\mathrm{Var}[\ln r(X)]\right.\\
&\left.-2\lambda\left(D_{\mathrm{KL}}(P\|\Pi)+D_{\mathrm{KL}}(\Pi\|P)\right)\right)+\frac{1}{mT}\mathrm{Var}\left[\eta_{\varepsilon,\delta}\right]
\end{aligned}
$$

$$= \frac{1}{mT}\left(\lambda^2 \operatorname{Var}[r(X)] + \mathbb{E}[\ln^2 r(X)] - \mathbb{E}[\ln r(X)]^2\right.$$
$$\left.-2\lambda\left(D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)\right)\right) + \frac{1}{mT}\operatorname{Var}\left[\eta_{\varepsilon,\delta}\right] \tag{2}$$

Substituting $\mathbb{E}_{X\sim\Pi}[r(X)]^2 = \left(\sum_{x\in\mathcal{X}}\Pi(x)\ln\frac{P(X)}{\Pi(X)}\right)^2 = D_{\mathrm{KL}}(\Pi\|P)^2$ into (2), we obtain:

$$= \frac{1}{mT}\left(\lambda^2 \operatorname{Var}[r(X)] + \mathbb{E}[\ln^2 r(X)] - D_{\mathrm{KL}}(\Pi\|P)^2\right.$$
$$\left.-\ 2\lambda\left(D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)\right)\right) + \frac{1}{mT}\operatorname{Var}\left[\eta_{\varepsilon,\delta}\right]$$
$$= \frac{1}{mT}\left(\lambda^2\mathbb{E}[r(X)^2] - \lambda^2\mathbb{E}[r(X)]^2 + \mathbb{E}[\ln^2 r(X)]\right.$$
$$\left.-D_{\mathrm{KL}}(\Pi\|P)^2 - 2\lambda\left(D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)\right)\right)$$
$$+\ \frac{1}{mT}\operatorname{Var}\left[\eta_{\varepsilon,\delta}\right]$$
$$= \frac{1}{mT}\left(\lambda^2\mathbb{E}[r(X)^2] - \lambda^2 + \mathbb{E}[\ln^2 r(X)] - D_{\mathrm{KL}}(\Pi\|P)^2\right.$$
$$\left.-\ 2\lambda\left(D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)\right)\right) + \frac{1}{mT}\operatorname{Var}\left[\eta_{\varepsilon,\delta}\right] \tag{3}$$

Recalling that $1-\frac{1}{x} \le \ln x \le x-1\ \forall\, x\in\mathbb{R}_{\ge 0}$, we get

$$\ln^2(r(X)) \le \begin{cases}(r(X)-1)^2, & \text{if } r(X)\ge 1\\ \left(1-\frac{1}{r(X)}\right)^2, & \text{if } r\in(0,1).\end{cases}$$

Therefore, $\mathbb{E}\left[\ln^2(r(X))\right]$

$$\le \max\left\{\mathbb{E}\left[(r(X)-1)^2\right], \mathbb{E}\left[\left(1-\frac{1}{r(X)}\right)^2\right]\right\}$$
$$= \max\left\{\mathbb{E}\left[r(X)^2\right] + 1 - 2\,\mathbb{E}\left[r(X)\right],\right.$$
$$\left.1 + \mathbb{E}\left[\frac{1}{r(X)^2}\right] - 2\,\mathbb{E}\left[\frac{1}{r(X)}\right]\right\}$$
$$= \max\left\{\mathbb{E}\left[r(X)^2\right] - 1, 1 + \mathbb{E}\left[\frac{1}{r(X)^2}\right] - 2\,\mathbb{E}\left[\frac{1}{r(X)}\right]\right\}$$
$$\le \max\left\{\mathbb{E}\left[r(X)^2\right] - 1, 1 + \mathbb{E}\left[\frac{1}{r(X)^2}\right] - \frac{2}{\mathbb{E}\left[r(X)\right]}\right\}$$
[as $\mathbb{E}\left[\frac{1}{z}\right] \ge \frac{1}{\mathbb{E}[z]}\forall z\ 0$ by Jensen's inequality]
$$= \max\left\{\mathbb{E}\left[r(X)^2\right] - 1, \mathbb{E}\left[\frac{1}{r(X)^2}\right] - 1\right\}$$
$$\le \max\left\{\sum_{x\in\mathcal{X}}\frac{P(x)^2}{\Pi(x)^2}\Pi(x), \sum_{x\in\mathcal{X}}\frac{\Pi(x)^2}{P(x)^2}\Pi(x)\right\} - 1$$
$$= \max\left\{\sum_{x\in\mathcal{X}}\frac{P(x)^2}{\Pi(x)}, \sum_{x\in\mathcal{X}}\frac{\Pi(x)^3}{P(x)^2}\right\} - 1$$
$$\le \max\left\{\sum_{x\in\mathcal{X}}\alpha P(x), \sum_{x\in\mathcal{X}}\beta^2\Pi(x)\right\} - 1$$
$$= \max\left\{\alpha, \beta^2\right\} - 1. \tag{4}$$

Moreover, $\mathbb{E}_{X\sim\Pi}[r(X)^2] = \sum_{x\in\mathcal{X}}\Pi(x)\frac{P^2(X)}{\Pi^2(X)} = \sum_{x\in\mathcal{X}}\frac{P^2(X)}{\Pi(X)} \le \alpha$. Combining this and (4) with (3) gives us:

$$\operatorname{Var}\left[\hat{D}_{\mathrm{KL}}\left(\Pi\|P\right)_{[\lambda,T,\varepsilon,\delta]}\right]$$
$$\le \frac{1}{mT}\left(\lambda^2(\alpha-1) + \max\{\alpha,\beta^2\} - 1 - D_{\mathrm{KL}}(\Pi\|P)^2\right.$$
$$\left.-2\lambda\left(D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)\right) + \operatorname{Var}[\eta_{\varepsilon,\delta}]\right) \tag{5}$$

Recalling the definition of $\eta_{\varepsilon,\delta}$ and using Theorem 4.5, we get

$$\operatorname{Var}[\eta_{\varepsilon,\delta}] \le \frac{2\ln(1.25/\delta)\left(\frac{|\lambda|}{N\pi_{\min}} + \ln 2\right)^2}{(\varepsilon T)^2}. \tag{6}$$

Combining (5) and (6), we obtain the desired result. □

F. Proof of Theorem V.1

Proof. Using Equation (3) in the proof of Theorem IV.8, we have $\operatorname{Var}\left[\hat{D}_{\mathrm{KL}}\left(\Pi\|P\right)_{[\lambda,T,\varepsilon,\delta]}\right] =$

$$\frac{1}{mT}\left(\lambda^2\mathbb{E}[r(X)^2] - \lambda^2 + \mathbb{E}[\ln^2 r(X)] - D_{\mathrm{KL}}(\Pi\|P)^2\right.$$
$$\left.-2\lambda\left(D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)\right)\right) + \frac{1}{mT}\operatorname{Var}\left[\eta_{\varepsilon,\delta}\right]. \tag{7}$$

Recalling that the setting of this theorem assumes no DP noise has been added to the estimator, implying that $\eta_{\varepsilon,\delta} = 0$, we essentially have $\operatorname{Var}[\eta_{\varepsilon,\delta}] = 0$. Hence, for fixed P, Π, m, and T, we note that Equation (7) is a differentiable function w.r.t. λ. Accordingly, we write $H : \mathbb{R} \mapsto \mathbb{R}$ as the corresponding auxiliary function w.r.t. λ s.t.

$$H(\lambda) = \frac{1}{mT}\left(\lambda^2\mathbb{E}[r(X)^2] - \lambda^2 + \mathbb{E}[\ln^2 r(X)]\right.$$
$$\left.-D_{\mathrm{KL}}(\Pi\|P)^2 - 2\lambda\left(D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)\right)\right).$$

Recalling Fact IV.1, which essentially ensures that $\mathbb{E}[r(X)^2] - 1$ equals $\operatorname{Var}[r(X)]$, we crucially observe that $\mathbb{E}[r(X)^2] \ge 1$, and thus conclude that $H(\lambda)$ is convex w.r.t. λ.

Therefore, to find the optimal λ_0, it suffices to find the solutions in $\lambda = \lambda_0$ such that $\frac{dH}{d\lambda}(\lambda_0) = 0$. For simplicity, let us write $\zeta_\Pi(X) = \mathbb{E}_{X\sim\Pi}[r(X)^2] - 1$. We start by computing

$$\frac{dH}{d\lambda} = \frac{1}{mT}\left(2\lambda\zeta_\Pi(X) - 2\left(D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)\right)\right). \tag{8}$$

Now we equate Equation (8) with 0 to get

$$\lambda_0 = \frac{D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)}{\zeta_\Pi(X)}. \tag{9}$$

Next, observing that $\zeta_\Pi(X) = \operatorname{Var}_{X\sim\Pi}[r(X)]$, we get the desired result. □

G. Proof of Theorem V.3

Proof. Shadowing the proof of Theorem V.1, we use Equation (7) to define an auxiliary function $G : \mathbb{R} \to \mathbb{R}$ in terms of λ, which incorporates the noise term $\eta_{\varepsilon,\delta}$. Thus,

$$G(\lambda) = \frac{1}{mT}\left(\lambda^2 \operatorname{Var}[r(X)] + \mathbb{E}[\ln^2 r(X)]\right.$$
$$\left.-D_{\mathrm{KL}}(\Pi\|P)^2 - 2\lambda\left(D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)\right)\right)$$
$$+\ \frac{1}{mT}\operatorname{Var}\left[\eta_{\varepsilon,\delta}\right].$$

Using Theorem IV.8, we deduce that

$$\frac{1}{mT}\operatorname{Var}\left[\eta_{\varepsilon,\delta}\right] = \frac{1}{mT^3\varepsilon^2} 2\ln(1.25/\delta)\left(\frac{|\lambda|}{N\pi_{\min}} + \ln 2\right)^2.$$

We substitute the above expression for $\frac{1}{mT}\operatorname{Var}\left[\eta_{\varepsilon,\delta}\right]$ in $G(\lambda)$ to get

$$\begin{aligned} G(\lambda) &= \frac{1}{mT}\left(\lambda^2 \operatorname{Var}[r(X)] + \mathbb{E}[\ln^2 r(X)] - D_{\mathrm{KL}}(\Pi\|P)^2\right. \\ &\left. -2\lambda\left(D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)\right)\right) \\ &+ \frac{1}{mT^3\varepsilon^2} 2\ln(1.25/\delta)\left(\frac{|\lambda|}{N\pi_{\min}} + \ln 2\right)^2. \end{aligned}$$

Hereby, we divide our analysis in two cases: *Case 1:* $\lambda \geq 0$, and *Case 2:* $\lambda < 0$. Individually, both cases ensure that $G(\lambda)$ is convex and differentiable w.r.t. λ. Hence, in order to find the optimal λ_0 for each case, we need to find the solutions in λ that make $\frac{dG}{d\lambda} = 0$.

Case 1. $\lambda \geq 0$:

$$\begin{aligned} G(\lambda) &= \frac{1}{mT}\left(\lambda^2 \operatorname{Var}[r(X)] + \mathbb{E}[\ln^2 r(X)]\right. \\ &\left. -D_{\mathrm{KL}}(\Pi\|P)^2 - 2\lambda\left(D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)\right)\right) \\ &+ \frac{1}{mT^3\varepsilon^2} 2\ln(1.25/\delta)\left(\frac{\lambda}{N\pi_{\min}} + \ln 2\right)^2 \end{aligned}$$

Hence,

$$\begin{aligned} \frac{dG}{d\lambda} &= \frac{1}{mT}\left(2\lambda \operatorname{Var}[r(X)]\right. \\ &\left. -2\left(D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)\right)\right) \\ &+ \frac{4\ln(1.25/\delta)}{mT^3\varepsilon^2}\left(\frac{\lambda}{N\pi_{\min}} + \ln 2\right)\frac{1}{N\pi_{\min}}. \end{aligned} \tag{10}$$

Now we equate Equation (10) with 0 to get

$$\begin{aligned} &\lambda_0\left(\operatorname{Var}_{X\sim\Pi}[r(X)] + \frac{2\ln(1.25/\delta)}{(T\varepsilon N\pi_{\min})^2}\right) \\ &= D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P) - \frac{2\ln(1.25/\delta)\ln 2}{(T\varepsilon)^2 N\pi_{\min}} \\ &\text{or, } \lambda_0 = \frac{D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P) - \frac{\ln(1.25/\delta)\ln 4}{(T\varepsilon)^2 N\pi_{\min}}}{\operatorname{Var}_{X\sim\Pi}[r(X)] + \frac{2\ln(1.25/\delta)}{(T\varepsilon N\pi_{\min})^2}}. \end{aligned}$$

Noting that all the terms in the denominator are non-negative, in line with the premise of Case 1., if $D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P) \geq \frac{\ln(1.25/\delta)\ln 4}{(T\varepsilon)^2 N\pi_{\min}}$, we have the desired optimal variance parameter in $\mathbb{R}_{\geq 0}$.

Case 2. $\lambda < 0$:

$$\begin{aligned} G(\lambda) &= \frac{1}{mT}\left(\lambda^2 \operatorname{Var}[r(X)] + \mathbb{E}[\ln^2 r(X)] - D_{\mathrm{KL}}(\Pi\|P)^2\right. \\ &\left. -2\lambda\left(D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)\right)\right) \\ &- \frac{1}{mT^3\varepsilon^2} 2\ln(1.25/\delta)\left(\frac{\lambda}{N\pi_{\min}} + \ln 2\right)^2 \end{aligned}$$

Hence,

$$\begin{aligned} \frac{dG}{d\lambda} &= \frac{1}{mT}\left(2\lambda \operatorname{Var}[r(X)]\right. \\ &\left. -2\left(D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)\right)\right) \\ &- \frac{4\ln(1.25/\delta)}{mT^3\varepsilon^2}\left(\frac{\lambda}{N\pi_{\min}} + \ln 2\right)\frac{1}{N\pi_{\min}} \end{aligned} \tag{11}$$

Now we equate Equation (11) with 0 to get

$$\begin{aligned} &\lambda_0\left(\operatorname{Var}_{X\sim\Pi}[r(X)] - \frac{2\ln(1.25/\delta)}{(T\varepsilon N\pi_{\min})^2}\right) \\ &= D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P) + \frac{2\ln(1.25/\delta)\ln 2}{(T\varepsilon)^2 N\pi_{\min}} \\ &\text{or, } \lambda_0 = \frac{D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P) + \frac{\ln(1.25/\delta)\ln 4}{(T\varepsilon)^2 N\pi_{\min}}}{\operatorname{Var}_{X\sim\Pi}[r(X)] - \frac{2\ln(1.25/\delta)}{(T\varepsilon N\pi_{\min})^2}}. \end{aligned} \tag{12}$$

We observe that all the terms in the numerator are non-negative. Thus, in line with the premise of Case 2., if $\operatorname{Var}_{X\sim\Pi}[r(X)] < \frac{2\ln(1.25/\delta)}{(T\varepsilon N\pi_{\min})^2}$, we have the desired optimal parameter in $\mathbb{R}_{<0}$.

□

H. Proof of Corollary V.4

Proof. Let $\tilde{D}(\Pi, P) = D_{\mathrm{KL}}(P\|\Pi) + D_{\mathrm{KL}}(\Pi\|P)$ and $\omega = \frac{\ln(1.25/\delta)\ln 4}{(T\varepsilon)^2 N\pi_{\min}}$. Then, using Theorem V.1, the optimal variance parameter for FedPriKL without DP noise reduces down to $\frac{\tilde{D}(\Pi,P)}{\zeta_\Pi(X)}$. For comparing this with the optimal noise parameter for FedPriKL with DP noise, we divide our analysis into two cases:

Case 1. $\lambda_0 \in \mathbb{R}_{\geq 0}$: Under the assumptions of Theorem V.3, the optimal λ_0 in the domain of $\mathbb{R}_{\geq 0}$ can be written as $\frac{\tilde{D}(\Pi,P)-\omega}{\zeta_\Pi(X)+\omega\ell}$ where $\zeta_\Pi(X)$ is as defined in the proof of Theorem V.1 and $\ell = (N\pi_{\min}\ln 2)^{-1}$. Thus, comparing the numerators and the denominators, it immediately follows that $\frac{\tilde{D}(\Pi,P)-\omega}{\zeta_\Pi(X)+\omega\ell} < \frac{\tilde{D}(\Pi,P)}{\zeta_\Pi(X)}$. As we are operating in the domain of non-negative reals for λ_0, we have $|\lambda_0| = \lambda_0$, and, thus, the desired result follows.

Case 2. $\lambda_0 \in \mathbb{R}_{<0}$: Under the assumptions of Theorem V.3, the optimal λ_0 in the domain of $\mathbb{R}_{<0}$ can be written as $\frac{\tilde{D}(\Pi,P)+\omega}{\zeta_\Pi(X)-\omega\ell}$ where $\zeta_\Pi(X)$, ω, and ℓ are same as above. Thus, comparing the numerators and the denominators, it immediately follows that $\frac{\tilde{D}(\Pi,P)+\omega}{\zeta_\Pi(X)-\omega\ell} > \frac{\tilde{D}(\Pi,P)}{\zeta_\Pi(X)}$. But as we are in the domain of negative reals for λ_0, we have $|\lambda_0| = -\lambda_0$ and, hence, $\left|\frac{\tilde{D}(\Pi,P)+\omega}{\zeta_\Pi(X)-\omega\ell}\right| < \left|\frac{\tilde{D}(\Pi,P)}{\zeta_\Pi(X)}\right|$, leading to the desired result.

□