Inference from Privatized Data

Sampling-based and Variational-based Bayesian Computation

Yifei Xiong

Department of Statistics, Purdue University

April 2026

https://yifei-xiong.github.io/

Joint work with Nianqiao Phyllis Ju (Dartmouth College)

Why inference from privatized data?

- In many modern data-sharing settings, the raw confidential data cannot be released directly.
- Instead, a data curator releases only a *privatized* or *privacy-protected* summary.
- This already appears in settings such as tech companies, the U.S. Census Bureau, and social science data releases.

raw data $\longrightarrow$ summary $\longrightarrow$ add privacy noise $\longrightarrow$ released value

Central question

How can we perform inference when we only observe a noisy, privatized release rather than the raw data?

Why inference from privatized data?

- In many modern data-sharing settings, the raw confidential data cannot be released directly.
- Instead, a data curator releases only a *privatized* or *privacy-protected* summary.
- This already appears in settings such as tech companies, the U.S. Census Bureau, and social science data releases.

raw data $\longrightarrow$ summary $\longrightarrow$ add privacy noise $\longrightarrow$ released value

Central question

How can we perform inference when we only observe a noisy, privatized release rather than the raw data?

Differential privacy and the Laplace mechanism

Two datasets $\mathbf{x}, \mathbf{x}' \in \mathbb{X}^n$ are neighboring if they differ in one individual record.

ϵ-differential privacy (Dwork, 2006)

A mechanism η is ϵ-DP if for any neighboring $\mathbf{x}, \mathbf{x}'$ and any measurable set B,

$$\mathbb{P}\big(\eta(\mathbf{x}) \in B\big) \leq e^{\epsilon}\, \mathbb{P}\big(\eta(\mathbf{x}') \in B\big).$$

For a query $s : \mathbb{X}^n \to \mathbb{R}^d$, define its ℓ_1-sensitivity by

$$\Delta_1(s) := \sup_{d(\mathbf{x},\mathbf{x}')=1} \|s(\mathbf{x}) - s(\mathbf{x}')\|_1.$$

Laplace mechanism

Release

$$S_{\mathrm{dp}} = s(\mathbf{X}) + Z, \qquad Z_j \overset{\mathrm{iid}}{\sim} \mathrm{Laplace}\left(0, \frac{\Delta_1(s)}{\epsilon}\right).$$

Then S_{dp} is ϵ-differentially private.

Differential privacy and the Laplace mechanism

Two datasets $\mathbf{x}, \mathbf{x}' \in \mathbb{X}^n$ are neighboring if they differ in one individual record.

ϵ-differential privacy (Dwork, 2006)

A mechanism η is ϵ-DP if for any neighboring $\mathbf{x}, \mathbf{x}'$ and any measurable set B,

$$\mathbb{P}\big(\eta(\mathbf{x}) \in B\big) \leq e^{\epsilon}\, \mathbb{P}\big(\eta(\mathbf{x}') \in B\big).$$

For a query $s : \mathbb{X}^n \to \mathbb{R}^d$, define its ℓ_1-sensitivity by

$$\Delta_1(s) := \sup_{d(\mathbf{x},\mathbf{x}')=1} \|s(\mathbf{x}) - s(\mathbf{x}')\|_1.$$

Laplace mechanism

Release

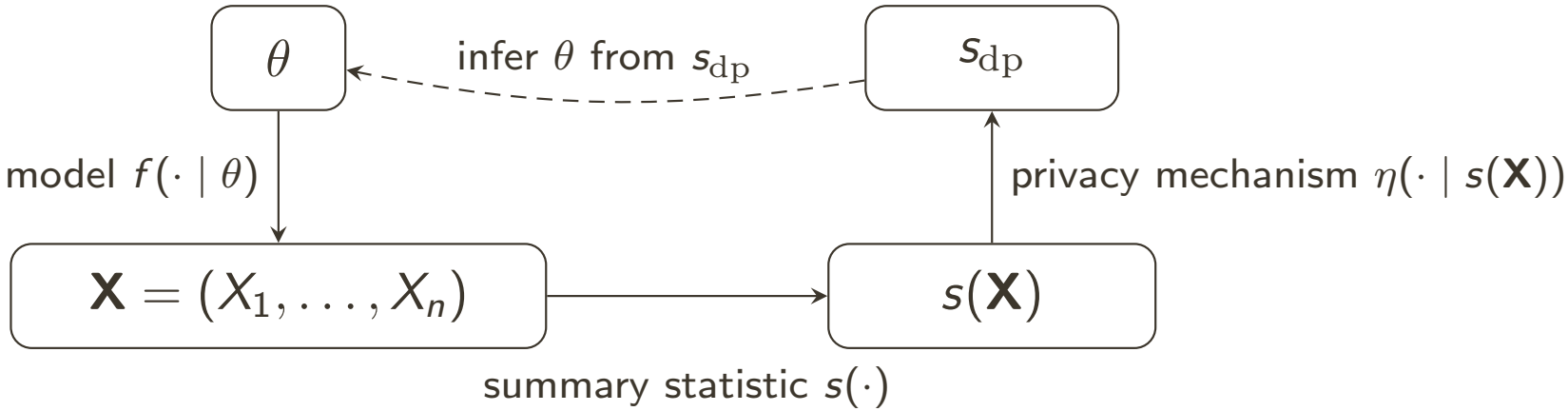
$$S_{\mathrm{dp}} = s(\mathbf{X}) + Z, \qquad Z_j \overset{\mathrm{iid}}{\sim} \mathrm{Laplace}\left(0, \frac{\Delta_1(s)}{\epsilon}\right).$$

Then S_{dp} is ϵ-differentially private.

A generative view of privatized inference

Notations

- θ: model parameters
- $\mathbf{X} = (X_1, \ldots, X_n)$: confidential data
- $s(\mathbf{X})$: summary statistics of $\mathbf{X}$
- s_{dp}: privatized query of $s(\mathbf{X})$.

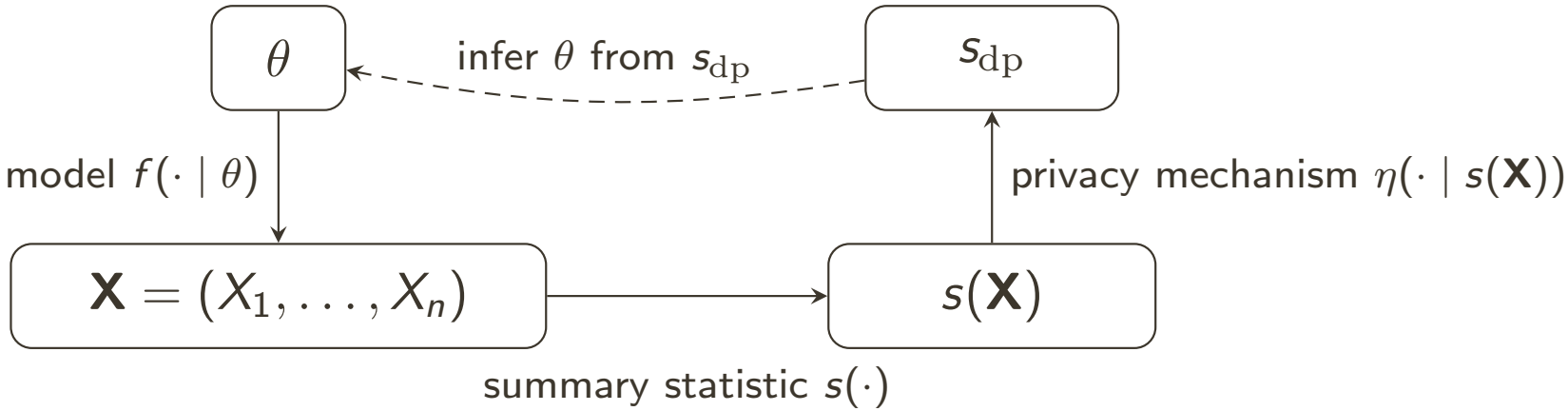


- Posterior: $\pi(\theta \mid s_{\mathrm{dp}}) \propto \pi(\theta) f(s_{\mathrm{dp}} \mid \theta)$
- Likelihood: $f(s_{\mathrm{dp}} \mid \theta) = \int_{\mathbb{X}^n} f(\mathbf{x} \mid \theta)\, \eta(s_{\mathrm{dp}} \mid s(\mathbf{x}))\, \mathrm{d}\mathbf{x}$

A generative view of privatized inference

Notations

- θ: model parameters
- $\mathbf{X} = (X_1, \ldots, X_n)$: confidential data
- $s(\mathbf{X})$: summary statistics of $\mathbf{X}$
- s_{dp}: privatized query of $s(\mathbf{X})$.



- Posterior: $\pi(\theta \mid s_{\mathrm{dp}}) \propto \pi(\theta) f(s_{\mathrm{dp}} \mid \theta)$
- Likelihood: $f(s_{\mathrm{dp}} \mid \theta) = \int_{\mathbb{X}^n} f(\mathbf{x} \mid \theta)\, \eta(s_{\mathrm{dp}} \mid s(\mathbf{x}))\, \mathrm{d}\mathbf{x}$

Two computational viewpoints for inference from s_{dp}

Our target is the private posterior $\pi(\theta \mid s_{\mathrm{dp}}) \propto \pi(\theta)\, f(s_{\mathrm{dp}} \mid \theta)$.

Sampling-based methods

Construct a Markov chain whose stationary distribution is

$$\pi(\theta \mid s_{\mathrm{dp}}) \quad \text{or} \quad \pi(\theta, \mathbf{x} \mid s_{\mathrm{dp}}),$$

and use posterior samples $\theta^{(1)}, \ldots, \theta^{(M)}$ to approximate expectations under the posterior.

- exact or asymptotically exact
- often computationally expensive

Variational-based methods

Choose a tractable family

$$q_\phi(\theta \mid s_{\mathrm{dp}})$$

and fit ϕ so that

$$q_\phi(\theta \mid s_{\mathrm{dp}}) \approx \pi(\theta \mid s_{\mathrm{dp}})$$

as a distribution approximation.

- typically non-exact
- usually faster and more scalable

Two computational viewpoints for inference from s_{dp}

Our target is the private posterior $\pi(\theta \mid s_{\mathrm{dp}}) \propto \pi(\theta)\, f(s_{\mathrm{dp}} \mid \theta)$.

Sampling-based methods

Construct a Markov chain whose stationary distribution is

$$\pi(\theta \mid s_{\mathrm{dp}}) \quad \text{or} \quad \pi(\theta, \mathbf{x} \mid s_{\mathrm{dp}}),$$

and use posterior samples $\theta^{(1)}, \ldots, \theta^{(M)}$ to approximate expectations under the posterior.

- exact or asymptotically exact
- often computationally expensive

Variational-based methods

Choose a tractable family

$$q_\phi(\theta \mid s_{\mathrm{dp}})$$

and fit ϕ so that

$$q_\phi(\theta \mid s_{\mathrm{dp}}) \approx \pi(\theta \mid s_{\mathrm{dp}})$$

as a distribution approximation.

- typically non-exact
- usually faster and more scalable

Sampling-based idea: data augmentation

A natural strategy is to augment the unobserved confidential data

$$\mathbf{X} = (X_1, \ldots, X_n)$$

and work with the joint posterior

$$\pi(\theta, \mathbf{x} \mid s_{\mathrm{dp}}) \propto \pi(\theta)\, f(\mathbf{x} \mid \theta)\, \eta(s_{\mathrm{dp}} \mid s(\mathbf{x})).$$

If the data model factorizes as

$$f(\mathbf{x} \mid \theta) = \prod_{i=1}^{n} f(x_i \mid \theta),$$

then

$$\pi(\theta, \mathbf{x} \mid s_{\mathrm{dp}}) \propto \pi(\theta) \left[\prod_{i=1}^{n} f(x_i \mid \theta) \right] \eta(s_{\mathrm{dp}} \mid s(\mathbf{x})).$$

Idea

If we can sample from $\pi(\theta, \mathbf{x} \mid s_{\mathrm{dp}})$, then the θ-marginal targets the posterior of interest $\pi(\theta \mid s_{\mathrm{dp}})$.

Gibbs update

The augmented posterior suggests a two-block Gibbs strategy.

Inference step

$$\pi(\theta \mid \mathbf{x}, s_{\mathrm{dp}}) = \pi(\theta \mid \mathbf{x}) \propto \pi(\theta) \prod_{i=1}^{n} f(x_i \mid \theta).$$

Imputation step

$$\pi(\mathbf{x} \mid \theta, s_{\mathrm{dp}}) \propto \left[\prod_{i=1}^{n} f(x_i \mid \theta) \right] \eta(s_{\mathrm{dp}} \mid s(\mathbf{x})).$$

- The privacy mechanism only enters through the imputation step.
- This is the computational bottleneck: $\mathbf{x}$ is high-dimensional, and the privacy term couples all coordinates through $s(\mathbf{x})$.

Gibbs update

The augmented posterior suggests a two-block Gibbs strategy.

Inference step

$$\pi(\theta \mid \mathbf{x}, s_{\mathrm{dp}}) = \pi(\theta \mid \mathbf{x}) \propto \pi(\theta) \prod_{i=1}^{n} f(x_i \mid \theta).$$

Imputation step

$$\pi(\mathbf{x} \mid \theta, s_{\mathrm{dp}}) \propto \left[\prod_{i=1}^{n} f(x_i \mid \theta) \right] \eta(s_{\mathrm{dp}} \mid s(\mathbf{x})).$$

- The privacy mechanism only enters through the imputation step.
- This is the computational bottleneck: $\mathbf{x}$ is high-dimensional, and the privacy term couples all coordinates through $s(\mathbf{x})$.

Independent Metropolis-within-Gibbs sampler

- We want to sample from

$$\pi(\mathbf{x} \mid \theta, s_{\mathrm{dp}}) \propto \prod_{i=1}^{n} f(x_i \mid \theta) \cdot \eta(s_{\mathrm{dp}} \mid s(\mathbf{x})). \tag{1}$$

- Ju et al. (2022) target the DP imputation distribution using a **independent Metropolis-within-Gibbs** (IMwG) sampler.

Algorithm: IMwG Update

Given current state $\mathbf{x} = (x_1, \ldots, x_n)$:

1. Pick $I \sim \mathrm{Unif}\{1, \ldots, n\}$.
2. Propose $y \sim q(\cdot)$, independent of x_I. (We can choose $q(\cdot) = f(\cdot \mid \theta)$)
3. Form candidate $\mathbf{x}' = [\mathbf{x}_{-I}, y]$.
4. Accept with probability

$$\alpha_I^{\mathrm{IMwG}}(y, \mathbf{x}) = 1 \wedge \frac{\pi(\mathbf{x}' \mid \theta, s_{\mathrm{dp}})\, q(x_I)}{\pi(\mathbf{x} \mid \theta, s_{\mathrm{dp}})\, q(y)}. \quad \left(= 1 \wedge \frac{\eta(s_{\mathrm{dp}} \mid s(\mathbf{x}'))}{\eta(s_{\mathrm{dp}} \mid s(\mathbf{x}))} \geq e^{-\epsilon} \right)$$

Independent Metropolis-within-Gibbs sampler

- We want to sample from

$$\pi(\mathbf{x} \mid \theta, s_{\mathrm{dp}}) \propto \prod_{i=1}^{n} f(x_i \mid \theta) \cdot \eta(s_{\mathrm{dp}} \mid s(\mathbf{x})). \tag{1}$$

- Ju et al. (2022) target the DP imputation distribution using a **independent Metropolis-within-Gibbs** (IMwG) sampler.

Algorithm: IMwG Update

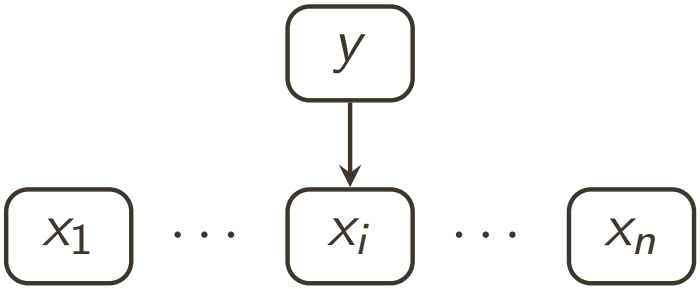
Given current state $\mathbf{x} = (x_1, \ldots, x_n)$:

1. Pick $I \sim \mathrm{Unif}\{1, \ldots, n\}$.
2. Propose $y \sim q(\cdot)$, independent of x_I. (We can choose $q(\cdot) = f(\cdot \mid \theta)$)
3. Form candidate $\mathbf{x}' = [\mathbf{x}_{-I}, y]$.
4. Accept with probability

$$\alpha_I^{\mathrm{IMwG}}(y, \mathbf{x}) = 1 \wedge \frac{\pi(\mathbf{x}' \mid \theta, s_{\mathrm{dp}})\, q(x_I)}{\pi(\mathbf{x} \mid \theta, s_{\mathrm{dp}})\, q(y)}. \quad \left(= 1 \wedge \frac{\eta(s_{\mathrm{dp}} \mid s(\mathbf{x}'))}{\eta(s_{\mathrm{dp}} \mid s(\mathbf{x}))} \geq e^{-\epsilon} \right)$$

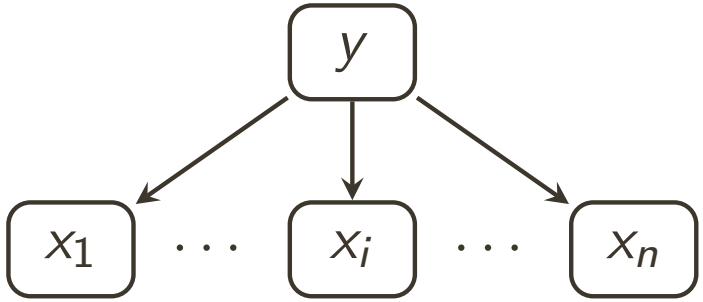
Idea: reusing proposals

Independent Metropolis (IMwG)



one proposal $\rightarrow$ one coordinate

SOMA

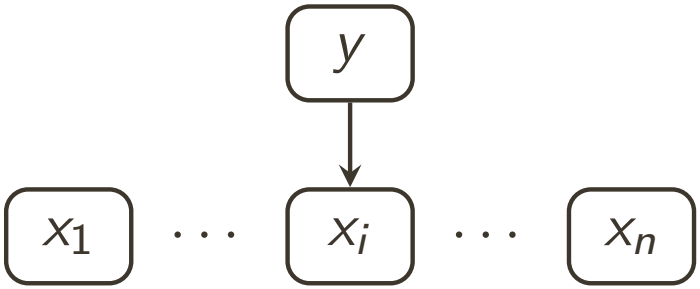


one proposal $\rightarrow$ many coordinates

- In IMwG, each proposal y only gets *one chance* to replace a randomly chosen coordinate x_i.
- Our idea: keep **one** proposal y, but let it compete across **all** coordinates and move it where it helps the most.
- We let y "compete" across the slots and update the most promising coordinate. This is SOMA: **Single-Offer-Multiple-Attempts**.

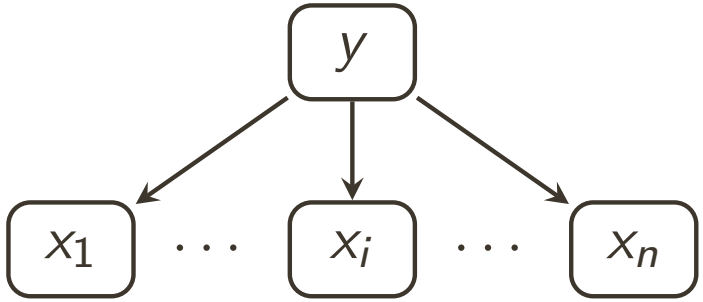
Idea: reusing proposals

Independent Metropolis (IMwG)



one proposal $\rightarrow$ one coordinate

SOMA



one proposal $\rightarrow$ many coordinates

- In IMwG, each proposal y only gets *one chance* to replace a randomly chosen coordinate x_i.
- Our idea: keep **one** proposal y, but let it compete across **all** coordinates and move it where it helps the most.
- We let y "compete" across the slots and update the most promising coordinate. This is SOMA: **Single-Offer-Multiple-Attempts**.

SOMA kernel (one iteration)

One SOMA iteration targeting $\pi(\mathbf{x})$

1. **Propose** a single value $y \sim q(\cdot)$.
2. **Compute weights**

$$w_0(\mathbf{x}) = \frac{\pi(\mathbf{x})}{\prod_{j=1}^{n} q(x_j)}, \quad w_i(y, \mathbf{x}) = \frac{\pi([\mathbf{x}_{-i}, y])}{q(y) \prod_{j \neq i} q(x_j)},$$

   and set $W(y, \mathbf{x}) = \sum_{i=1}^{n} w_i(y, \mathbf{x})$.

3. **Choose a coordinate**

$$\mathbb{P}(I = i \mid y, \mathbf{x}) = \frac{w_i(y, \mathbf{x})}{W(y, \mathbf{x})}, \quad i = 1, \ldots, n.$$

4. **Accept / reject** replacement of x_I by y with probability

$$\alpha_I^{\mathrm{SOMA}}(y, \mathbf{x}) = \min\left\{1, \frac{W(y, \mathbf{x})}{W(y, \mathbf{x}) + w_0(\mathbf{x}) - w_I(y, \mathbf{x})}\right\}.$$

Why SOMA is valid and useful

Reversibility

If the target $\pi(\mathbf{x})$ is permutation invariant in $(x_1, \ldots, x_n)$, then SOMA is reversible with respect to $\pi(\mathbf{x})$.

For the DP imputation target

$$\pi(\mathbf{x}) \propto \left[\prod_{i=1}^{n} f(x_i \mid \theta)\right] \eta(s_{\mathrm{dp}} \mid s(\mathbf{x})),$$

this holds whenever $s(\mathbf{x})$ is permutation invariant.

Advantage over IMwG

For any state $\mathbf{x}$, coordinate i, and proposal y,

$$\alpha_i^{\mathrm{SOMA}}(y, \mathbf{x}) \geq \alpha_i^{\mathrm{IMwG}}(y, \mathbf{x}).$$

Under ϵ-DP, the overall acceptance probability satisfied

$$A^{\mathrm{SOMA}}(\mathbf{x}) \geq \frac{n}{n + e^{\epsilon} - 1}, \qquad A^{\mathrm{IMwG}}(\mathbf{x}) \geq e^{-\epsilon}.$$

Why SOMA is valid and useful

Reversibility

If the target $\pi(\mathbf{x})$ is permutation invariant in $(x_1, \ldots, x_n)$, then SOMA is reversible with respect to $\pi(\mathbf{x})$.

For the DP imputation target

$$\pi(\mathbf{x}) \propto \left[\prod_{i=1}^{n} f(x_i \mid \theta)\right] \eta(s_{\mathrm{dp}} \mid s(\mathbf{x})),$$

this holds whenever $s(\mathbf{x})$ is permutation invariant.

Advantage over IMwG

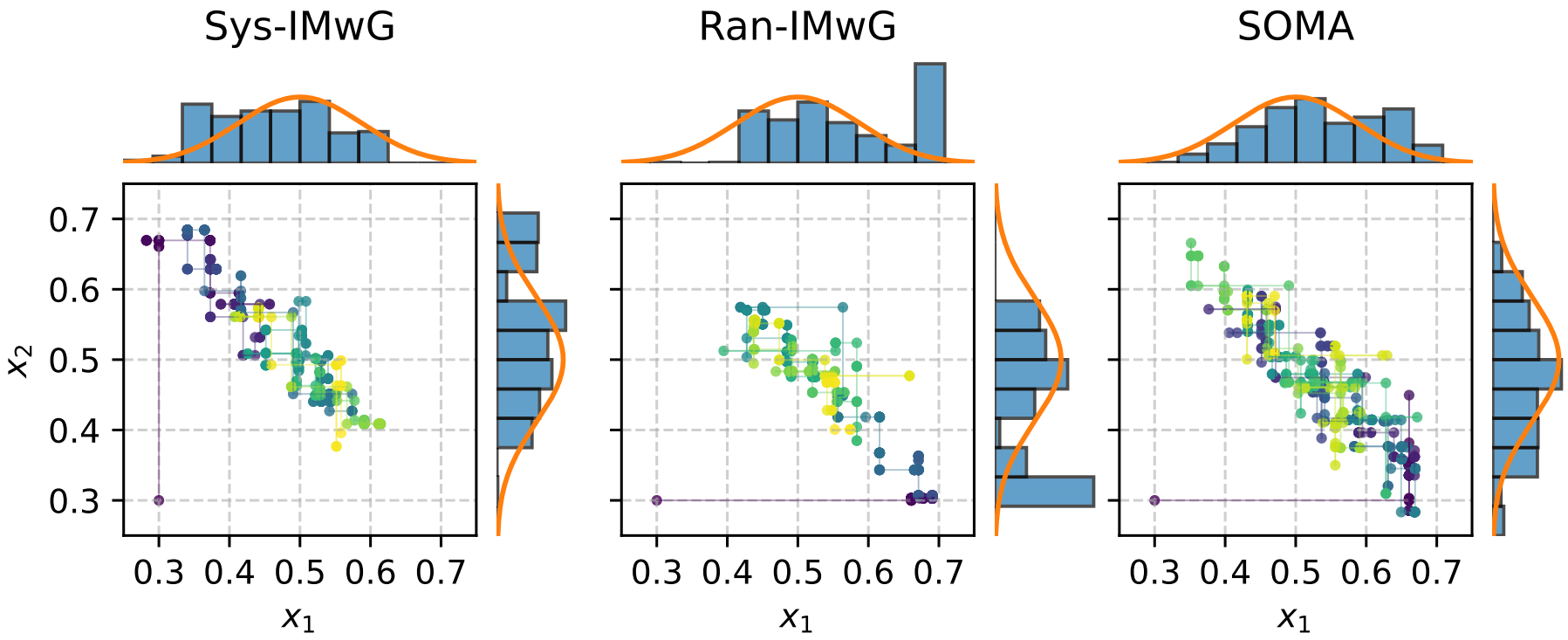
For any state $\mathbf{x}$, coordinate i, and proposal y,

$$\alpha_i^{\mathrm{SOMA}}(y, \mathbf{x}) \geq \alpha_i^{\mathrm{IMwG}}(y, \mathbf{x}).$$

Under ϵ-DP, the overall acceptance probability satisfied

$$A^{\mathrm{SOMA}}(\mathbf{x}) \geq \frac{n}{n + e^{\epsilon} - 1}, \qquad A^{\mathrm{IMwG}}(\mathbf{x}) \geq e^{-\epsilon}.$$

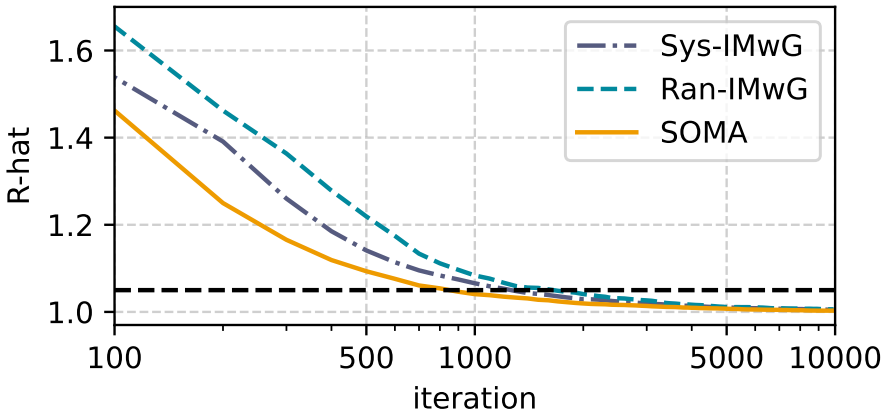
Synthetic Example: Trace Plots ($n = 2$)



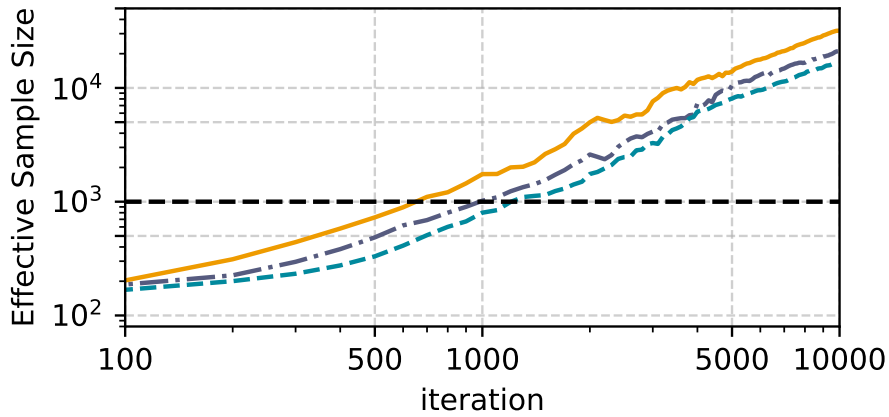
- All chains start from the same initial state $(0.3, 0.3)$ over 500 iterations.
- SOMA (right) traverses the high-density region of π much more quickly and explores the state space more efficiently.
- Ran-IMwG tends to get stuck in local regions, leading to under-exploration of the posterior support.

Convergence Diagnostics at $n = 100$

- Privatised Bayesian linear regression (Bernstein and Sheldon, 2019) with $n = 100$.
- The state space has dimension $n \cdot (p + 2) + (p + 1) = 403$; we use standard MCMC diagnostics: Gelman–Rubin $\hat{R}$ and effective sample size (ESS).



$\hat{R}$ across iterations (dashed line: 1.05)



Bulk ESS across iterations

- $\hat{R}$: SOMA reaches $\hat{R} < 1.05$ after about 800 iterations, while both Sys-IMwG and Ran-IMwG need around 1,500 iterations.
- ESS: SOMA yields substantially larger effective sample size than the two IMwG baselines at the same iteration budget.

Coupling time and Acceptance rate

Table: Coupling time, convergence rate, acceptance rate for $n = 10$ and $\epsilon = 3$ and $\epsilon = 30$. All values are presented as mean with quartiles.

Method	Coupling Time	Convergence Rate	Acceptance Rate (%)
		$n = 10,\ \epsilon = 3$	
Ran-IMwG	156.04 (46, 216)	0.9937 (0.9933, 0.9941)	94.80 (94.52, 95.11)
Sys-IMwG	**99.60** (**28**, **138**)	**0.9904** (**0.9895**, **0.9911**)	94.81 (94.55, 95.09)
SOMA	152.98 (42, 211)	0.9937 (0.9932, 0.9942)	**99.49** (**99.44**, **99.55**)
		$n = 10,\ \epsilon = 30$	
Ran-IMwG	900.79 (261, 1262)	0.9989 (0.9988, 0.9990)	50.95 (50.64, 51.25)
Sys-IMwG	549.07 (170, 764)	0.9982 (0.9980, 0.9983)	50.94 (50.66, 51.23)
SOMA	**120.64** (**46**, **163**)	**0.9909** (**0.9902**, **0.9915**)	**91.91** (**91.80**, **92.01**)

Sampling-based approach: takeaway

What we gained

- Data augmentation turns inference from s_{dp} into sampling from

$$\pi(\theta, \mathbf{x} \mid s_{\mathrm{dp}}).$$

- SOMA improves the imputation step by reusing one proposal across coordinates.
- Under DP, acceptance can be controlled explicitly.

Limitation

Sampling-based methods are exact or asymptotically exact, but they still require iterative MCMC over a high-dimensional latent state.

This motivates a variational approximation to the private posterior.

Why a variational approximation?

Recall the private posterior

$$\pi(\theta \mid s_{\mathrm{dp}}) \propto \pi(\theta)\, f(s_{\mathrm{dp}} \mid \theta), \qquad f(s_{\mathrm{dp}} \mid \theta) = \int_{\mathbb{X}^n} f(\mathbf{x} \mid \theta)\, \eta(s_{\mathrm{dp}} \mid s(\mathbf{x}))\, \mathrm{d}\mathbf{x}.$$

Difficulty

Even if we can simulate from

$$\theta \sim \pi(\theta), \qquad \mathbf{X} \sim f(\cdot \mid \theta), \qquad s_{\mathrm{dp}} \sim \eta(\cdot \mid s(\mathbf{X})),$$

the private likelihood $f(s_{\mathrm{dp}} \mid \theta)$ is usually intractable.

- Sampling-based methods target the posterior $\pi(\theta \mid s_{\mathrm{dp}})$ by MCMC.
- Here we instead learn an approximation q_ϕ from simulated privatized data.

Two approximation targets: posterior or likelihood

There are two natural objects to approximate.

Privatized posterior estimation (PPE)

Approximate

$$\pi(\theta \mid s_{\mathrm{dp}}) \quad \text{by} \quad q_\phi(\theta \mid s_{\mathrm{dp}}).$$

Privatized likelihood estimation (PLE)

Approximate

$$f(s_{\mathrm{dp}} \mid \theta) \quad \text{by} \quad q_\phi(s_{\mathrm{dp}} \mid \theta).$$

- In practice, q_ϕ can be any flexible conditional density model, e.g., normalizing flows (Durkan et al., 2019) or diffusion models.
- We do not need its exact architecture here; what matters is the loss function it minimizes.

PLE: start from a KL objective

We want to approximate the private likelihood

$$f(s_{\mathrm{dp}} \mid \theta) \quad \text{by} \quad q_\phi(s_{\mathrm{dp}} \mid \theta).$$

A natural population loss is the prior-averaged KL divergence:

$$\ell_{\mathrm{PLE}}(\phi) = \mathbb{E}_{\pi(\theta)}\left[\mathrm{KL}\big(f(s_{\mathrm{dp}} \mid \theta) \,\|\, q_\phi(s_{\mathrm{dp}} \mid \theta)\big)\right].$$

Expanding the KL divergence,

$$\begin{aligned}
\ell_{\mathrm{PLE}}(\phi) &= \int \pi(\theta) \int f(s_{\mathrm{dp}} \mid \theta) \log \frac{f(s_{\mathrm{dp}} \mid \theta)}{q_\phi(s_{\mathrm{dp}} \mid \theta)} \, \mathrm{d}s_{\mathrm{dp}} \, \mathrm{d}\theta \\
&= \int \pi(\theta) \int f(s_{\mathrm{dp}} \mid \theta) \log f(s_{\mathrm{dp}} \mid \theta) \, \mathrm{d}s_{\mathrm{dp}} \, \mathrm{d}\theta \\
&\quad - \int \pi(\theta) \int f(s_{\mathrm{dp}} \mid \theta) \log q_\phi(s_{\mathrm{dp}} \mid \theta) \, \mathrm{d}s_{\mathrm{dp}} \, \mathrm{d}\theta.
\end{aligned}$$

The first term does not depend on ϕ.

PLE: start from a KL objective

We want to approximate the private likelihood

$$f(s_{\mathrm{dp}} \mid \theta) \quad \text{by} \quad q_\phi(s_{\mathrm{dp}} \mid \theta).$$

A natural population loss is the prior-averaged KL divergence:

$$\ell_{\mathrm{PLE}}(\phi) = \mathbb{E}_{\pi(\theta)}\left[\mathrm{KL}\big(f(s_{\mathrm{dp}} \mid \theta) \,\|\, q_\phi(s_{\mathrm{dp}} \mid \theta)\big)\right].$$

Expanding the KL divergence,

$$\begin{aligned}
\ell_{\mathrm{PLE}}(\phi) &= \int \pi(\theta) \int f(s_{\mathrm{dp}} \mid \theta) \log \frac{f(s_{\mathrm{dp}} \mid \theta)}{q_\phi(s_{\mathrm{dp}} \mid \theta)} \, \mathrm{d}s_{\mathrm{dp}} \, \mathrm{d}\theta \\
&= \int \pi(\theta) \int f(s_{\mathrm{dp}} \mid \theta) \log f(s_{\mathrm{dp}} \mid \theta) \, \mathrm{d}s_{\mathrm{dp}} \, \mathrm{d}\theta \\
&\quad - \int \pi(\theta) \int f(s_{\mathrm{dp}} \mid \theta) \log q_\phi(s_{\mathrm{dp}} \mid \theta) \, \mathrm{d}s_{\mathrm{dp}} \, \mathrm{d}\theta.
\end{aligned}$$

The first term does not depend on ϕ.

PLE: from KL to a trainable loss

The first term does not depend on ϕ. Hence

$$\ell_{\mathrm{PLE}}(\phi) \equiv -\int \pi(\theta) \int f(s_{\mathrm{dp}} \mid \theta) \log q_\phi(s_{\mathrm{dp}} \mid \theta)\, \mathrm{d}s_{\mathrm{dp}}\, \mathrm{d}\theta.$$

Now use

$$f(s_{\mathrm{dp}} \mid \theta) = \int f(\mathbf{x} \mid \theta)\, \eta(s_{\mathrm{dp}} \mid s(\mathbf{x}))\, \mathrm{d}\mathbf{x}.$$

Substituting this into the loss,

$$\begin{aligned}\ell_{\mathrm{PLE}}(\phi) &\equiv -\int \pi(\theta) \int \left[\int f(\mathbf{x} \mid \theta)\, \eta(s_{\mathrm{dp}} \mid s(\mathbf{x}))\, \mathrm{d}\mathbf{x}\right] \log q_\phi(s_{\mathrm{dp}} \mid \theta)\, \mathrm{d}s_{\mathrm{dp}}\, \mathrm{d}\theta \\ &= \mathbb{E}_{p(\theta,\mathbf{x})}\left[-\int \eta(s_{\mathrm{dp}} \mid s(\mathbf{x})) \log q_\phi(s_{\mathrm{dp}} \mid \theta)\, \mathrm{d}s_{\mathrm{dp}}\right].\end{aligned}$$

So the trainable loss is

$$\ell_{\mathrm{PLE}}(\phi) = \mathbb{E}_{p(\theta,\mathbf{x})}\left[-\int \eta(s_{\mathrm{dp}} \mid s(\mathbf{x})) \log q_\phi(s_{\mathrm{dp}} \mid \theta)\, \mathrm{d}s_{\mathrm{dp}}\right].$$

PPE: start from a KL objective

We now approximate the private posterior

$$\pi(\theta \mid s_{\mathrm{dp}}) \quad \text{by} \quad q_\phi(\theta \mid s_{\mathrm{dp}}).$$

A natural population loss is the average conditional KL divergence:

$$\ell_{\mathrm{PPE}}(\phi) = \mathbb{E}_{p(s_{\mathrm{dp}})}\left[\mathrm{KL}\big(\pi(\theta \mid s_{\mathrm{dp}}) \,\|\, q_\phi(\theta \mid s_{\mathrm{dp}})\big)\right].$$

Expanding the KL divergence,

$$\begin{aligned}
\ell_{\mathrm{PPE}}(\phi) &= \int p(s_{\mathrm{dp}}) \int \pi(\theta \mid s_{\mathrm{dp}}) \log \frac{\pi(\theta \mid s_{\mathrm{dp}})}{q_\phi(\theta \mid s_{\mathrm{dp}})} \,\mathrm{d}\theta \,\mathrm{d}s_{\mathrm{dp}} \\
&= \int p(s_{\mathrm{dp}}) \int \pi(\theta \mid s_{\mathrm{dp}}) \log \pi(\theta \mid s_{\mathrm{dp}}) \,\mathrm{d}\theta \,\mathrm{d}s_{\mathrm{dp}} \\
&\quad - \int p(s_{\mathrm{dp}}) \int \pi(\theta \mid s_{\mathrm{dp}}) \log q_\phi(\theta \mid s_{\mathrm{dp}}) \,\mathrm{d}\theta \,\mathrm{d}s_{\mathrm{dp}}.
\end{aligned}$$

Again, the first term does not depend on ϕ.

PPE: from KL to a trainable loss

Again, the first term does not depend on ϕ. Hence

$$\ell_{\mathrm{PPE}}(\phi) = -\int p(s_{\mathrm{dp}}) \int \pi(\theta \mid s_{\mathrm{dp}}) \log q_\phi(\theta \mid s_{\mathrm{dp}}) \, \mathrm{d}\theta \, \mathrm{d}s_{\mathrm{dp}}.$$

Using the joint density

$$p(\theta, s_{\mathrm{dp}}) = p(s_{\mathrm{dp}}) \pi(\theta \mid s_{\mathrm{dp}}),$$

we rewrite this as

$$\ell_{\mathrm{PPE}}(\phi) = -\int p(\theta, s_{\mathrm{dp}}) \log q_\phi(\theta \mid s_{\mathrm{dp}}) \, \mathrm{d}\theta \, \mathrm{d}s_{\mathrm{dp}}.$$

Since

$$p(\theta, s_{\mathrm{dp}}) = \int \pi(\theta) \, f(\mathbf{x} \mid \theta) \, \eta(s_{\mathrm{dp}} \mid s(\mathbf{x})) \, \mathrm{d}\mathbf{x},$$

we obtain

$$\ell_{\mathrm{PPE}}(\phi) = \mathbb{E}_{p(\theta, \mathbf{x})} \left[-\int \eta(s_{\mathrm{dp}} \mid s(\mathbf{x})) \log q_\phi(\theta \mid s_{\mathrm{dp}}) \, \mathrm{d}s_{\mathrm{dp}} \right].$$

So PPE learns the posterior directly, while PLE learns the private likelihood.

PPE: from KL to a trainable loss

Again, the first term does not depend on ϕ. Hence

$$\ell_{\mathrm{PPE}}(\phi) = -\int p(s_{\mathrm{dp}}) \int \pi(\theta \mid s_{\mathrm{dp}}) \log q_\phi(\theta \mid s_{\mathrm{dp}}) \, \mathrm{d}\theta \, \mathrm{d}s_{\mathrm{dp}}.$$

Using the joint density

$$p(\theta, s_{\mathrm{dp}}) = p(s_{\mathrm{dp}}) \pi(\theta \mid s_{\mathrm{dp}}),$$

we rewrite this as

$$\ell_{\mathrm{PPE}}(\phi) = -\int p(\theta, s_{\mathrm{dp}}) \log q_\phi(\theta \mid s_{\mathrm{dp}}) \, \mathrm{d}\theta \, \mathrm{d}s_{\mathrm{dp}}.$$

Since

$$p(\theta, s_{\mathrm{dp}}) = \int \pi(\theta) \, f(\mathbf{x} \mid \theta) \, \eta(s_{\mathrm{dp}} \mid s(\mathbf{x})) \, \mathrm{d}\mathbf{x},$$

we obtain

$$\ell_{\mathrm{PPE}}(\phi) = \mathbb{E}_{p(\theta, \mathbf{x})} \left[-\int \eta(s_{\mathrm{dp}} \mid s(\mathbf{x})) \log q_\phi(\theta \mid s_{\mathrm{dp}}) \, \mathrm{d}s_{\mathrm{dp}} \right].$$

So PPE learns the posterior directly, while PLE learns the private likelihood.

A unified form of the PPE / PLE losses

Both losses have the same nested-integral structure:

$$\ell(\phi) = -\mathbb{E}_{p(\theta,\mathbf{x})}\left[\int \eta(s_{\mathrm{dp}} \mid s(\mathbf{x}))\, g_\phi(s_{\mathrm{dp}}, \theta)\, \mathrm{d}s_{\mathrm{dp}}\right].$$

Here

$$g_\phi(s_{\mathrm{dp}}, \theta) = \begin{cases} \log q_\phi(\theta \mid s_{\mathrm{dp}}), & \text{for PPE,} \\ \log q_\phi(s_{\mathrm{dp}} \mid \theta), & \text{for PLE.} \end{cases}$$

Key point

The optimization over ϕ is driven by repeated evaluation of the inner integral

$$I_\phi(\theta, \mathbf{x}) = \int \eta(s_{\mathrm{dp}} \mid s(\mathbf{x}))\, g_\phi(s_{\mathrm{dp}}, \theta)\, \mathrm{d}s_{\mathrm{dp}}.$$

This is exactly where quasi-Monte Carlo can help.

A unified form of the PPE / PLE losses

Both losses have the same nested-integral structure:

$$\ell(\phi) = -\mathbb{E}_{p(\theta,\mathbf{x})}\left[\int \eta(s_{\mathrm{dp}} \mid s(\mathbf{x}))\, g_\phi(s_{\mathrm{dp}}, \theta)\, \mathrm{d}s_{\mathrm{dp}}\right].$$

Here

$$g_\phi(s_{\mathrm{dp}}, \theta) = \begin{cases} \log q_\phi(\theta \mid s_{\mathrm{dp}}), & \text{for PPE,} \\ \log q_\phi(s_{\mathrm{dp}} \mid \theta), & \text{for PLE.} \end{cases}$$

Key point

The optimization over ϕ is driven by repeated evaluation of the inner integral

$$I_\phi(\theta, \mathbf{x}) = \int \eta(s_{\mathrm{dp}} \mid s(\mathbf{x}))\, g_\phi(s_{\mathrm{dp}}, \theta)\, \mathrm{d}s_{\mathrm{dp}}.$$

This is exactly where quasi-Monte Carlo can help.

Monte Carlo estimator for the inner privacy integral

Suppose the privacy mechanism can be written as

$$s_{\mathrm{dp}} = \tau(u; s(\mathbf{x})), \qquad u \sim \mathrm{Unif}([0,1]^r),$$

where $r = \dim(\mathbb{S})$ is the dimension of the released private summary.

Then

$$I_\phi(\theta, \mathbf{x}) = \int_{[0,1]^r} g_\phi\big(\tau(u; s(\mathbf{x})), \theta\big)\, \mathrm{d}u.$$

Using i.i.d. uniforms $u^{(1)}, \ldots, u^{(M)}$, the standard MC estimator is

$$\widehat{I}^{\mathrm{MC}}_{\theta,\mathbf{x}} = \frac{1}{M} \sum_{m=1}^{M} g_\phi\big(\tau(u^{(m)}; s(\mathbf{x})), \theta\big),$$

with root-mean-squared-error (RMSE) typically of order

$$\mathcal{O}(M^{-1/2}).$$

Monte Carlo estimator for the inner privacy integral

Suppose the privacy mechanism can be written as

$$s_{\mathrm{dp}} = \tau(u; s(\mathbf{x})), \qquad u \sim \mathrm{Unif}([0,1]^r),$$

where $r = \dim(\mathbb{S})$ is the dimension of the released private summary.

Then

$$I_\phi(\theta, \mathbf{x}) = \int_{[0,1]^r} g_\phi\big(\tau(u; s(\mathbf{x})), \theta\big)\, \mathrm{d}u.$$

Using i.i.d. uniforms $u^{(1)}, \ldots, u^{(M)}$, the standard MC estimator is

$$\widehat{I}^{\mathrm{MC}}_{\theta,\mathbf{x}} = \frac{1}{M} \sum_{m=1}^{M} g_\phi\big(\tau(u^{(m)}; s(\mathbf{x})), \theta\big),$$

with root-mean-squared-error (RMSE) typically of order

$$\mathcal{O}(M^{-1/2}).$$

RQMC estimator for the inner privacy integral

Now replace the i.i.d. uniforms by randomized low-discrepancy points

$$v^{(1)}, \ldots, v^{(M)} \in [0,1]^r.$$

The RQMC estimator is

$$\widehat{I}_{\theta,\mathbf{x}}^{\mathrm{RQMC}} = \frac{1}{M} \sum_{m=1}^{M} g_\phi\big(\tau(v^{(m)}; s(\mathbf{x})), \theta\big).$$

If the integrand

$$\tilde{g}_{\theta,\mathbf{x}}(u) := g_\phi\big(\tau(u; s(\mathbf{x})), \theta\big)$$

has bounded Hardy–Krause variation, then

$$\mathbb{E}\Big[\big(\widehat{I}_{\theta,\mathbf{x}}^{\mathrm{RQMC}} - I_\phi(\theta,\mathbf{x})\big)^2\Big] = \mathcal{O}(M^{-2+2\delta}),$$

so the RMSE is

$$\mathcal{O}(M^{-1+\delta}), \qquad \delta > 0.$$

RQMC estimator for the inner privacy integral

Now replace the i.i.d. uniforms by randomized low-discrepancy points

$$v^{(1)}, \ldots, v^{(M)} \in [0,1]^r.$$

The RQMC estimator is

$$\widehat{I}_{\theta,\mathbf{x}}^{\mathrm{RQMC}} = \frac{1}{M} \sum_{m=1}^{M} g_\phi\big(\tau(v^{(m)}; s(\mathbf{x})), \theta\big).$$

If the integrand

$$\tilde{g}_{\theta,\mathbf{x}}(u) := g_\phi\big(\tau(u; s(\mathbf{x})), \theta\big)$$

has bounded Hardy–Krause variation, then

$$\mathbb{E}\Big[\big(\widehat{I}_{\theta,\mathbf{x}}^{\mathrm{RQMC}} - I_\phi(\theta, \mathbf{x})\big)^2\Big] = \mathcal{O}(M^{-2+2\delta}),$$

so the RMSE is

$$\mathcal{O}(M^{-1+\delta}), \qquad \delta > 0.$$

Why RQMC is especially natural here

- The difficult integral is over the released private query s_{dp}, not over the full confidential data $\mathbf{X}$.
- In many examples, $\dim(\mathbb{S})$ is small even when $\dim(\mathbb{X}^n)$ is very large.
- This is the regime where RQMC can substantially reduce variance.

Computation

For each training pair $(\theta, \mathbf{x})$, we approximate

$$I_\phi(\theta, \mathbf{x}) = \int \eta(s_{\mathrm{dp}} \mid s(\mathbf{x}))\, g_\phi(s_{\mathrm{dp}}, \theta)\, \mathrm{d}s_{\mathrm{dp}}$$

by RQMC, and then optimize the empirical loss over ϕ.

RQMC improves the inner-integral estimation

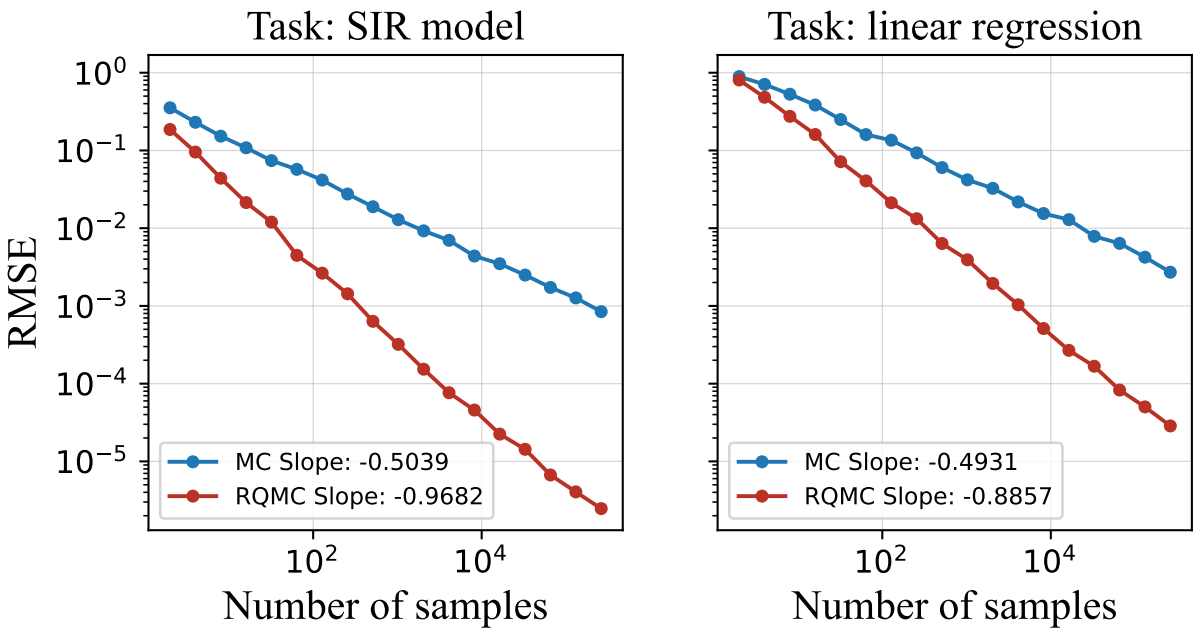


Figure: Rate of convergence.

- MC: RMSE typically scales as $M^{-1/2}$.
- RQMC: RMSE can approach M^{-1}.
- The gain is strongest when the released summary has low dimension.

Message

In PPE / PLE, the computational bottleneck is the inner privacy integral, so better integration directly improves training.

Performance comparison

Table: Estimated posterior mean and 95% credible intervals for the privatized Bayesian linear regression example using various methods. Here privacy loss budget is set to $\epsilon = 10$.

	β_0	β_1	β_2
Confidential Posterior	-2.15 (-2.68, -1.61)	-2.79 (-3.08, -2.50)	-0.83 (-1.08, -0.58)
Naive posterior	-4.63 (-5.04, -4.22)	-6.23 (-6.57, -5.90)	-5.10 (-5.40, -4.79)
DA-MCMC*	-0.62 (-2.50, 0.99)	-2.72 (-3.74, -0.96)	0.54 (-1.06, 2.46)
SMC-ABC	-0.59 (-2.28, 0.93)	-2.44 (-3.67, -0.38)	0.85 (-0.88, 2.77)
PPE	-0.51 (-2.25, 1.07)	-2.40 (-3.61, -0.30)	0.90 (-0.89, 2.85)
PLE	-0.64 (-2.31, 0.96)	-2.61 (-3.65, -0.98)	0.64 (-0.93, 2.48)
Gibbs-SS*	-0.46 (-2.22, 1.39)	-0.50 (-2.08, 1.28)	0.41 (-1.68, 2.13)
RNPE	-1.40 (-3.59, 0.94)	-2.15 (-3.34, 0.51)	0.29 (-2.11, 2.97)

Performance comparison

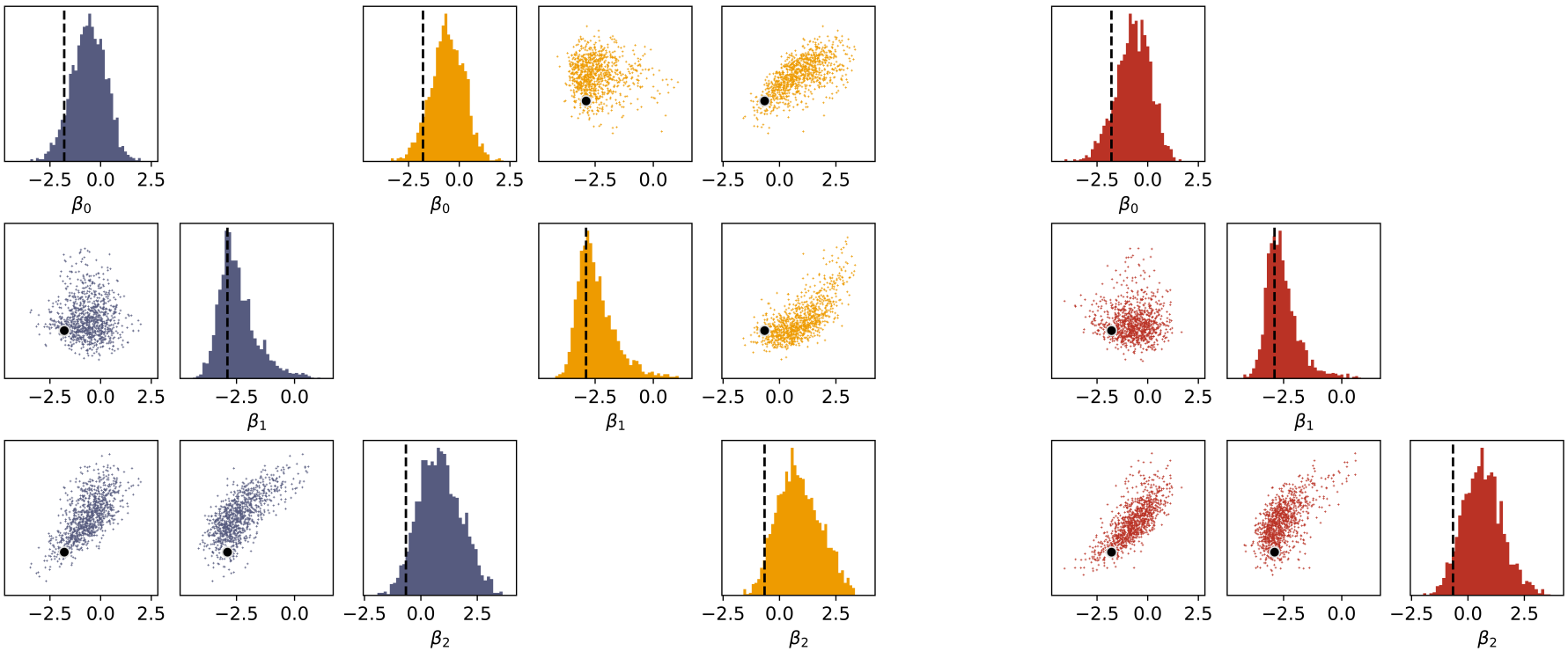


Figure: Posterior comparison on the Bayesian linear regression model. Grey: SMC-ABC; orange: PPE; red: PLE. The vertical lines and black dots indicate true data generating parameters.

Summary: one problem, two strategies

Original problem

The target posterior

$$\pi(\theta \mid s_{\mathrm{dp}}) \propto \pi(\theta)\, f(s_{\mathrm{dp}} \mid \theta)$$

is hard because the private likelihood

$$f(s_{\mathrm{dp}} \mid \theta) = \int_{\mathbb{X}^n} f(\mathbf{x} \mid \theta)\, \eta(s_{\mathrm{dp}} \mid s(\mathbf{x}))\, \mathrm{d}\mathbf{x}$$

is typically intractable.

Sampling-based strategy

Augment the latent confidential data and sample from $\pi(\theta, \mathbf{x} \mid s_{\mathrm{dp}})$ using MCMC; SOMA improves the imputation step.

Variational-based strategy

Approximate either the posterior (PPE) or the private likelihood (PLE); RQMC improves the numerical evaluation of the resulting loss.

References

Bernstein, G. and Sheldon, D. R. (2019). Differentially private Bayesian linear regression. *Advances in Neural Information Processing Systems*, 32.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. (2019). Neural spline flows. *Advances in Neural Information Processing Systems*, 32.

Dwork, C. (2006). Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer.

Ju, N., Awan, J., Gong, R., and Rao, V. (2022). Data augmentation MCMC for Bayesian inference from privatized data. *Advances in Neural Information Processing Systems*, 35:12732–12743.

Liu, S. (2024). Transport quasi-monte carlo. *arXiv preprint arXiv:2412.16416*.

Lueckmann, J.-M., Boelts, J., Greenberg, D., Goncalves, P., and Macke, J. (2021). Benchmarking simulation-based inference. In *International Conference on Artificial Intelligence and Statistics*, pages 343–351. PMLR.

Qin, Q. and Jones, G. L. (2022). Convergence rates of two-component MCMC samplers. *Bernoulli*, 28(2):859–885.

Xiong, Y. and Ju, N. P. (2025). SOMA: A novel sampler for Bayesian inference from privatized data. *arXiv preprint arXiv:2505.00635*.

Xiong, Y., Ju, N. P., and Zhang, S. (2025). Simulation-based Bayesian inference from privacy protected data. *Transactions of Machine Learning Research*.

Thank you

Thank you!

Questions and comments are welcome.

Random-Scan and Systematic-Scan IMwG

- To compare SOMA with more familiar samplers, we consider two standard **independent Metropolis-within-Gibbs** schemes targeting the same $\pi(\mathbf{x})$.

Single-coordinate IMwG update at index i

1. Propose $y \sim q(\cdot)$.
2. Form $\mathbf{x}' = [\mathbf{x}_{-i}, y]$.
3. Accept with probability $\alpha_i^{\mathrm{IMwG}}(y, \mathbf{x}) = 1 \wedge \frac{\pi(\mathbf{x}')\, q(x_i)}{\pi(\mathbf{x})\, q(y)}$.

- **Random-scan IMwG (Ran-scan):**
  - at each step, draw $I \sim \mathrm{Unif}\{1, \ldots, n\}$;
  - perform the IMwG update at coordinate I.
- **Systematic-scan IMwG (Sys-scan):**
  - in each sweep, visit $i = 1, \ldots, n$ in a fixed order;
  - at each i perform the IMwG update above.
- See Qin and Jones (2022) for a detailed comparison of Ran-scan and Sys-scan when $n = 2$.

Theoretical Comparison: SOMA vs IMwG

Theorem (Acceptance dominance)

For any state $\mathbf{x}$, coordinate i and proposal y,

$$\alpha_i^{\mathrm{SOMA}}(y, \mathbf{x}) \;\geq\; \alpha_i^{\mathrm{IMwG}}(y, \mathbf{x}). \tag{2}$$

Let $A^{\mathrm{SOMA}}(\mathbf{x})$ and $A^{\mathrm{RAN}}(\mathbf{x})$ denote the overall acceptance probabilities from $\mathbf{x}$ for SOMA and Ran-scan IMwG, respectively. Then

$$A^{\mathrm{SOMA}}(\mathbf{x}) \;\geq\; A^{\mathrm{RAN}}(\mathbf{x}) \quad \text{for all } \mathbf{x}. \tag{3}$$

Corollary (DP-based lower bounds)

If the mechanism is ϵ-differentially private, we can show

$$A^{\mathrm{SOMA}}(\mathbf{x}) \;\geq\; \frac{n}{n + e^{\epsilon} - 1}, \qquad A^{\mathrm{RAN}}(\mathbf{x}) \;\geq\; e^{-\epsilon}. \tag{4}$$

- As $n \to \infty$, SOMA becomes almost rejection-free, while the bound for Ran-scan IMwG stays at $e^{-\epsilon}$.

Theoretical Comparison: SOMA vs IMwG

Theorem (Acceptance dominance)

For any state $\mathbf{x}$, coordinate i and proposal y,

$$\alpha_i^{\mathrm{SOMA}}(y, \mathbf{x}) \;\geq\; \alpha_i^{\mathrm{IMwG}}(y, \mathbf{x}). \tag{2}$$

Let $A^{\mathrm{SOMA}}(\mathbf{x})$ and $A^{\mathrm{RAN}}(\mathbf{x})$ denote the overall acceptance probabilities from $\mathbf{x}$ for SOMA and Ran-scan IMwG, respectively. Then

$$A^{\mathrm{SOMA}}(\mathbf{x}) \;\geq\; A^{\mathrm{RAN}}(\mathbf{x}) \quad \text{for all } \mathbf{x}. \tag{3}$$

Corollary (DP-based lower bounds)

If the mechanism is ϵ-differentially private, we can show

$$A^{\mathrm{SOMA}}(\mathbf{x}) \;\geq\; \frac{n}{n + e^{\epsilon} - 1}, \qquad A^{\mathrm{RAN}}(\mathbf{x}) \;\geq\; e^{-\epsilon}. \tag{4}$$

- As $n \to \infty$, SOMA becomes almost rejection-free, while the bound for Ran-scan IMwG stays at $e^{-\epsilon}$.

Convergence Rate under ϵ-DP

- We measure convergence by **geometric ergodicity**: for a kernel P with invariant distribution π,

$$\|P^t(\cdot \mid x) - \pi(\cdot)\|_{\mathrm{TV}} \;\le\; C(x)\, r^t,$$

where $r \in (0,1)$ is the **convergence rate** (smaller $r \Rightarrow$ faster mixing).

Theorem (Upper bound on convergence rate)

For the case $n = 2$, the convergence rate of the SOMA algorithm is upper bounded by

$$r_{\mathrm{SOMA}} \le \frac{e^{2\epsilon} + 3e^{\epsilon} - 2 + \sqrt{e^{4\epsilon} + 2e^{3\epsilon} + 9e^{2\epsilon} - 8e^{\epsilon}}}{2(1+e^{\epsilon})^2}. \tag{5}$$

Smaller loss budget ϵ (i.e., larger DP noise) leads to a smaller upper bound r_{SOMA} and hence faster convergence in this example.

Computational Cost per iteration

- We compare SOMA with a single IMwG update from the viewpoint of **computational work per proposal** y.
- Both methods use the same proposal distribution $q(\cdot)$ and draw one proposal y per iteration.

Evaluating the target

- IMwG at index i: evaluate $\pi(\mathbf{x})$ and $\pi([\mathbf{x}_{-i}, y])$ for a *single* coordinate.
- SOMA: evaluate $\pi([\mathbf{x}_{-i}, y])$ for *all* $i = 1, \ldots, n$ plus $\pi(\mathbf{x})$ (to form w_0).

- Naively, this is an extra factor of n target evaluations for SOMA.
- However these n weights can be computed in **parallel** (or vectorized). In our DP applications, the wall-clock time per SOMA step is still comparable to a single-try IMwG update.

SOMA is a good sampler, but not magical

Drawbacks

- It is still based on an independent Metropolis proposal; performance depends on how well q matches π.
- Although the time cost may be comparable, the computational cost remains large.

Good news: when does SOMA shine?

- Target is **permutation-invariant** and naturally written as n exchangeable coordinates.
- Expensive part of $\pi(\mathbf{x})$ can be evaluated in **parallel** for all coordinates.
- In this setting, SOMA
  - strictly dominates random-scan IMwG in acceptance probability;
  - can be almost rejection-free when n is large;
  - empirically mixes faster than both random- and systematic-scan IMwG.

Normalizing flows

Start from a simple base density

$$z \sim q_0(z), \qquad \text{e.g. } q_0 = N(0, I_d),$$

and apply an invertible map

$$x = \tau(z).$$

Change of variables

If τ is invertible and differentiable, then

$$q_\tau(x) = q_0(z) \left|\det J_\tau(z)\right|^{-1}, \qquad z = \tau^{-1}(x).$$

Hence

$$\log q_\tau(x) = \log q_0(\tau^{-1}(x)) - \log \left|\det J_\tau(\tau^{-1}(x))\right|.$$

- A normalizing flow chooses τ to be flexible but still easy to invert and differentiate.
- Relatedly, Liu (2024) studies QMC-friendly transport maps inspired by normalizing flows.