

Article

ABCDP: Approximate Bayesian Computation with Differential Privacy

Mijung Park ^{1,*}, Margarita Vinaroz ^{2,3} and Wittawat Jitkrittum ⁴ 

- ¹ Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, Canada
² Max Planck Institute for Intelligent Systems, 72076 Tübingen, Germany; mvinaroz@tuebingen.mpg.de
³ Department of Computer Science, University of Tübingen, 72076 Tübingen, Germany
⁴ Google Research, 80636 Munich, Germany; wittawat@google.com
* Correspondence: mijungp@cs.ubc.ca

Abstract: We developed a novel approximate Bayesian computation (ABC) framework, *ABCDP*, which produces differentially private (DP) and approximate posterior samples. Our framework takes advantage of the sparse vector technique (SVT), widely studied in the differential privacy literature. SVT incurs the privacy cost only when a condition (whether a quantity of interest is above/below a threshold) is met. If the condition is sparsely met during the repeated queries, SVT can drastically reduce the cumulative privacy loss, unlike the usual case where every query incurs the privacy loss. In ABC, the quantity of interest is the distance between observed and simulated data, and only when the distance is below a threshold can we take the corresponding prior sample as a posterior sample. Hence, applying SVT to ABC is an organic way to transform an ABC algorithm to a privacy-preserving variant with minimal modification, but yields the posterior samples with a high privacy level. We theoretically analyzed the interplay between the noise added for privacy and the accuracy of the posterior samples. We apply ABCDP to several data simulators and show the efficacy of the proposed framework.



Citation: Park, M.; Vinaroz, M.; Jitkrittum, W. ABCDP: Approximate Bayesian Computation with Differential Privacy. *Entropy* **2021**, *23*, 961. <https://doi.org/10.3390/e23080961>

Academic Editor: Pierre Alquier

Received: 19 May 2021
Accepted: 20 July 2021
Published: 27 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: approximate Bayesian computation (ABC); differential privacy (DP); sparse vector technique (SVT)

1. Introduction

Approximate Bayesian computation (ABC) aims to identify the posterior distribution over simulator parameters. The posterior distribution is of interest as it provides the mechanistic understanding of the stochastic procedure that directly generates data in many areas such as climate and weather, ecology, cosmology, and bioinformatics [1–4].

Under these complex models, directly evaluating the likelihood of data is often intractable given the parameters. ABC resorts to an approximation of the likelihood function using simulated data that are *similar* to the actual observations.

In the simplest form of ABC called *rejection ABC* [5], we proceed by sampling multiple model parameters from a prior distribution $\pi: \theta_1, \theta_2, \dots \sim \pi$. For each θ_t , a pseudo dataset Y_t is generated from a simulator (the forward sampler associated with the intractable likelihood $P(y|\theta)$). The parameter θ_t for which the generated Y_t are similar to the observed Y^* , as decided by $\rho(Y_t, Y^*) < \epsilon_{abc}$, is accepted. Here, ρ is a notion of distance, for instance, L2 distance between Y_t and Y^* in terms of a pre-chosen summary statistic. Whether the distance is small or large is determined by ϵ_{abc} , a *similarity threshold*. The result is samples $\{\theta_t\}_{t=1}^M$ from a distribution, $\tilde{P}_\epsilon(\theta|Y^*) \propto \pi(\theta)\tilde{P}_\epsilon(Y^*|\theta)$, where $\tilde{P}_\epsilon(Y^*|\theta) = \int_{B_\epsilon(Y^*)} P(Y|\theta)dY$ and $B_\epsilon(Y^*) = \{Y : \rho(Y, Y^*) < \epsilon_{abc}\}$. As the likelihood computation is approximate, so is the posterior distribution. Hence, this framework is named by *approximate* Bayesian computation, as we do not compute the likelihood of data explicitly.

Most ABC algorithms evaluate the data similarity in terms of summary statistics computed by an aggregation of individual datapoints [6–11]. However, this seemingly

innocuous step of similarity check could impose a privacy threat, as aggregated statistics could still reveal an individual's participation to the dataset with the help of combining other publicly available datasets (see [12,13]). In addition, in some studies, the actual observations are privacy-sensitive in nature, e.g., Genotype data for estimating tuberculosis transmission parameters [14]. Hence, it is necessary to privatize the step of similarity check in ABC algorithms.

In this light, we introduce an ABC framework that obeys the notion of *differential privacy*. The differential privacy definition provides a way to quantify the amount of information that the distance computed on the privacy-sensitive data contains, whether or not a single individual's data are included (or modified) in the data [15]. Differential privacy also provides rigorous privacy guarantees in the presence of *arbitrary side information* such as similar public data.

A common form of applying DP to an algorithm is by adding noise to outputs of the algorithm, called *output perturbation* [16]. In the case of ABC, we found that *adding noise to the distance* computed on the real observations and pseudo-data suffices for the privacy guarantee of the resulting posterior samples. However, if we choose to simply add noise to the distance in every ABC inference step, this DP-ABC inference imposes an additional challenge due to the *repeated* use of the real observations. The *composition* property of differential privacy states that the privacy level degrades over the repeated use of data. To overcome this challenge, we adopt the *sparse vector technique* (SVT) [17], and apply it to the rejection ABC paradigm. The SVT outputs *noisy* answers of whether or not a stream of queries is above a certain threshold, where privacy cost incurs only when the SVT outputs at most c "above threshold" answers. This is a significant saving in privacy cost, as arbitrarily many "below threshold" answers are privacy cost free.

We name our framework, which combines ABC with SVT, as *ABCDP* (approximate Bayesian computation with differential privacy). Under ABCDP, we theoretically analyze the effect of noise added to the distance in the resulting posterior samples and the subsequent posterior integrals. Putting together, we summarize our main contributions:

1. We provide a novel ABC framework, ABCDP, which combines the *sparse vector technique* (SVT) [17] with the rejection ABC paradigm. The resulting ABCDP framework can improve the trade-off between the privacy and accuracy of the posterior samples, as the privacy cost under ABCDP is a function of the number of *accepted* posterior samples only.
2. We theoretically analyze ABCDP by focusing on the effect of noisy posterior samples in terms of two quantities. The first quantity provides the probability of an output of ABCDP being different from that of ABC at any given time during inference. The second quantity provides the convergence rate, i.e., how fast the posterior integral using ABCDP's noisy samples' approaches that using non-private ABC's samples. We write both quantities as a function of added noise for privacy to better understand the characteristics of ABCDP.
3. We validate our theory in the experiments using several simulators. The results of these experiments are consistent with our theoretical findings on the flip probability and the average error induced by the noise addition for privacy.

Unlike other existing ABC frameworks that typically rely on a pre-specified set of summary statistics, we use a kernel-based distance metric called *maximum mean discrepancy* following K2-ABC [18] to eliminate the necessity of pre-selecting a summary statistic. Using a kernel for measuring similarity between two empirical distributions was also proposed in K-ABC [19]. K-ABC formulates ABC as a problem of estimating a conditional mean embedding operator mapping (induced by a kernel) from summary statistics to corresponding parameters. However, unlike our algorithm, K-ABC still relies on a particular choice of summary statistics. In addition, K-ABC is a soft-thresholding ABC algorithm, while ours is a rejection-ABC algorithm.

To avoid the necessity of pre-selecting summary statistics, one could resort to methods that automatically or semi-automatically learn the best summary statistics given in a

dataset, and use the learned summary statistics in our ABCDP framework. An example is semi-auto ABC [6], where the authors suggest using the posterior mean of the parameters as a summary statistic. Another example is the indirect-score ABC [20], where the authors suggest using an auxiliary model which determines a score vector as a summary statistic. However, the posterior mean of the parameters in semi-auto ABC as well as the parameters of the auxiliary model in indirect-score ABC need to be estimated. The estimation step can incur a further privacy loss if the real data need to be used for estimating them. Our ABCDP framework does not involve such an estimation step and is more economical in terms of privacy budget to be spent than semi-auto ABC and indirect-score ABC.

2. Background

We start by describing relevant background information.

2.1. Approximate Bayesian Computation

Given a set Y^* containing observations, **rejection ABC** [5] yields samples from an approximate posterior distribution by repeating the following three steps:

$$\theta \sim \pi(\theta), \quad (1)$$

$$Y = \{y_1, y_2, \dots\} \sim P(y|\theta), \quad (2)$$

$$P_{\epsilon_{abc}}(\theta|Y^*) \sim P_{\epsilon_{abc}}(Y^*|\theta)\pi(\theta), \quad (3)$$

where the pseudo dataset Y is compared with the observations Y^* via:

$$P_{\epsilon_{abc}}(Y^*|\theta) = \int_{B_{\epsilon_{abc}}(Y^*)} P(Y|\theta) dY, \\ B_{\epsilon_{abc}}(Y^*) = \{Y|\rho(Y, Y^*) \leq \epsilon_{abc}\}, \quad (4)$$

where ρ is a divergence measure between two datasets. Any distance metric can be used for ρ . For instance, one can use the L2 distance under two datasets in terms of a pre-chosen set of summary statistics, i.e., $\rho(Y, Y^*) = D(S(Y), S(Y^*))$, with an L2 distance measure D on the statistics computed by S .

A more statistically sound choice for ρ would be *maximum mean discrepancy* (MMD, [21]) as used in [18]. Unlike a pre-chosen finite dimensional summary statistic typically used in ABC, MMD compares two distributions in terms of all the possible moments of the random variables described by the two distributions. Hence, ABC frameworks using the MMD metric such as [18] can avoid the problem of non-sufficiency of a chosen summary statistic that may occur in many ABC methods. For this reason, in this paper, we demonstrate our algorithm using the MMD metric. However, other metrics can be used as we illustrated in our experiments.

Maximum Mean Discrepancy

Assume that the data $Y \subset \mathcal{X}$ and let $k: \mathcal{X} \times \mathcal{X}$ be a positive definite kernel. MMD between two distributions P, Q is defined as

$$\text{MMD}^2(P, Q) = \mathbb{E}_{x, x' \sim P} k(x, x') + \mathbb{E}_{y, y' \sim Q} k(y, y') - 2\mathbb{E}_{x \sim P} \mathbb{E}_{y \sim Q} k(x, y). \quad (5)$$

By following the convention in kernel literature, we call MMD^2 simply MMD.

The Moore–Aronszajn theorem states that there is a unique Hilbert space \mathcal{H} on which k defines an inner product. As a result, there exists a feature map $\phi: \mathcal{X} \rightarrow \mathcal{H}$ such that $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}} = \langle \cdot, \cdot \rangle$ denotes the inner product on \mathcal{H} . The MMD in (5) can be written as

$$\text{MMD}^2(P, Q) = \|\mathbb{E}_{x \sim P}[\phi(x)] - \mathbb{E}_{y \sim Q}[\phi(y)]\|_{\mathcal{H}}^2$$

where $\mathbb{E}_{x \sim P}[\phi(x)] \in \mathcal{H}$ is known as the (kernel) mean embedding of P , and exists if $\mathbb{E}_{x \sim P} \sqrt{k(x, x)} < \infty$ [22]. The MMD can be interpreted as the distance between the mean embeddings of the two distributions. If k is a characteristic kernel [23], then $P \mapsto \mathbb{E}_{x \sim P}[\phi(x)]$ is injective, implying that $\text{MMD}(P, Q) = 0$, if and only if $P = Q$. When P, Q are observed through samples $X_m = \{x_i\}_{i=1}^m \sim P$ and $Y_n = \{y_i\}_{i=1}^n \sim Q$, MMD can be estimated by empirical averages [21] (Equation (3)): $\widehat{\text{MMD}}^2(X_m, Y_n) = \frac{1}{m^2} \sum_{i,j=1}^m k(x_i, x_j) + \frac{1}{n^2} \sum_{i,j=1}^n k(y_i, y_j) - \frac{2}{mn} \sum_{i=1}^m \sum_{j=1}^n k(x_i, y_j)$. When applied in the ABC setting, one input to $\widehat{\text{MMD}}$ is the observed dataset Y^* and the other input is a pseudo dataset $Y_t \sim p(\cdot | \theta_t)$ generated by the simulator given $\theta_t \sim \pi(\theta)$.

2.2. Differential Privacy

An output from an algorithm that takes in sensitive data as input will naturally contain some information of the sensitive data \mathcal{D} . The goal of differential privacy is to augment such an algorithm so that useful information about the population is retained, while sensitive information such as an individual's participation in the dataset cannot be learned [17]. A common way to achieve these two seemingly paradoxical goals is by deliberately injecting a controlled level of random noise to the to-be-released quantity. The modified procedure, known as a DP mechanism, now gives a stochastic output due to the injected noise. In the DP framework, a higher level of noise provides stronger privacy guarantee at the expense of less accurate population-level information that can be derived from the released quantity. Less noise added to the output thus reveals more about an individual's presence in the dataset.

More formally, given a mechanism \mathcal{M} (a *mechanism* takes a dataset as input and produces stochastic outputs) and neighboring datasets $\mathcal{D}, \mathcal{D}'$ differing by a single entry (either by replacing one's datapoint with another, or by adding/removing a datapoint to/from \mathcal{D}), the *privacy loss* of an outcome o is defined by

$$L^{(o)} = \log \frac{\mathbb{P}(\mathcal{M}(\mathcal{D}) = o)}{\mathbb{P}(\mathcal{M}(\mathcal{D}') = o)}. \quad (6)$$

The mechanism \mathcal{M} is called ϵ -DP if and only if $|L^{(o)}| \leq \epsilon$, for all possible outcomes o and for all possible neighboring datasets $\mathcal{D}, \mathcal{D}'$. The definition states that a single individual's participation in the data does not change the output probabilities by much; this limits the amount of information that the algorithm reveals about any one individual. A weaker or an *approximate* version of the above notion is (ϵ, δ) -DP: \mathcal{M} is (ϵ, δ) -DP if $|L^{(o)}| \leq \epsilon$, with probability $1 - \delta$, where δ is often called a failure probability which quantifies how often the DP guarantee of the mechanism fails.

Output perturbation is a commonly used DP mechanism to ensure the outputs of an algorithm to be differentially private. Suppose a deterministic function $h : \mathcal{D} \mapsto \mathbb{R}^p$ computed on sensitive data \mathcal{D} outputs a p -dimensional vector quantity. In order to make h private, we can add noise to the output of h , where the level of noise is calibrated to the *global sensitivity* [24], Δ_h , defined by the maximum difference in terms of some norm $\|h(\mathcal{D}) - h(\mathcal{D}')\|$ for neighboring \mathcal{D} and \mathcal{D}' (i.e., differ by one data sample).

There are two important properties of differential privacy. First, the *post-processing invariance* property [24] tells us that the composition of any arbitrary data-independent mapping with an (ϵ, δ) -DP algorithm is also (ϵ, δ) -DP. Second, the *composability* theorem [24] states that the strength of privacy guarantee degrades with the repeated use of DP-algorithms. Formally, given an ϵ_1 -DP mechanism \mathcal{M}_1 and an ϵ_2 -DP mechanism \mathcal{M}_2 , the mechanism $\mathcal{M}(\mathcal{D}) := (\mathcal{M}_1(\mathcal{D}), \mathcal{M}_2(\mathcal{D}))$ is $(\epsilon_1 + \epsilon_2)$ -DP. This composition is often-called *linear* composition, under which the total privacy loss linearly increases with the number of repeated use of DP-algorithms. The *strong* composition [17] [Theorem 3.20] improves the linear composition, while the resulting DP guarantee becomes weaker (i.e., approximate (ϵ, δ) -DP). Recently, more refined methods further improve the privacy loss (e.g., [25]).

2.3. AboveThreshold and Sparse Vector Technique

Among the DP mechanisms, we will utilize *AboveThreshold* and *sparse vector technique* (SVT) [17] to make the rejection ABC algorithm differentially private. AboveThreshold outputs 1 when a query value exceeds a pre-defined threshold, or 0 otherwise. This resembles rejection ABC where the output is 1 when the distance is less than a chosen threshold. To ensure the output is differentially private, AboveThreshold adds noise to both the threshold and the query value. We take the same route as AboveThreshold to make our ABCDP outputs differentially private. Sparse vector technique (SVT) consists of c calls to AboveThreshold, where c in our case determines how many posterior samples ABCDP releases.

Before presenting our ABCDP framework, we first describe the privacy setup we consider in this paper.

3. Problem Formulation

We assume a *data owner* who owns sensitive data Y^* and is willing to contribute to the posterior inference.

We also assume a *modeler* who aims to learn the posterior distribution of the parameters of a simulator. Our ABCDP algorithm proceeds with the two steps:

1. *Non-private step*: The modeler draws a parameter sample $\theta_t \sim \pi(\theta)$; then generates a pseudo-dataset Y_t , where $Y_t \sim P(y|\theta_t)$ for $t = 1, \dots, T$ for a large T . We assume these parameter-pseudo-data pairs $\{\theta_t, Y_t\}_{t=1}^T$ are publicly available (even to an adversary).
2. *Private step*: the data owner takes the whole sequence of parameter-pseudo-data pairs $\{(\theta_t, Y_t)\}_{t=1}^T$ and runs our ABCDP algorithm in order to output a set of *differentially private* binary indicators determining whether or not to accept each θ_t .

Note that T is the maximum number of parameter-pseudo-data pairs that are publicly available. We will run our algorithm for T steps, while our algorithm can terminate as soon as we output the c number of accepted posterior samples. So, generally, $c \ll T$. The details are then introduced.

4. ABCDP

Recall that the only place where the real data Y^* appear in the ABC algorithm is when we judge whether the simulated data are similar to the real data, i.e., as in (4). Our method hence adds noise to this step. In order to take advantage of the privacy analysis of SVT, we also add noise to the ABC threshold and to the ABC distance. Consequently, we introduce two perturbation steps.

Before we introduce them, we describe the global sensitivity of the distance, as this quantity tunes the amount of noise we will add in the two perturbation steps. For $\rho(Y^*, Y) = \widehat{\text{MMD}}(Y^*, Y)$ with a bounded kernel, then the sensitivity of the distance is $\Delta_\rho = O(1/N)$ as shown in Lemma 1.

Lemma 1 ($\Delta_\rho = O(1/N)$ for MMD). *Assume that Y^* and each pseudo dataset Y_t are of the same cardinality N . Set $\rho(Y^*, Y) = \widehat{\text{MMD}}(Y^*, Y)$ with a kernel k bounded by $B_k > 0$, i.e., $\sup_{x,y \in \mathcal{X}} k(x, y) \leq B_k < \infty$. Then:*

$$\sup_{(Y^*, Y^{*'}), Y} |\rho(Y^*, Y) - \rho(Y^{*'}, Y)| \leq \Delta_\rho := \frac{2}{N} \sqrt{B_k}$$

and $\sup_{Y^*, Y} \rho(Y^*, Y) \leq 2\sqrt{B_k}$.

A proof is given in Appendix B. For $\rho = \widehat{\text{MMD}}$ using a Gaussian kernel, $k(x, y) = \exp\left(-\frac{\|x-y\|^2}{2l^2}\right)$ where $l > 0$ is the bandwidth of the kernel, $B_k = 1$ for any $l > 0$.

Now, we introduce the two perturbation steps used in our algorithm summarized in Algorithm 1.

Algorithm 1 Proposed c -sample ABCDP

Require: Observations Y^* , Number of accepted posterior sample size c , privacy tolerance ϵ_{total} , ABC threshold ϵ_{abc} , distance ρ , and parameter-pseudo-data pairs $\{(\theta_t, Y_t)\}_{t=1}^T$, and option RESAMPLE.

Ensure: ϵ_{total} -DP indicators $\{\tilde{\tau}_t\}_{t=1}^T$ for corresponding samples $\{\theta_t\}_{t=1}^T$

```

1: Calculate the noise scale  $b$  by Theorem 1.
2: Privatize ABC threshold:  $\hat{\epsilon}_{abc} = \epsilon_{abc} + m_t$  via (7)
3: Set count=0
4: for  $t = 1, \dots, T$  do
5:   Privatize distance:  $\hat{\rho}_t = \rho(Y^*, Y_t) + v_t$  via (8)
6:   if  $\hat{\rho}_t \leq \hat{\epsilon}_{abc}$  then
7:     Output  $\tilde{\tau}_t = 1$ 
8:     count = count+1
9:     if RESAMPLE then
10:       $\hat{\epsilon}_{abc} = \epsilon_{abc} + m_t$  via (7)
11:    end if
12:   else
13:     Output  $\tilde{\tau}_t = 0$ 
14:   end if
15:   if count  $\geq c$  then
16:     Break the loop
17:   end if
18: end for

```

Step 1: Noise for privatizing the ABC threshold.

$$\hat{\epsilon}_{abc} = \epsilon_{abc} + m_t \quad (7)$$

where $m_t \sim \text{Lap}(b)$, i.e., drawn from the zero-mean Laplace distribution with a scale parameter b .

Step 2: Noise for privatizing the distance.

$$\hat{\rho}_t = \rho(Y^*, Y_t) + v_t \quad (8)$$

where $v_t \sim \text{Lap}(2b)$.

Due to these perturbations, Algorithm 1 runs with the privatized threshold and distance. We can choose to perturb the threshold only once, or every time we output 1 by setting RESAMPLE to false or true. After outputting c number of 1's, the algorithm is terminated. How do we calculate the resulting privacy loss under the different options we choose?

We formally state the relationship between the noise scale and the final privacy loss ϵ_{tot} for the Laplace noise in Theorem 1.

Theorem 1 (Algorithm 1 is ϵ_{total} -DP). *For any neighboring datasets $Y^*, Y^{*'}$ of size N and any dataset Y , assume that ρ is such that $0 < \sup_{(Y^*, Y^{*'}), Y} |\rho(Y^*, Y) - \rho(Y^{*'}, Y)| \leq \Delta_\rho < \infty$. Algorithm 1 is ϵ_{total} -DP, where:*

$$\epsilon_{total} = \begin{cases} \frac{(c+1)\Delta_\rho}{b} & \text{if RESAMPLE is False,} \\ \frac{2c\Delta_\rho}{b} & \text{if RESAMPLE is True.} \end{cases} \quad (9)$$

A proof is given in Appendix A. The proof uses linear composition, i.e., the privacy level linearly degrading with c . However, using the strong composition or more advanced compositions can reduce the resulting privacy loss, while these compositions turn pure-DP

into a weaker, approximate-DP. In this paper, we focus on the pure-DP. For the case of RESAMPLE = True, the proof directly follows the proof of the standard SVT algorithm using the linear composition method [17], with an exception that we utilize the quantity representing the minimum noisy value of any query evaluated on Y^* , as opposed to the maximum utilized in SVT. For the case of RESAMPLE = False, the proof follows the proof of Algorithm 1 in [26].

Note that the DP analysis in Theorem 1 holds for other types of distance metrics and not limited to only MMD, as long as there is a bounded sensitivity Δ_ρ under the chosen metric. When there is no bounded sensitivity, one could impose a clipping bound C to the distance by taking the distance from $\min[\rho(Y_t, Y^*), C]$, such that the resulting distance between any pseudo data Y_t and $Y^{*'}$ with modifying one datapoint in Y^* cannot exceed that clipping bound. In fact, we use this trick in our experiments when there is no bounded sensitivity.

4.1. Effect of Noise Added to ABC

Here, we would like to analyze the effect of noise added to ABC. In particular, we are interested in analyzing the probability that the output of ABCDP differs from that of ABC: $\mathbb{P}[\tilde{\tau}_t \neq \tau_t | \tau_t]$ at any given time t . To compute this probability, we first compute the probability density function (PDF) of the random variables $m_t - v_t$ in the following Lemma.

Lemma 2. Recall $m_t \sim \text{Lap}(b)$, $v_t \sim \text{Lap}(2b)$. The subtraction of these yields another random variable $Z = m_t - v_t$, where the PDF of Z is given by

$$f_Z(z) = \frac{1}{6b} \left[2 \exp\left(-\frac{|z|}{2b}\right) - \exp\left(-\frac{|z|}{b}\right) \right]. \quad (10)$$

Furthermore, for $a \geq 0$, $G_b(a) := \int_a^\infty f_Z(z) dz = \frac{1}{6} [4 \exp(-\frac{a}{2b}) - \exp(-\frac{a}{b})]$, and the CDF of Z is given by $F_Z(a) = H[a] + (1 - 2H[a])G_b(|a|)$ where $H[a]$ is the Heaviside step function.

See Appendix C for the proof. Using this PDF, we now provide the following proposition:

Proposition 1. Denote the output of Algorithm 1 at time t by $\tilde{\tau}_t \in \{0, 1\}$ and the output of ABC by $\tau_t \in \{0, 1\}$. The flip probability, the probability that the outputs of ABCDP and ABC differ given the output of ABC, is given by $P[\tilde{\tau}_t \neq \tau_t | \tau_t] = G_b(|\rho_t - \epsilon_{abc}|)$, where $G_b(a)$ is defined in Lemma 2, and $\rho_t := \rho(Y^*, Y_t)$.

See Appendix D for proof.

To provide an intuition of Proposition 1, we visualize the flip probability in Figure 1. This flip probability provides a guideline for choosing the accepted sample size c given the dataset size N and the desired privacy level ϵ_{total} . For instance, if a given dataset is extremely small, e.g., containing datapoints on the order of 10, c has to be chosen such that the flip probability of each posterior sample remains low for a given privacy guarantee (ϵ_{total}). If a higher number of posterior samples are needed, then one has to reduce the desired privacy level for the posterior sample of ABCDP to be similar to that of ABC. Otherwise, with a small ϵ_{total} with a large c , the accepted posterior samples will be poor. On the other hand, if the dataset is bigger, then a larger c can be taken for a reasonable level of privacy.

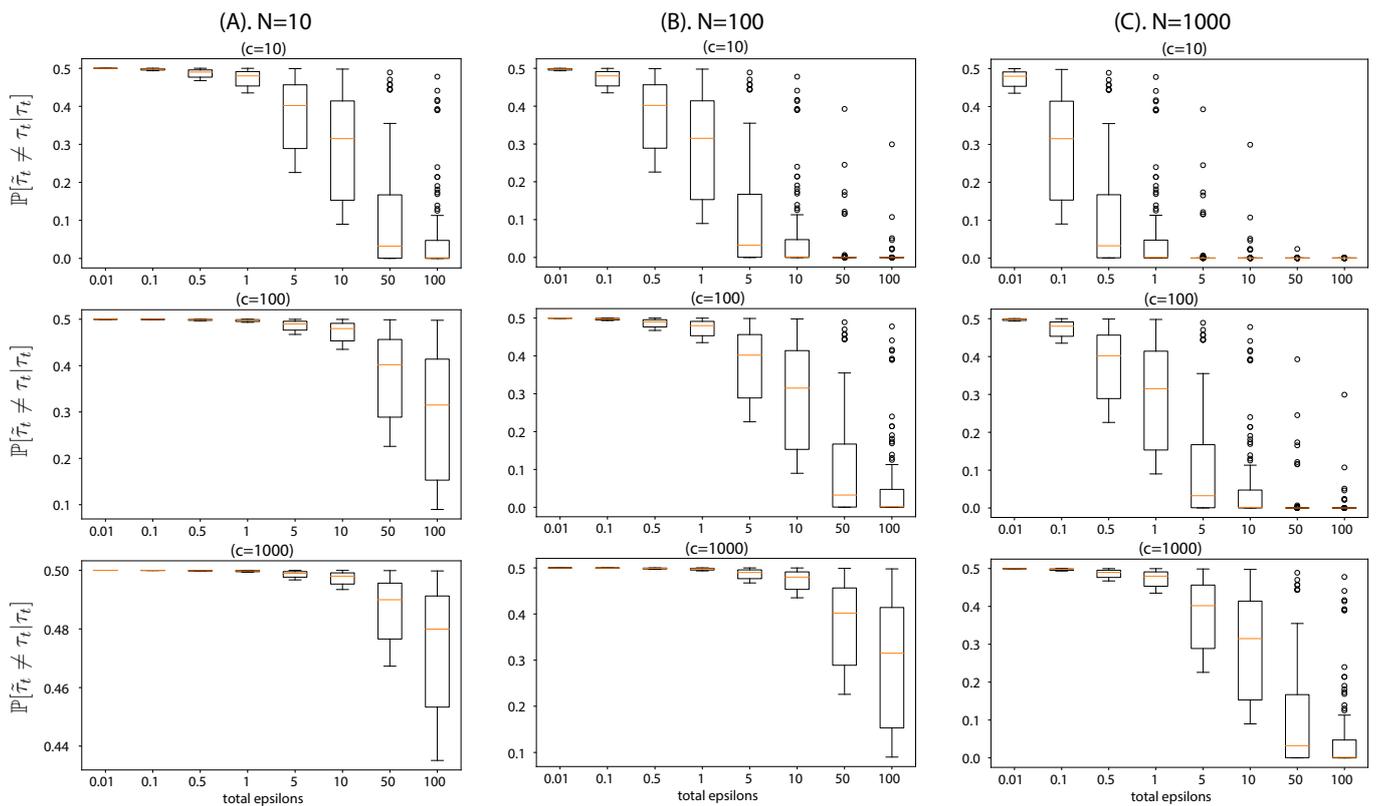


Figure 1. Visualization of flip probability derived in Proposition 1, the probability that the outputs of ABCDP and ABC differ given an output of ABC, with different dataset size N and accepted posterior sample size c . We simulated $\rho \sim \text{Uniform}[0, 1]$ (drew 100 values for ρ) and used $\epsilon_{abc} = 0.2$: (A) This column shows the flip probability at a regime of extremely small datasets, $N = 10$. Top plot shows the probability at $c = 10$, middle plot at $c = 100$, and bottom plot at $c = 1000$. In this regime, even $\epsilon_{total} = 100$ cannot reduce the flip probability to perfectly zero when $c = 10$. The flip probability remains high when we accept more samples, i.e., $c = 1000$; (B) the flip probability at $N = 100$; (C) the flip probability at $N = 1000$. As we increase the dataset size N (moving from the left to right columns), the flip probability approaches zero at a smaller privacy loss ϵ_{total} .

4.2. Convergence of Posterior Expectation of Rejection-ABCDP to Rejection-ABC.

The flip probability studied in Section 4.1 only accounts for the effect of noise added to a single output of ABCDP. Building further on this result, we analyzed the discrepancy between the posterior expectations derived from ABCDP and from the rejection ABC. This analysis requires quantifying the effect of noise added to the whole sequence of outputs of ABCDP. The result is presented in Theorem 2.

Theorem 2. Given Y^* of size N , and $\{(\theta_t, Y_t)\}_{t=1}^T$ as input, let $\tilde{\tau}_t \in \{0, 1\}$ be the output from Algorithm 1 where $\tilde{\tau}_t = 1$ indicates that (θ_t, Y_t) is accepted, for $t = 1, \dots, T$. Similarly, let τ_t denote the output from the traditional rejection ABC algorithm, for $t = 1, \dots, T$. Let f be an arbitrary vector-valued function of θ . Assume that the numbers of accepted samples from Algorithm 1, and the traditional rejection ABC algorithm are $c := \sum_{t=1}^T \tilde{\tau}_t \geq 1$ and $c' := \sum_{t=1}^T \tau_t \geq 1$, respectively. Let $b = \frac{4c\sqrt{B_k}}{\epsilon_{total}N}$ if RESAMPLE=True, and $b = \frac{2(c+1)\sqrt{B_k}}{\epsilon_{total}N}$ if RESAMPLE=False (see Theorem 1). Define $K_T := \max_{t=1, \dots, T} \|f(\theta_t)\|_2$. Then, the following statements hold for both RESAMPLE options:

1. $\mathbb{E}_{\tilde{\tau}_1, \dots, \tilde{\tau}_T} \left\| \frac{1}{c} \sum_{t=1}^T f(\theta_t) \tilde{\tau}_t - \frac{1}{c'} \sum_{t=1}^T f(\theta_t) \tau_t \right\|_2 \leq \frac{2K_T}{c'} \sum_{t=1}^T G_b(|\rho_t - \epsilon_{abc}|)$, where the decreasing function $G_b(x) \in (0, \frac{1}{2}]$ for any $x \geq 0$ is defined in Lemma 2;
2. $\mathbb{E}_{\tilde{\tau}_1, \dots, \tilde{\tau}_T} \left\| \frac{1}{c} \sum_{t=1}^T f(\theta_t) \tilde{\tau}_t - \frac{1}{c'} \sum_{t=1}^T f(\theta_t) \tau_t \right\|_2 \rightarrow 0$ as $N \rightarrow \infty$;

3. For any $a > 0$:

$$P\left(\left\|\frac{1}{c}\sum_{t=1}^T f(\boldsymbol{\theta}_t)\tilde{\tau}_t - \frac{1}{c'}\sum_{t=1}^T f(\boldsymbol{\theta}_t)\tau_t\right\|_2 \leq a\right) \geq 1 - \frac{4K_T}{3ac'}\sum_{t=1}^T \exp\left(-\frac{|\rho_t - \epsilon_{abc}|}{2b}\right)$$

where the probability is taken with respect to $\tilde{\tau}_1, \dots, \tilde{\tau}_T$.

Theorem 2 contains three statements. The first states that the expected error between the two posterior expectations of an arbitrary function f is bounded by a constant factor of the sum of the flip probability in each rejection/acceptance step. As we have seen in Section 4.1, the flip probability is determined by the scale parameter b of the Laplace distribution. Since $b = O(1/N)$ (see Theorem 1 and Lemma 1), it follows that the expected error decays as N increases, giving the second statement.

The third statement gives a probabilistic bound on the error. The bound guarantees that the error decays exponentially in N . Our proof relies on establishing an upper bound on the error as a function of the total number of flips $\sum_{t=1}^T |\tilde{\tau}_t - \tau_t|$ which is a random variable. Bounding the error of interest then amounts to characterizing the tail behavior of this quantity. Observe that in Theorem 2, we consider ABCDP and rejection ABC with the same computational budget, i.e., the same total number of iterations T performed. However, the number of accepted samples may be different in each case (c for ABCDP and c' for reject ABC). The fact that c itself is a random quantity due to injected noise presents its own technical challenge in the proof. Our proof can be found in Appendix E.

5. Related Work

Combining DP with ABC is relatively novel. The only related work is [27], which states that a rejection ABC algorithm produces posterior samples from the exact posterior distribution given perturbed data, when the kernel and bandwidth of rejection ABC are chosen in line with the data perturbation mechanism. The focus of [27] is to identify the condition when the posterior becomes exact in terms of the kernel and bandwidth of the kernel through the lens of data perturbation. On the other hand, we use the sparse vector technique to reduce the total privacy loss. The resulting theoretical studies including the flip probability and the error bound on the posterior expectation are new.

6. Experiments

6.1. Toy Examples

We start by investigating the interplay between ϵ_{abc} and ϵ_{total} , in a synthetic dataset where the ground truth parameters are known. Following [18], we also consider a symmetric Dirichlet prior π and a likelihood $p(y|\theta)$ given by a mixture of uniform distributions as

$$\begin{aligned}\pi(\theta) &= \text{Dirichlet}(\theta; \mathbf{1}), \\ P(y|\theta) &= \sum_{i=1}^5 \theta_i \text{Uniform}(y; [i-1, i]).\end{aligned}\quad (11)$$

A vector of mixing proportions is our model parameters θ , where the ground truth is $\theta^* = [0.25, 0.04, 0.33, 0.04, 0.34]^\top$ (see Figure 2). The goal is to estimate $\mathbb{E}[\theta|Y^*]$ where Y^* is generated with θ^* .

We first generated 5000 samples for Y^* drawn from (11) with true parameters θ^* . Then, we tested our two ABCDP frameworks with varying ϵ_{abc} and ϵ_{total} . In these experiments, we set $\rho = \widehat{\text{MMD}}$ with a Gaussian kernel. We set the bandwidth of the Gaussian kernel using the median heuristic computed on the simulated data (i.e., we did not use the real data for this, hence there is no privacy violation in this regard).

We drew 5000 pseudo-samples for Y_t at each time. We tested various settings, as shown in Figure 3, where we vary the number of posterior samples, $c = \{10, 100, 1000\}$,

$\epsilon_{abc} = \{0.05, 0.1, 0.2, 0.5\}$ and $\epsilon_{total} = \{0.5, 1.0, 10, \infty\}$. We showed the result of ABCDP for both RESAMPLE options in Figure 3.

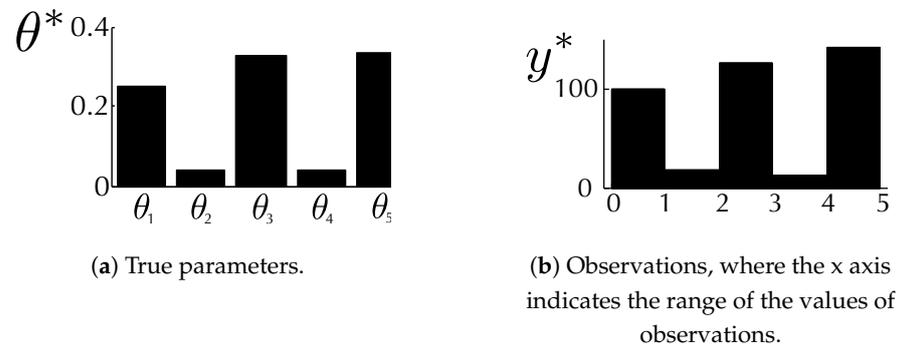


Figure 2. Synthetic data. (a): 5-dimensional true parameters; (b): observations sampled from the mixture of uniform distributions in (11) with the true parameters.

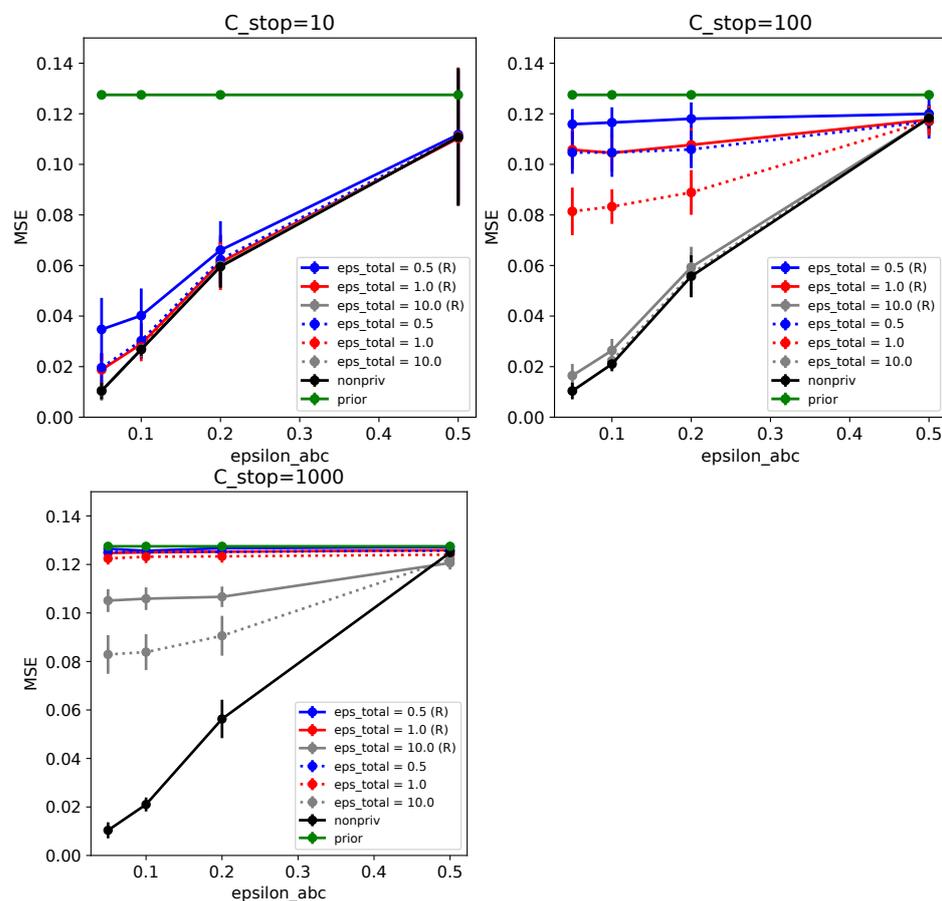


Figure 3. ABCDP on synthetic data. Mean-squared error (between true parameters and posterior mean) as a function of similarity threshold ϵ_{abc} given each privacy level. We ran ABCDP with the following options: *RESAMPLE = True* (denoted by R and solid line); or *RESAMPLE = False* (without R and dotted line) for 60 independent runs. (Top Left) When $c_{stop} = 10$ at different values of ϵ_{abc} , ABCDP and non-private ABC (black trace) achieved the highest accuracy (lowest MSE) at the smallest ϵ_{abc} ($\epsilon_{abc} = 0.01$). Notice that ABCDP *RESAMPLE = False* (dotted) outperformed ABCDP *RESAMPLE = True* (solid) for the same privacy tolerance (ϵ_{total}) at small values of ϵ_{abc} . (Top Right) MSE for $c_{stop} = 100$ at different values of ϵ_{abc} ; (Bottom Left) MSE for $c_{stop} = 1000$ at different values of ϵ_{abc} . We can observe when ϵ_{abc} is large, ABCDP (gray) marginally outperforms non-private ABC (black) due to the excessive noise added in ABCDP.

6.2. Coronavirus Outbreak Data

In this experiment, we modelled coronavirus outbreak in the Netherlands using a polynomial model consisting of four parameters a_0, a_1, a_2, a_3 , which we aimed to infer, where:

$$y(t) = a_3 + a_2t + a_1t^2 + a_0t^3. \quad (12)$$

The observed (<https://www.ecdc.europa.eu/en/publications-data/download-todays-data-geographic-distribution-COVID-19-cases-worldwide>, accessed on 10 October 2020) data are the number of cases of the coronavirus outbreak from 27 February to 17 March 2020, which amounts to 18 datapoints ($N = 18$). The presented experiment imposes privacy concern as each datapoint is a count of the individuals who are COVID positive at each time. The goal is to identify the approximate posterior distribution $\tilde{P}(a_0, a_1, a_2, a_3 | y^*)$ over these parameters, given a set of observations.

Recalling from Figure 1 that the small size of data worsens the privacy and accuracy trade-off, the inference is restricted to a small number of posterior samples (we chose $c = 5$) since the number of datapoints is extremely limited in this dataset. We used the same prior distributions for the four parameters as $a_i \sim \mathcal{N}(0, 1)$ for all $i = 0, 1, 2, 3$. We drew 50,000 samples from the Gaussian prior, and performed our ABCDP algorithm with $\epsilon_{total} = \{13, 22, 44\}$ and $\epsilon_{abc} = 0.1$, as shown in Figure 4.

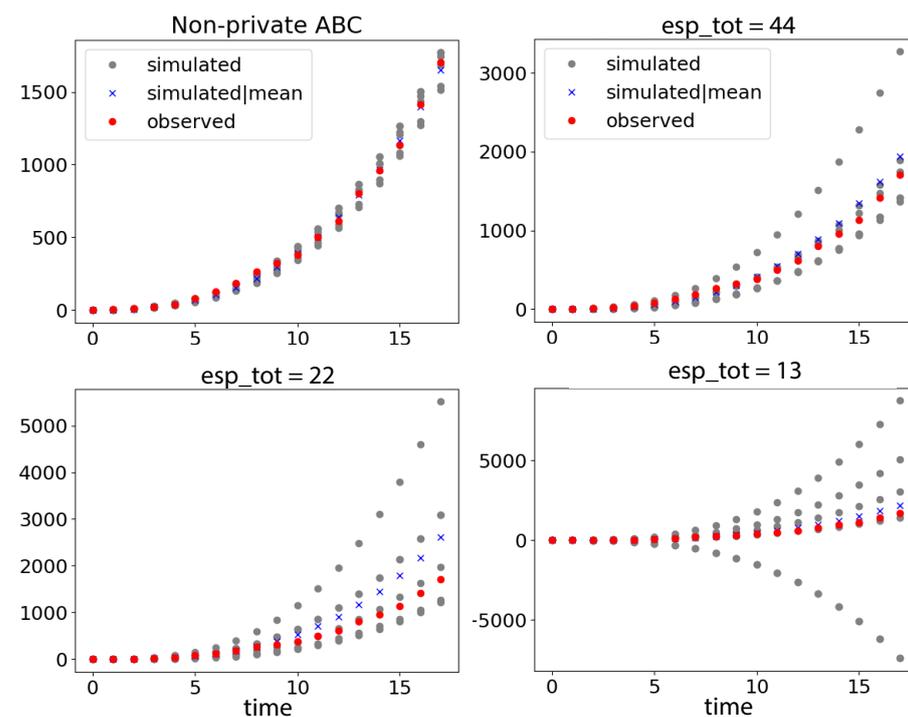


Figure 4. COVID-19 outbreak data ($N = 18$) and simulated data under different privacy guarantees. Red dots show observed data, and gray dots show simulated data drawn from 5 posterior samples accepted in each case. The blue crosses are simulated data given the posterior mean in each case: **(Top left)** simulated data by non-private ABC; **(Top right)** simulated data by ABCDP with $\epsilon_{total} = 44$ are relatively well aligned with regard to the extremely small size of the data. Note that we use a different scale for left and right plots for better visibility. If we use the same y scale in both plots, the simulated and observed points are not distinguishable on the left plot: **(Bottom left)** the simulated data given 5 posterior samples exhibit a large variance when $\epsilon_{total} = 22$; and **(Bottom right)** when $\epsilon_{total} = 13$, the simulated data exhibit an excessively large variance.

6.3. Modeling Tuberculosis (TB) Outbreak Using Stochastic Birth–Death Models

In this experiment, we used the stochastic birth–death models to model Tuberculosis (TB) outbreak. There are four parameters that we aim to infer, which go into the

communicable disease outbreak simulator as inputs: burden rate β , transmission rate t_1 , reproductive numbers R_1 and R_2 . The goal is to identify the approximate posterior distribution $\tilde{p}(R_1, t_1, R_2, \beta | y^*)$ over these parameters given a set of observations. Please refer to Section 3 in [28] for the description of the birth–death process of the model. We used the same prior distributions for the four parameters as in [28]: $\beta \sim \mathcal{N}(200, 30)$, $R_1 \sim \text{Unif}(1.01, 20)$, $R_2 \sim \text{Unif}(0.01, (1 - 0.05R_1)/0.95)$, $t_1 \sim \text{Unif}(0.01, 30)$.

To illustrate the privacy and accuracy trade-off, we first generated two sets of observations y^* ($n = 100$ and $n = 1000$) by some *true* model parameters (shown as black bars in Figure 5). We then tested our ABCDP algorithm with a privacy level $\epsilon = 1$. We used the summary statistic described in Table 1 in [28] and used a weighted L2 distance as ρ as done in [28], together with $\epsilon_{abc} = 150$. Since there is no bounded sensitivity in this case, we impose an artificial boundedness by clipping the distance by C (we set $C = 200$) when the distance goes beyond C .

As an error metric, we computed the mean absolute distance between each posterior mean and the true parameter values. The top row in Figure 5 shows that the mean of the prior (red) is far from the true value (black) that we chose. As we increase the data size from $n = 100$ (middle) to $n = 1000$ (bottom), the distance between true values and estimates reduces, as reflected in the error from 4.71 to 2.20 for RESAMPLE = True; and from 4.51 to 2.10 for RESAMPLE=False.

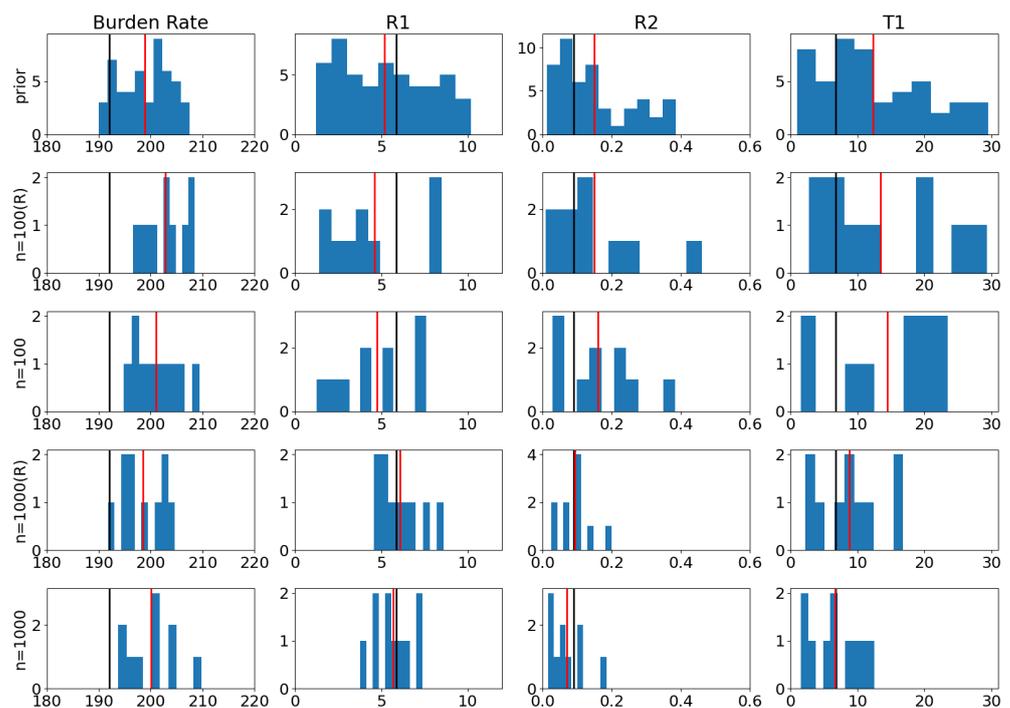


Figure 5. Posterior samples for modeling tuberculosis (TB) outbreak. In all ABCDP methods, we set $\epsilon_{total} = 1$. True values in black. Mean of samples in red: (R) indicates ABCDP with Resampling = True. (1st row): Histogram of 50 samples drawn from the prior (we used the same prior as [28]); (2nd row): 10 posterior samples from ABCDP with (R) given $n = 100$ observations; (3rd row): 10 posterior samples from ABCDP without (R) given $n = 100$ observations; (4th row): 10 posterior samples from ABCDP with (R) given $n = 1000$ observations; and (5th row): 10 posterior samples from ABCDP without (R) given $n = 1000$ observations. The distance between the black bar (true) and red bar (estimate) reduces as the size of data increases from 100 to 1000. ABCDP with Resampling=False performs better regardless of the data size.

7. Summary and Discussion

We presented the ABCDP algorithm by combining DP with ABC. Our method outputs differentially private binary indicators, yielding differentially private posterior samples.

To analyze the proposed algorithm, we derived the probability of flip from the rejection ABC's indicator to the ABCDP's indicator, as well as the average error bound of the posterior expectation.

We showed experimental results that output a relatively small number of posterior samples. This is due to the fact that the cumulative privacy loss increases linearly with the number of posterior samples (i.e., c) that our algorithm outputs. For a large-sized dataset (i.e., N is large), one can still increase the number of posterior samples while providing a reasonable level of privacy guarantee. However, for a small-sized dataset (i.e., N is small), a more refined privacy composition (e.g., [29]) would be necessary to keep the cumulative privacy loss relatively small, at the expense of providing an *approximate* DP guarantee rather than the pure DP guarantee that ABCDP provides.

When we presented our work to the ABC community, we often received the question of whether we could apply ABCDP to other types of ABC algorithms such as the sequential Monte Carlo algorithm which outputs the significance of each proposal sample, as opposed to its acceptance or rejection as in the rejection ABC algorithm. Directly applying the current form of ABCDP to these algorithms is not possible, while applying the Gaussian mechanism to the significance of each proposal sample can guarantee differential privacy for the output of the sequential Monte Carlo algorithm. However, the cumulative privacy loss will be relatively large, as now it is a function of the number of proposal samples, whether they are taken as good posterior samples or not.

A natural by-product of ABCDP is differentially private synthetic data, as the simulator is a public tool that anybody can run and hence differentially private posterior samples suffice for differentially private synthetic data without any further privacy cost. Applying ABCDP to generate complex datasets is an intriguing future direction.

Author Contributions: M.P. and W.J. contributed to conceptualization and methodology development. M.P. and M.V. contributed to software development. All three authors contributed to writing the paper. All authors have read and agreed to the published version of the manuscript.

Funding: Gips Schüle Foundation and the Institutional Strategy of the University of Tübingen (ZUK63).

Data Availability Statement: A publicly available dataset was analyzed in this study. This data can be found here: <https://www.ecdc.europa.eu/en/publications-data/download-todays-data-geographic-distribution-covid-19-cases-worldwide>.

Conflicts of Interest: The authors declare no conflict of interest.

Note: Our code is available at <https://github.com/ParkLabML/ABCDP>.

Appendix A. Proof of Theorem 1

Proof of Theorem 1. *Case I: RESAMPLE = True.* We prove the case of $c = 1$ first. The case of $c > 1$ is a c -composition of the case of $c = 1$, where the privacy loss linearly increases with c .

Given any neighboring datasets Y^* and $Y^{*'}$ of size N and any dataset Y , assume that ρ is such that $0 < \sup_{(Y^*, Y^{*'}), Y} |\rho(Y^*, Y) - \rho(Y^{*'}, Y)| < \Delta_\rho < \infty$ and ρ is bounded by B_ρ .

Let A denote the random variable that represents the outputs Algorithm 1 given $(\{\theta_t, Y_t\}_{t=1}^T, Y^*, \rho, \epsilon_{abc}, \epsilon)$ and A' the random variable that represents the outputs given $(\{\theta_t, Y_t\}_{t=1}^T, Y^{*'}, \rho, \epsilon_{abc}, \epsilon)$. The output of the algorithm is some realization of these variables, $\tau \in \{1, 0\}^k$ where $0 < k \leq T$ and for all $t < k$, $\tau_t = 0$ and $\tau_k = 1$. For the rest of the proof, we will fix the arbitrary values of ν_1, \dots, ν_{k-1} and take probabilities over the randomness of ν_k and ϵ_{abc} . We define the deterministic quantity (ν_1, \dots, ν_{k-1} are fixed):

$$g(Y^*) = \min_{t < k} (\rho(Y_t, Y^*) + \nu_t) \quad (\text{A1})$$

that represents the minimum noised value of the distance evaluated on any dataset Y^* .

Let $P[\hat{\epsilon}_{abc} = a]$ be the pdf of $\hat{\epsilon}_{abc}$ evaluated on a and $P[\nu_k = v]$ the pdf of ν_k evaluated on v , and $\mathbb{1}[x]$ the indicator function of event x . We have:

$$P_{\hat{\epsilon}_{abc}, \nu_k}[A = \tau_k] = P[\hat{\epsilon}_{abc} < g(Y^*)]$$

and:

$$\begin{aligned} \rho(Y_k, Y^*) + \nu_k \leq \hat{\epsilon}_{abc} &= P[\hat{\epsilon}_{abc} \in [\rho(Y_k, Y^*) + \nu_k, g(Y^*)]] \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P[\nu_k = v] P[\hat{\epsilon}_{abc} = a] \mathbb{1}[a \in [\rho(Y_k, Y^*) + \nu_k, g(Y^*)]] dv da \end{aligned}$$

Now, we define the following variables:

$$\begin{aligned} \hat{a} &= a + g(Y^*) - g(Y^{*'}) \\ \hat{v} &= v_s. + g(Y^*) - g(Y^{*'}) + \rho(Y_k, Y^{*'}) - \rho(Y_k, Y^*) \end{aligned}$$

We know that for each $Y^*, Y^{*'}$, ρ is Δ_ρ -sensitive and hence, the quantity $g(Y^*)$ is Δ_ρ -sensitive as well. In this way, we obtain that $|\hat{a} - a| \leq \Delta_\rho$ and $|\hat{v} - v_s. | \leq 2\Delta_\rho$. Applying these changes of variables, we have:

$$\begin{aligned} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P[\nu_k = \hat{v}] P[\hat{\epsilon}_{abc} = \hat{a}] \mathbb{1}[a + g(Y^*) - g(Y^{*'}) \in [v + g(Y^*) - g(Y^{*'}) + \\ &\quad \rho(Y_k, Y^{*'}), g(Y^*)]] dv da \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P[\nu_k = \hat{v}] P[\hat{\epsilon}_{abc} = \hat{a}] \mathbb{1}[a \in [v + \rho(Y_k, Y^{*'}), g(Y^*)]] dv da \\ &\leq \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp\left(\frac{\epsilon}{2}\right) P[\nu_k = v] \exp\left(\frac{\epsilon}{2}\right) P[\hat{\epsilon}_{abc} = a] \mathbb{1}[a \in [v + \rho(Y_k, Y^{*'}), g(Y^*)]] dv da \\ &\leq \exp(\epsilon) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P[\nu_k = v] P[\hat{\epsilon}_{abc} = a] \mathbb{1}[a \in [v + \rho(Y_k, Y^{*'}), g(Y^*)]] dv da \\ &= \exp(\epsilon) P[\hat{\epsilon}_{abc} < g(Y^{*'}) \text{ and } \rho(Y_k, Y^{*'}) + \nu_k \leq \hat{\epsilon}_{abc}] = \exp(\epsilon) P_{\hat{\epsilon}_{abc}, \nu_k}[A' = \tau_k] \end{aligned}$$

where the inequality comes from the bounds considered throughout the proof (i.e., $|\hat{a} - a| \leq \Delta_\rho$ and $|\hat{v} - v_s. | \leq 2\Delta_\rho$) and the form of the cdf for the Laplace distribution.

Case II: RESAMPLE = False. In this case, the proof follows the proof of Algorithm 1 in [26], with an exception that positive events for [26] become negative events for us and vice versa as we find the value below a threshold, where [26] finds the value above a threshold. \square

Appendix B. Proof of Lemma 1

Proof of Lemma 1. We will establish Δ_ρ when ρ is MMD. Recall that $(Y^*, Y^{*'})$ is a pair of neighboring datasets, and Y is an arbitrary dataset. Without loss of generality, assume that $Y^* = \{x_1, \dots, x_N\}$, $Y^{*'} = \{x'_1, \dots, x'_N\}$ such that $x_i = x'_i$ for all $i = 1, \dots, N - 1$, and $Y = \{y_1, \dots, y_m\}$. We start with:

$$\begin{aligned}
 & \sup_{(Y^*, Y^{*'}), Y} |\rho(Y^*, Y) - \rho(Y^{*'}, Y)| \\
 &= \sup_{(Y^*, Y^{*'}), Y} |\widehat{\text{MMD}}(Y^*, Y) - \widehat{\text{MMD}}(Y^{*'}, Y)| \\
 &= \sup_{(Y^*, Y^{*'}), Y} \left| \left\| \frac{1}{N} \sum_{i=1}^N \phi(x_i) - \frac{1}{m} \sum_{j=1}^m \phi(y_j) \right\|_{\mathcal{H}} - \left\| \frac{1}{N} \sum_{i=1}^N \phi(x'_i) - \frac{1}{m} \sum_{j=1}^m \phi(y_j) \right\|_{\mathcal{H}} \right| \\
 &\stackrel{(a)}{\leq} \sup_{(X, X')} \left\| \frac{1}{N} \sum_{i=1}^N \phi(x_i) - \frac{1}{N} \sum_{i=1}^N \phi(x'_i) \right\|_{\mathcal{H}} \\
 &= \sup_{(x_N, x'_N)} \left\| \frac{1}{N} \phi(x_N) - \frac{1}{N} \phi(x'_N) \right\|_{\mathcal{H}} \\
 &= \sup_{(x_N, x'_N)} \frac{1}{N} \sqrt{k(x_N, x_N) + k(x'_N, x'_N) - 2k(x_N, x'_N)} \\
 &\leq \frac{2}{N} \sqrt{B_k},
 \end{aligned}$$

where at (a), we use the reverse triangle inequality. Furthermore:

$$\begin{aligned}
 & \sup_{Y^*, Y} \rho(Y^*, Y) \\
 &\leq \sup_{Y^*, Y} \sqrt{\left\| \frac{1}{N} \sum_{i=1}^N \phi(x_i) - \frac{1}{m} \sum_{i=1}^m \phi(y_i) \right\|_{\mathcal{H}}^2} \\
 &= \sup_{Y^*, Y} \sqrt{\frac{1}{N^2} \sum_{i,j=1}^N k(x_i, x_j) + \frac{1}{m^2} \sum_{i,j=1}^m k(y_i, y_j) - \frac{2}{mn} \sum_{i=1}^N \sum_{j=1}^m k(x_i, y_j)} \\
 &= \sqrt{B_k + B_k + 2B_k} = 2\sqrt{B_k}.
 \end{aligned}$$

□

Appendix C. Proof of Lemma 2

Proof of Lemma 2. The PDF is computed from the convolution of two PDFs:

$$f_{m_t-v_t}(z) = \int_{-\infty}^{\infty} f_{m_t}(x) f_{v_t}(x-z) dx, \tag{A2}$$

where $f_{m_t}(x) = \frac{1}{2b} \exp(-\frac{|x|}{b})$ and $f_{v_t}(y) = \frac{1}{4b} \exp(-\frac{|y|}{2b})$:

$$f_{m_t-v_t}(z) = \frac{1}{8b^2} \int_{-\infty}^{\infty} \exp\left(-\frac{|x|}{b} - \frac{|x-z|}{2b}\right) dx \tag{A3}$$

For case $z \geq 0$:

$$f_{m_t - v_t}(z) = \frac{1}{8b^2} \int_{-\infty}^0 \exp\left(\frac{x}{b} + \frac{x-z}{2b}\right) dx + \frac{1}{8b^2} \int_0^z \exp\left(-\frac{x}{b} + \frac{x-z}{2b}\right) dx$$

$$+ \frac{1}{8b^2} \int_z^{\infty} \exp\left(-\frac{x}{b} - \frac{x-z}{2b}\right) dx, \quad (\text{A4})$$

$$= \frac{1}{8b^2} \int_{-\infty}^0 \exp\left(\frac{3x-z}{2b}\right) dx + \frac{1}{8b^2} \int_0^z \exp\left(\frac{-x-z}{2b}\right) dx$$

$$+ \frac{1}{8b^2} \int_z^{\infty} \exp\left(\frac{-3x+z}{2b}\right) dx \quad (\text{A5})$$

$$= \frac{\exp\left(\frac{-z}{2b}\right)}{8b^2} \int_{-\infty}^0 \exp\left(\frac{3x}{2b}\right) dx + \frac{\exp\left(\frac{-z}{2b}\right)}{8b^2} \int_0^z \exp\left(\frac{-x}{2b}\right) dx$$

$$+ \frac{\exp\left(\frac{z}{2b}\right)}{8b^2} \int_z^{\infty} \exp\left(\frac{-3x}{2b}\right) dx \quad (\text{A6})$$

$$= \frac{\exp\left(\frac{-z}{2b}\right) 2b}{8b^2} \frac{1}{3} - \frac{\exp\left(\frac{-z}{2b}\right) 2b}{8b^2} \left(\exp\left(\frac{-z}{2b}\right) - 1\right) + \frac{\exp\left(\frac{z}{2b}\right) 2b}{8b^2} \frac{1}{3} \exp\left(\frac{-3z}{2b}\right), \quad (\text{A7})$$

$$= \frac{1}{12b} \left[\exp\left(\frac{-z}{2b}\right) + 3 \exp\left(\frac{-z}{2b}\right) \left(1 - \exp\left(\frac{-z}{2b}\right)\right) + \exp\left(\frac{-z}{b}\right) \right], \quad (\text{A8})$$

$$= \frac{1}{12b} \left[4 \exp\left(\frac{-z}{2b}\right) - 2 \exp\left(\frac{-z}{b}\right) \right], \quad (\text{A9})$$

$$= \frac{1}{6b} \left[2 \exp\left(\frac{-z}{2b}\right) - \exp\left(\frac{-z}{b}\right) \right] \quad (\text{A10})$$

For case $z < 0$:

$$f_{m_t - v_t}(z) = \frac{1}{8b^2} \int_{-\infty}^z \exp\left(\frac{x}{b} + \frac{x-z}{2b}\right) dx + \frac{1}{8b^2} \int_z^0 \exp\left(\frac{x}{b} - \frac{x-z}{2b}\right) dx$$

$$+ \frac{1}{8b^2} \int_0^{\infty} \exp\left(-\frac{x}{b} - \frac{x-z}{2b}\right) dx, \quad (\text{A11})$$

$$= \frac{1}{8b^2} \int_{-\infty}^z \exp\left(\frac{3x-z}{2b}\right) dx + \frac{1}{8b^2} \int_z^0 \exp\left(\frac{x+z}{2b}\right) dx$$

$$+ \frac{1}{8b^2} \int_0^{\infty} \exp\left(\frac{-3x+z}{2b}\right) dx \quad (\text{A12})$$

$$= \frac{\exp\left(\frac{-z}{2b}\right)}{8b^2} \int_{-\infty}^z \exp\left(\frac{3x}{2b}\right) dx + \frac{\exp\left(\frac{z}{2b}\right)}{8b^2} \int_z^0 \exp\left(\frac{x}{2b}\right) dx$$

$$+ \frac{\exp\left(\frac{z}{2b}\right)}{8b^2} \int_0^{\infty} \exp\left(\frac{-3x}{2b}\right) dx \quad (\text{A13})$$

$$= \frac{\exp\left(\frac{-z}{2b}\right) 2b}{8b^2} \frac{1}{3} \exp\left(\frac{3z}{2b}\right) + \frac{\exp\left(\frac{z}{2b}\right) 2b}{8b^2} \left(1 - \exp\left(\frac{z}{2b}\right)\right) + \frac{\exp\left(\frac{z}{2b}\right) 2b}{8b^2} \frac{1}{3}, \quad (\text{A14})$$

$$= \frac{1}{12b} \left[\exp\left(\frac{z}{b}\right) - 3 \exp\left(\frac{z}{2b}\right) \left(\exp\left(\frac{z}{2b}\right) - 1\right) + \exp\left(\frac{z}{2b}\right) \right], \quad (\text{A15})$$

$$= \frac{1}{12b} \left[-2 \exp\left(\frac{z}{b}\right) + 4 \exp\left(\frac{z}{2b}\right) \right], \quad (\text{A16})$$

$$= \frac{1}{6b} \left[2 \exp\left(\frac{z}{2b}\right) - \exp\left(\frac{z}{b}\right) \right]. \quad (\text{A17})$$

With the obtained PDF $f_Z(z) = \frac{1}{6b} \left[2 \exp\left(-\frac{|z|}{2b}\right) - \exp\left(-\frac{|z|}{b}\right) \right]$, for $Z := m_t - v_t$, it is straightforward to compute $G_b(a) := \int_a^{\infty} f_Z(z) dz = \frac{1}{6} \left[4 \exp\left(-\frac{a}{2b}\right) - \exp\left(-\frac{a}{b}\right) \right]$ for $a \geq 0$. In other words, $G_b(a) = 1 - F_Z(a)$ for $a \geq 0$ where F_Z denotes the CDF of Z .

To show that the CDF of Z is $F_Z(a) = H[a] + (1 - 2H[a])G_b(|a|)$ where $H[a]$ is the Heaviside step function, we note that the density $f_Z(z)$ is an even function, i.e., $f_Z(z) = f_Z(-z)$ for any z . It follows that if $a < 0$, $1 - F_Z(a) = 1 - G_b(-a)$. This means that:

$$1 - F_Z(a) = \begin{cases} G_b(a) & \text{if } a \geq 0, \\ 1 - G_b(-a) & \text{if } a < 0, \end{cases}$$

or equivalently:

$$F_Z(a) = \begin{cases} 1 - G_b(a) & \text{if } a \geq 0, \\ G_b(-a) & \text{if } a < 0. \end{cases}$$

More concisely:

$$\begin{aligned} F_Z(a) &= (1 - G_b(|a|))\mathbb{I}[a \geq 0] + G_b(|a|)\mathbb{I}[a < 0] \\ &= \mathbb{I}[a \geq 0] + (\mathbb{I}[a < 0] - \mathbb{I}[a \geq 0])G_b(|a|) \\ &\stackrel{(a)}{=} H[a] + (1 - 2H[a])G_b(|a|), \end{aligned}$$

where at (a), we use $(\mathbb{I}[a < 0] - \mathbb{I}[a \geq 0]) = (1 - 2H[a])$. \square

Appendix D. Proof of Proposition 1

Proof of Proposition 1. Using this pdf above, we can compute the following probabilities:

$$P[\tilde{\tau}_t = 1 | \tau_t = 0] \tag{A18}$$

$$= P[0 \leq \rho_t - \epsilon_{abc} \leq Z], \tag{A19}$$

$$= \int_{\rho_t - \epsilon_{abc}}^{\infty} f(z) dz, \quad \text{where } \rho_t - \epsilon_{abc} \geq 0 \tag{A20}$$

$$= \int_{\rho_t - \epsilon_{abc}}^{\infty} \frac{1}{6b} \left[2 \exp\left(-\frac{|z|}{2b}\right) - \exp\left(-\frac{|z|}{b}\right) \right] dz, \text{ by definition of } f(z) \tag{A21}$$

$$= \int_{\rho_t - \epsilon_{abc}}^{\infty} \frac{1}{6b} \left[2 \exp\left(-\frac{z}{2b}\right) - \exp\left(-\frac{z}{b}\right) \right] dz, \quad \text{because } \rho_t - \epsilon_{abc} \geq 0 \tag{A22}$$

$$= \frac{1}{6b} \left[4b \exp\left(-\frac{\rho_t - \epsilon_{abc}}{2b}\right) - b \exp\left(-\frac{\rho_t - \epsilon_{abc}}{b}\right) \right], \tag{A23}$$

$$= \frac{1}{6} \left[4 \exp\left(-\frac{\rho_t - \epsilon_{abc}}{2b}\right) - \exp\left(-\frac{\rho_t - \epsilon_{abc}}{b}\right) \right], \text{ where } \rho_t - \epsilon_{abc} \geq 0, \tag{A24}$$

and:

$$\begin{aligned} P[\tilde{\tau}_t = 0 | \tau_t = 1] \\ = P[Z \leq \rho_t - \epsilon_{abc} \leq 0], \end{aligned} \tag{A25}$$

$$= \int_{-\infty}^{\rho_t - \epsilon_{abc}} f(z) dz, \text{ where } \rho_t - \epsilon_{abc} \leq 0, \tag{A26}$$

$$= \int_{-\infty}^{\rho_t - \epsilon_{abc}} \frac{1}{6b} \left[2 \exp\left(\frac{z}{2b}\right) - \exp\left(\frac{z}{b}\right) \right] dz, \tag{A27}$$

$$= \frac{1}{6b} \left[4b \exp\left(\frac{\rho_t - \epsilon_{abc}}{2b}\right) - b \exp\left(\frac{\rho_t - \epsilon_{abc}}{b}\right) \right], \tag{A28}$$

$$= \frac{1}{6} \left[4 \exp\left(\frac{\rho_t - \epsilon_{abc}}{2b}\right) - \exp\left(\frac{\rho_t - \epsilon_{abc}}{b}\right) \right], \text{ where } \rho_t - \epsilon_{abc} \leq 0. \tag{A29}$$

So:

$$P[\tilde{\tau}_t \neq \tau_t | \tau_t] = \begin{cases} P[\tilde{\tau}_t = 1 | \tau_t = 0], & \text{if } \rho_t \geq \epsilon_{abc} \\ P[\tilde{\tau}_t = 0 | \tau_t = 1], & \text{otherwise} \end{cases} \tag{A30}$$

$$= \begin{cases} \frac{1}{6} \left[4 \exp\left(-\frac{\rho_t - \epsilon_{abc}}{2b}\right) - \exp\left(-\frac{\rho_t - \epsilon_{abc}}{b}\right) \right], & \text{if } \rho_t \geq \epsilon_{abc} \\ \frac{1}{6} \left[4 \exp\left(\frac{\rho_t - \epsilon_{abc}}{2b}\right) - \exp\left(\frac{\rho_t - \epsilon_{abc}}{b}\right) \right], & \text{otherwise.} \end{cases}$$

The two cases can be combined with the use of an absolute value to give the result. \square

Appendix E. Proof of Theorem 2

Proof of Theorem 2. Let $H(x)$ be the Heaviside step function. Recall from our algorithm that each accepted sample (θ, Y) is associated with two independent noise realizations: $m_t \sim \text{Lap}(b)$ (i.e., $\hat{\epsilon}_{abc} = \epsilon_{abc} + m_t$) and $v_t \sim \text{Lap}(2b)$ (added to $\rho(Y^*, Y_t)$). With this notation, we have $\tilde{\tau}_t = H[\epsilon_{abc} - \rho(Y_t, Y^*) + m_t - v_t]$ for $t = 1, \dots, T$. Similarly, $\tau_t := H[\epsilon_{abc} - \rho(Y_t, Y^*)]$. For brevity, we define $\rho_t := \rho(Y_t, Y^*)$. It follows that $\tilde{\tau}_t \sim \text{Bernoulli}(p_t)$ where $p_t := P(m_t - v_t > \rho_t - \epsilon_{abc}) = P(\tilde{\tau} = 1)$.

Proof of the first claim: we start by establishing an upper bound for:

$$\begin{aligned} & \left\| \frac{1}{c} \sum_{t=1}^T f(\theta_t) \tilde{\tau}_t - \frac{1}{c'} \sum_{t=1}^T f(\theta_t) \tau_t \right\|_2 \\ &= \left\| \frac{1}{c} \sum_{t=1}^T f(\theta_t) \tilde{\tau}_t - \frac{1}{c'} \sum_{t=1}^T f(\theta_t) \tilde{\tau}_t + \frac{1}{c'} \sum_{t=1}^T f(\theta_t) \tilde{\tau}_t - \frac{1}{c'} \sum_{t=1}^T f(\theta_t) \tau_t \right\|_2 \\ &\leq \left| \frac{1}{c} - \frac{1}{c'} \right| \left\| \sum_{t=1}^T f(\theta_t) \tilde{\tau}_t \right\| + \frac{1}{c'} \left\| \sum_{t=1}^T f(\theta_t) \tilde{\tau}_t - \sum_{t=1}^T f(\theta_t) \tau_t \right\| \\ &= \frac{1}{c'} |c' - c| \left\| \sum_{t=1}^T f(\theta_t) \tilde{\tau}_t \right\| + \frac{1}{c'} \left\| \sum_{t=1}^T f(\theta_t) (\tilde{\tau}_t - \tau_t) \right\| \\ &\leq \frac{1}{c'} \left| \sum_{t=1}^T \tau_t - \sum_{t=1}^T \tilde{\tau}_t \right| \left\| \sum_{t=1}^T f(\theta_t) \tilde{\tau}_t \right\| + \frac{1}{c'} \sum_{t=1}^T \|f(\theta_t)\|_2 |\tilde{\tau}_t - \tau_t| \\ &\leq \frac{1}{c'} \sum_{t=1}^T |\tilde{\tau}_t - \tau_t| \frac{1}{c} \left\| \sum_{t=1}^T f(\theta_t) \tilde{\tau}_t \right\| + \frac{K_T}{c'} \sum_{t=1}^T |\tilde{\tau}_t - \tau_t|, \end{aligned} \tag{A31}$$

where $K_T := \max_{t=1, \dots, T} \|f(\theta_t)\|_2$. Consider $\frac{1}{c} \left\| \sum_{t=1}^T f(\theta_t) \tilde{\tau}_t \right\|$. We can show that it is bounded by K_T by

$$\frac{1}{c} \left\| \sum_{t=1}^T f(\theta_t) \tilde{\tau}_t \right\| \leq \frac{1}{c} \sum_{t=1}^T \|f(\theta_t)\|_2 \tilde{\tau}_t \leq \frac{K_T}{c} \sum_{t=1}^T \tilde{\tau}_t = K_T.$$

Combining this bound with (A31), we have:

$$\left\| \frac{1}{c} \sum_{t=1}^T f(\theta_t) \tilde{\tau}_t - \frac{1}{c'} \sum_{t=1}^T f(\theta_t) \tau_t \right\|_2 \leq \frac{2K_T}{c'} \sum_{t=1}^T |\tilde{\tau}_t - \tau_t| \tag{A32}$$

We will need to characterize the distribution of $|\tilde{\tau}_t - \tau_t|$. Let $Z_t := m_t - v_t$. By Lemma 2, we have:

$$\begin{aligned} p_t &= P(\tilde{\tau}_t = 1) = P(Z_t > \rho_t - \epsilon_{abc}) = 1 - F_Z(\rho_t - \epsilon_{abc}) \\ &= 1 - H[\rho_t - \epsilon_{abc}] + (2H[\rho_t - \epsilon_{abc}] - 1)G_b(|\rho_t - \epsilon_{abc}|) \\ &= \tau_t + (1 - 2\tau_t)G_b(|\rho_t - \epsilon_{abc}|), \end{aligned}$$

where the decreasing function $G_b(x) \in (0, \frac{1}{2}]$ for any $x \geq 0$ is defined in Lemma 2. We observe that $|\tilde{\tau}_t - \tau_t| \sim \text{Bernoulli}(q_t)$ where $q_t := P(\tilde{\tau}_t \neq \tau_t) = (1 - p_t)\tau_t + p_t(1 - \tau_t)$. We can rewrite q_t as

$$\begin{aligned} q_t &= \tau_t + p_t(1 - 2\tau_t) \\ &= \tau_t + [\tau_t + (1 - 2\tau_t)G_b(|\rho_t - \epsilon_{abc}|)](1 - 2\tau_t) \\ &= G_b(|\rho_t - \epsilon_{abc}|). \end{aligned}$$

To prove the first claim, we take the expectation on both sides of (A32):

$$\begin{aligned}\mathbb{E}_{\tilde{\tau}_1, \dots, \tilde{\tau}_T} \left\| \frac{1}{c} \sum_{t=1}^T f(\boldsymbol{\theta}_t) \tilde{\tau}_t - \frac{1}{c'} \sum_{t=1}^T f(\boldsymbol{\theta}_t) \tau_t \right\|_2 &\leq \frac{2K_T}{c'} \mathbb{E}_{\tilde{\tau}_t} \left[\sum_{t=1}^T |\tilde{\tau}_t - \tau_t| \right] \\ &= \frac{2K_T}{c'} \mu_T,\end{aligned}$$

where $\mu_T = \mathbb{E}_{\tilde{\tau}_t} \left[\sum_{t=1}^T |\tilde{\tau}_t - \tau_t| \right] = \sum_{t=1}^T G_b(|\rho_t - \epsilon_{abc}|)$ and we use the fact that $\mathbb{E}_{\tilde{\tau}_t} |\tilde{\tau}_t - \tau_t| = q_t$. Note that these are T independent, marginal expectations, i.e., they do not depend on the condition that noise is added to the ABC threshold.

Proof of the second claim: observe that $G_b(|\rho_t - \epsilon_{abc}|) \rightarrow 0$ as $b \rightarrow 0$. The claim follows by noting that $b = O(1/N)$.

Proof of the third claim: based on (A32), characterizing the tail bound of $\left\| \frac{1}{c} \sum_{t=1}^T f(\boldsymbol{\theta}_t) \tilde{\tau}_t - \frac{1}{c'} \sum_{t=1}^T f(\boldsymbol{\theta}_t) \tau_t \right\|_2$ amounts to establishing a tail bound on $S_T := \sum_{t=1}^T |\tilde{\tau}_t - \tau_t|$. By Markov's inequality:

$$\begin{aligned}\mathbb{P}(S_T \leq s) &\geq 1 - \mathbb{E}[S_T]/s \\ &= 1 - \frac{1}{s} \sum_{t=1}^T G_b(|\rho_t - \epsilon_{abc}|) \\ &= 1 - \frac{1}{s} \sum_{t=1}^T \frac{1}{6} \left[4 \exp\left(-\frac{|\rho_t - \epsilon_{abc}|}{2b}\right) - \exp\left(-\frac{|\rho_t - \epsilon_{abc}|}{b}\right) \right] \\ &\geq 1 - \frac{2}{3s} \sum_{t=1}^T \exp\left(-\frac{|\rho_t - \epsilon_{abc}|}{2b}\right).\end{aligned}$$

Applying this bound to (A32) gives:

$$\mathbb{P}\left(\frac{2K_T}{c'} S_T \leq \frac{2K_T}{c'} s\right) = \mathbb{P}(S_T \leq s).$$

With a reparametrization $a := \frac{2K_T}{c'} s$ so that $s = \frac{ac'}{2K_T}$, we have:

$$\mathbb{P}\left(\frac{2K_T}{c'} S_T \leq a\right) \geq 1 - \frac{4K_T}{3ac'} \sum_{t=1}^T \exp\left(-\frac{|\rho_t - \epsilon_{abc}|}{2b}\right),$$

Since $\left\| \frac{1}{c} \sum_{t=1}^T f(\boldsymbol{\theta}_t) \tilde{\tau}_t - \frac{1}{c'} \sum_{t=1}^T f(\boldsymbol{\theta}_t) \tau_t \right\|_2 \leq \frac{2K_T}{c'} S_T$, we have:

$$\mathbb{P}\left(\left\| \frac{1}{c} \sum_{t=1}^T f(\boldsymbol{\theta}_t) \tilde{\tau}_t - \frac{1}{c'} \sum_{t=1}^T f(\boldsymbol{\theta}_t) \tau_t \right\|_2 \leq a\right) \geq \mathbb{P}\left(\frac{2K_T}{c'} S_T \leq a\right).$$

which gives the result in the third claim. \square

References

1. Tavaré, S.; Balding, D.J.; Griffiths, R.C.; Donnelly, P. Inferring coalescence times from DNA sequence data. *Genetics* **1997**, *145*, 505–518. [[CrossRef](#)] [[PubMed](#)]
2. Ratmann, O.; Jørgensen, O.; Hinkley, T.; Stumpf, M.; Richardson, S.; Wiuf, C. Using Likelihood-Free Inference to Compare Evolutionary Dynamics of the Protein Networks of *H. pylori* and *P. falciparum*. *PLoS Comput. Biol.* **2007**, *3*, e230. [[CrossRef](#)] [[PubMed](#)]
3. Bazin, E.; Dawson, K.J.; Beaumont, M.A. Likelihood-Free Inference of Population Structure and Local Adaptation in a Bayesian Hierarchical Model. *Genetics* **2010**, *185*, 587–602. [[CrossRef](#)] [[PubMed](#)]
4. Schafer, C.M.; Freeman, P.E. Likelihood-Free Inference in Cosmology: Potential for the Estimation of Luminosity Functions. In *Statistical Challenges in Modern Astronomy V*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 3–19.

5. Pritchard, J.; Seielstad, M.; Perez-Lezaun, A.; Feldman, M. Population growth of human Y chromosomes: A study of Y chromosome microsatellites. *Mol. Biol. Evol.* **1999**, *16*, 1791–1798. [[CrossRef](#)]
6. Fearnhead, P.; Prangle, D. Constructing summary statistics for approximate Bayesian computation: Semi-automatic approximate Bayesian computation. *J. R. Stat. Soc. Ser.* **2012**, *74*, 419–474. [[CrossRef](#)]
7. Joyce, P.; Marjoram, P. Approximately Sufficient Statistics and Bayesian Computation. *Stat. Appl. Genet. Molec. Biol.* **2008**, *7*, 1544–6115. [[CrossRef](#)] [[PubMed](#)]
8. Robert, C.P.; Cornuet, J.; Marin, J.; Pillai, N.S. Lack of confidence in approximate Bayesian computation model choice. *Proc. Natl. Acad. Sci. USA* **2011**, *108*, 15112–15117. [[CrossRef](#)] [[PubMed](#)]
9. Nunes, M.; Balding, D. On Optimal Selection of Summary Statistics for Approximate Bayesian Computation. *Stat. Appl. Genet. Molec. Biol.* **2010**, *9*. [[CrossRef](#)] [[PubMed](#)]
10. Aeschbacher, S.; Beaumont, M.A.; Futschik, A. A Novel Approach for Choosing Summary Statistics in Approximate Bayesian Computation. *Genetics* **2012**, *192*, 1027–1047. [[CrossRef](#)] [[PubMed](#)]
11. Drovandi, C.; Pettitt, A.; Lee, A. Bayesian Indirect Inference Using a Parametric Auxiliary Model. *Statist. Sci.* **2015**, *30*, 72–95. [[CrossRef](#)]
12. Homer, N.; Szelinger, S.; Redman, M.; Duggan, D.; Tembe, W.; Muehling, J.; Pearson, J.V.; Stephan, D.A.; Nelson, S.F.; Craig, D.W. Resolving Individuals Contributing Trace Amounts of DNA to Highly Complex Mixtures Using High-Density SNP Genotyping Microarrays. *PLoS Genet.* **2008**, *4*, e1000167. [[CrossRef](#)] [[PubMed](#)]
13. Johnson, A.; Shmatikov, V. Privacy-preserving Data Exploration in Genome-wide Association Studies. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 1079–1087.
14. Tanaka, M.M.; Francis, A.R.; Luciani, F.; Sisson, S.A. Using approximate Bayesian computation to estimate tuberculosis transmission parameters from genotype data. *Genetics* **2006**, *173*, 1511–1520. [[CrossRef](#)] [[PubMed](#)]
15. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating noise to sensitivity in private data analysis. In Proceedings of the TCC, New York, NY, USA, 4–7 March 2006; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3876, pp. 265–284.
16. Chaudhuri, K.; Monteleoni, C.; Sarwate, A.D. Differentially Private Empirical Risk Minimization. *J. Mach. Learn. Res.* **2011**, *12*, 1069–1109. [[PubMed](#)]
17. Dwork, C.; Roth, A. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* **2014**, *9*, 211–407. [[CrossRef](#)]
18. Park, M.; Jitkrittum, W.; Sejdinovic, D. K2-ABC: Approximate Bayesian Computation with Infinite Dimensional Summary Statistics via Kernel Embeddings. In Proceedings of the AISTATS, Cadiz, Spain, 9–11 May 2016.
19. Nakagome, S.; Fukumizu, K.; Mano, S. Kernel approximate Bayesian computation in population genetic inferences. *Stat. Appl. Genet. Mol. Biol.* **2013**, *12*, 667–678. [[CrossRef](#)] [[PubMed](#)]
20. Gleim, A.; Pigorsch, C. *Approximate Bayesian Computation with Indirect Summary Statistics*; University of Bonn: Bonn, Germany, 2013.
21. Gretton, A.; Borgwardt, K.; Rasch, M.; Schölkopf, B.; Smola, A. A Kernel Two-Sample Test. *J. Mach. Learn. Res.* **2012**, *13*, 723–773.
22. Smola, A.; Gretton, A.; Song, L.; Schölkopf, D. A Hilbert space embedding for distributions. In *Algorithmic Learning Theory, Proceedings of the 18th International Conference, Sendai, Japan, 1–4 October 2007*; Springer: Berlin/Heidelberg, Germany, 2007.
23. Sriperumbudur, B.; Fukumizu, K.; Lanckriet, G. Universality, characteristic kernels and RKHS embedding of measures. *J. Mach. Learn. Res.* **2011**, *12*, 2389–2410.
24. Dwork, C.; Kenthapadi, K.; McSherry, F.; Mironov, I.; Naor, M. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In *Advances in Cryptology—EUROCRYPT 2006, Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, 28 May–1 June 2006*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4004, pp. 486–503.
25. Mironov, I. Rényi Differential Privacy. In Proceedings of the 30th IEEE Computer Security Foundations Symposium (CSF), Santa Barbara, CA, USA, 21–25 August 2017; pp. 263–275.
26. Lyu, M.; Su, D.; Li, N. Understanding the Sparse Vector Technique for Differential Privacy. *Proc. VLDB Endow.* **2017**, *10*, 637–648. [[CrossRef](#)]
27. Gong, R. Exact Inference with Approximate Computation for Differentially Private Data via Perturbations. *arXiv* **2019**, arXiv:stat.CO/1909.12237.
28. Lintusaari, J.; Blomstedt, P.; Rose, B.; Sivula, T.; Gutmann, M.; Kaski, S.; Corander, J. Resolving outbreak dynamics using approximate Bayesian computation for stochastic birth?death models [version 2; peer review: 2 approved]. *Wellcome Open Res.* **2019**, *4*. [[CrossRef](#)]
29. Zhu, Y.; Wang, Y.X. Improving Sparse Vector Technique with Renyi Differential Privacy. In Proceedings of the 2020 Conference on Neural Information Processing Systems, Virtual, 6–12 December 2020; Volume 33.