

# Gradient-Free Optimal Postprocessing of MCMC Output

Artem Glebov

King's College London

2024

# Overview

## Problem

Develop a computationally efficient algorithm for summarising the output of a Markov Chain Monte Carlo simulation.

## Motivation

Uncertainty quantification in a multi-stage simulation of the functioning of the human heart.

## Existing solution

The optimisation algorithm of Riabiz et al. (2022) to select a subsample of MCMC output that minimises a measure of proximity to the target distribution (kernel Stein discrepancy), which requires the gradients of the log-posterior and is thus expensive.

## Proposal

Modify the algorithm of Riabiz et al. (2022) to use the gradient-free kernel Stein discrepancy of Fisher and Oates (2024).

# Table of Contents

## 1 Background

- Markov Chain Monte Carlo (MCMC)
- Challenges of running MCMC
- Stein thinning
- Gradient-free kernel Stein discrepancy

## 2 Methodology

- Proposed algorithm
- Evaluation

## 3 Results

- Bivariate Gaussian mixture
- Lotka-Volterra inverse problem

## 4 Conclusions

## 5 Further Research

## 6 References

# Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) are a popular class of algorithms for sampling from complex probability distributions.

Given a target distribution  $P$  defined on a state space  $\mathcal{X}$ , an MCMC algorithm proceeds by constructing a chain of random variables  $(X_i)_{i=0}^{\infty}$  which satisfy the Markov property:

$$\mathbb{P}(X_{i+1} \in A | X_0, \dots, X_i) = \mathbb{P}(X_{i+1} \in A | X_i) \quad \text{for any measurable } A \in \mathcal{X}.$$

Viewed as a function, the right-hand side above is called the Markov transition kernel and is denoted

$$R(A|x) := \mathbb{P}(X_{i+1} \in A | X_i = x).$$

The transition kernel  $R$  is selected so that it is easy to sample from and to ensure asymptotic convergence to the target distribution  $P$ :

$$P_i \xrightarrow{d} P \quad \text{as } i \rightarrow \infty.$$

A sample of size  $n$  is a realisation  $(x_i)_{i=0}^n$  of the first  $n$  variables in the chain, which is constructed sequentially.

# Challenges of running MCMC

- ① Dependency on the starting point of a chain
  - ⇒ detection and removal of burn-in
  - ⇒ running multiple chains
- ② Efficient traversal of the target domain
  - ⇒ calibration of the scale of the proposal distribution
- ③ Exploring the modes of a multimodal distribution
  - ⇒ calibration of the scale of the proposal distribution
  - ⇒ running multiple chains
- ④ Convergence detection
  - ⇒ convergence diagnostics
  - ⇒ running multiple chains
- ⑤ Autocorrelation between samples in a chain
  - ⇒ thinning
- ⑥ Summarising the sample for further expensive processing
  - ⇒ thinning

## Problem

Given MCMC output  $(x_i)_{i=1}^n$  of length  $n$ , identify  $m \ll n$  indices  $\pi(j) \in \{1, \dots, n\}$  with  $j \in \{1, \dots, m\}$ , such that the approximation provided by the subset of samples

$$\frac{1}{m} \sum_{j=1}^m \delta(x_{\pi(j)})$$

is closest to the target distribution.

We need a measure of proximity of the selected subsample to the target distribution.

# Measure of proximity

## Integral probability metric

An integral probability metric between two distributions  $P$  and  $P'$  is defined as

$$\mathcal{D}_{\mathcal{F}}(P, P') := \sup_{f \in \mathcal{F}} \left| \int_{\mathcal{X}} f \, dP - \int_{\mathcal{X}} f \, dP' \right|,$$

where  $\mathcal{X}$  is a measurable space on which both  $P$  and  $P'$  are defined and  $\mathcal{F}$  is a set of test functions.

The metric is said to be *measure-determining* if

$$\mathcal{D}_{\mathcal{F}}(P, P') = 0 \quad \text{iff} \quad P = P',$$

and it offers *convergence control* if

$$\mathcal{D}_{\mathcal{F}}(P, P'_m) \rightarrow 0 \quad \text{implies} \quad P'_m \xrightarrow{d} P$$

as  $m \rightarrow \infty$ , for any sequence of distributions  $P'_m$ .

## Integral probability metric

An integral probability metric between two distributions  $P$  and  $P'$  is defined as

$$\mathcal{D}_{\mathcal{F}}(P, P') := \sup_{f \in \mathcal{F}} \left| \int_{\mathcal{X}} f \, dP - \int_{\mathcal{X}} f \, dP' \right|,$$

where  $\mathcal{X}$  is a measurable space on which both  $P$  and  $P'$  are defined and  $\mathcal{F}$  is a set of test functions.

However, it is **difficult to compute** in practice:

- the integral  $\int_{\mathcal{X}} f \, dP$  is often intractable,
- the supremum requires optimisation.



# Stein discrepancy

## Integral probability metric

An integral probability metric between two distributions  $P$  and  $P'$  is defined as

$$\mathcal{D}_{\mathcal{F}}(P, P') := \sup_{f \in \mathcal{F}} \left| \int_{\mathcal{X}} f \, dP - \int_{\mathcal{X}} f \, dP' \right|,$$

where  $\mathcal{X}$  is a measurable space on which both  $P$  and  $P'$  are defined and  $\mathcal{F}$  is a set of test functions.

## Idea

Avoid the need to evaluate  $\int_{\mathcal{X}} f \, dP$  by choosing a set of functions  $\mathcal{F}$  such that  $\int_{\mathcal{X}} f \, dP = 0$  for all  $f \in \mathcal{F}$ .

# Stein discrepancy (continued)

Gorham and Mackey (2015) observed that the infinitesimal generator of a Markov process  $(Z_t)_{t \geq 0}$  given by

$$(\mathcal{L}u)(x) := \lim_{t \rightarrow 0} \frac{\mathbb{E}[u(Z_t) | Z_0 = x] - u(x)}{t} \quad \text{for } u : \mathbb{R}^d \rightarrow \mathbb{R}$$

satisfies

$$\mathbb{E}[(\mathcal{L}u)(Z)] = 0$$

under mild conditions on  $\mathcal{L}$  and  $u$ .

In the specific case of an overdamped Langevin diffusion

$$dZ_t = \frac{1}{2} \nabla \log p(Z_t) dt + dW_t,$$

where  $p$  is the density of  $P$  and  $W_t$  is the standard Brownian motion, the infinitesimal generator becomes

$$(\mathcal{L}_P u)(x) = \frac{1}{2} \langle \nabla u(x), \nabla \log p(x) \rangle + \frac{1}{2} \langle \nabla, \nabla u(x) \rangle.$$

# Stein discrepancy (continued)

The infinitesimal generator of an overdamped Langevin diffusion:

$$(\mathcal{L}_P u)(x) = \frac{1}{2} \langle \nabla u(x), \nabla \log p(x) \rangle + \frac{1}{2} \langle \nabla, \nabla u(x) \rangle.$$

Denoting  $g = \frac{1}{2} \nabla u$ , Gorham and Mackey (2015) obtain the Stein operator

$$\mathcal{A}_P g := \langle g, \nabla \log p \rangle + \langle \nabla, g \rangle = \langle p^{-1} \nabla, p g \rangle,$$

and rewrite the expression for the integral probability metric as

$$\mathcal{D}_{P,\mathcal{G}}(P') = \sup_{g \in \mathcal{G}} \left| \int_{\mathcal{X}} \mathcal{A}_P g \, dP' \right|$$

for a suitably chosen set  $\mathcal{G}$ .

# Stein discrepancy (continued)

Using the Langevin Stein operator, the integral probability metric specialises to

## Stein discrepancy

$$\mathcal{D}_{P,\mathcal{G}}(P') = \sup_{g \in \mathcal{G}} \left| \int_{\mathcal{X}} \mathcal{A}_P g \, dP' \right|$$

The difficulty evaluating the supremum still remains.

## Idea

Employ the kernel trick to eliminate the supremum in the expression for the integral probability metric.

# Reproducing kernel Hilbert space

A *Hilbert space* is a vector space  $V$  equipped with the inner product operation  $\langle \cdot, \cdot \rangle$  and its induced norm  $\| \cdot \|$  satisfying  $\|v\|^2 = \langle v, v \rangle$  for all  $v \in V$ , if it is complete:

$$\sum_{i=1}^{\infty} \|v_i\| < \infty \quad \text{implies} \quad \sum_{i=1}^{\infty} v_i \in V$$

for any sequence  $v_i \in V$ .

A Hilbert space  $\mathcal{H}$  of real-valued functions defined on a set  $\mathcal{X}$  is called a *reproducing kernel Hilbert space (RKHS)* if there exists a function  $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  such that:

- for every  $x \in \mathcal{X}$ , the function  $k(x, \cdot)$  belongs to  $\mathcal{H}$ ,
- $k$  satisfies the reproducing property  $\langle f(\cdot), k(\cdot, x) \rangle = f(x)$  for any  $f \in \mathcal{H}$  and  $x \in \mathcal{X}$ .

We denote  $\mathcal{H}(k)$  the RKHS with kernel  $k$ .

# Kernel Stein discrepancy

Taking the unit-ball in a Cartesian product of  $d$  copies  $\mathcal{H}(k)$

$$\mathcal{G} := \left\{ g : \mathbb{R}^d \rightarrow \mathbb{R}^d \left| \sum_{i=1}^d \|g_i\|_{\mathcal{H}(k)}^2 \leq 1 \right. \right\},$$

Proposition 2 in Gorham and Mackey (2017) shows that the Stein discrepancy becomes

$$\mathcal{D}_P^2(P') := \mathcal{D}_{P,\mathcal{G}}(P') = \iint_{\mathcal{X}} k_P(x, y) \, \mathrm{d}p'(x) \, \mathrm{d}p'(y),$$

where  $p'$  is the density of  $P'$ , and  $k_P(x, y)$  is given by

$$\begin{aligned} k_P(x, y) := & (\nabla_x \cdot \nabla_y) k(x, y) \\ & + \langle \nabla_x k(x, y), \nabla_y \log p(y) \rangle + \langle \nabla_y k(x, y), \nabla_x \log p(x) \rangle \\ & + k(x, y) \langle \nabla_x \log p(x), \nabla_y \log p(y) \rangle. \end{aligned}$$

# Kernel Stein discrepancy (continued)

## Kernel Stein discrepancy (KSD)

$$\mathcal{D}_P^2(P') := \iint_{\mathcal{X}} k_P(x, y) \, \mathrm{d}p'(x) \, \mathrm{d}p'(y),$$

If  $P'$  is the discrete distribution, this becomes

$$\mathcal{D}_P^2\left(\frac{1}{n} \sum_{i=1}^n \delta(x_i)\right) = \frac{1}{n^2} \sum_{i,j=1}^n k_P(x_i, x_j),$$

where

$$\begin{aligned} k_P(x, y) := & (\nabla_x \cdot \nabla_y) k(x, y) \\ & + \langle \nabla_x k(x, y), \nabla_y \log p(y) \rangle + \langle \nabla_y k(x, y), \nabla_x \log p(x) \rangle \\ & + k(x, y) \langle \nabla_x \log p(x), \nabla_y \log p(y) \rangle. \end{aligned}$$

The typical choice for  $k(x, y)$  is the inverse multiquadric kernel:

$$k(x, y) = \left( c^2 + \|\Gamma^{-1/2}(x - y)\| \right)^\beta.$$

# Inverse multiquadric kernel

The common choice of the kernel  $k$  is the inverse multiquadric kernel (IMQ)

$$k(x, y) = \left( c^2 + \|\Gamma^{-1/2}(x - y)\| \right)^\beta.$$

When  $\beta \in (-1, 0)$  and  $\Gamma = I$ , Gorham and Mackey (2017) demonstrate that  $\mathcal{D}_P(P')$  provides convergence control (Theorem 8). Theorem 4 in Chen et al. (2019) justifies the introduction of  $\Gamma$  in IMQ.



# Stein thinning

Riabiz et al. (2022) propose a greedy algorithm to select points from the sample that minimise the KSD at each iteration:

---

**Algorithm 1:** Stein thinning.

---

**Data:**

sample  $(x_i)_{i=1}^n$  from MCMC,

gradients  $(\nabla \log p(x_i))_{i=1}^n$

desired cardinality  $m \in \mathbb{N}$

**Result:** Indices  $\pi$  of a sequence  $(x_{\pi(j)})_{j=1}^m$  where  $\pi(j) \in \{1, \dots, n\}$ .

**for**  $j = 1, \dots, m$  **do**

$$\pi(j) \in \arg \min_{i=1, \dots, n} \frac{k_P(x_i, x_i)}{2} + \sum_{j'=1}^{j-1} k_P(x_{\pi(j')}, x_i)$$

**end**

---

# Stein thinning (continued)

The complication in using Stein thinning comes from the need to calculate gradients of the log-posterior to evaluate the kernel:

$$\begin{aligned} k_P(x, y) := & (\nabla_x \cdot \nabla_y) k(x, y) \\ & + \langle \nabla_x k(x, y), \nabla_y \log p(y) \rangle + \langle \nabla_y k(x, y), \nabla_x \log p(x) \rangle \\ & + k(x, y) \langle \nabla_x \log p(x), \nabla_y \log p(y) \rangle. \end{aligned}$$

This might be expensive, although it can be easily parallelised.

# Gradient-free kernel Stein discrepancy

Fisher and Oates (2024) introduce a gradient-free version of KSD. An auxiliary distribution  $Q$  need to be chosen by the user, then the gradient-free KSD is given by

$$k_{P,Q}(x, y) = \frac{q(x)}{p(x)} \frac{q(y)}{p(y)} k_Q(x, y),$$

where

$$\begin{aligned} k_Q(x, y) := & (\nabla_x \cdot \nabla_y) k(x, y) \\ & + \langle \nabla_x k(x, y), \nabla_y \log q(y) \rangle + \langle \nabla_y k(x, y), \nabla_x \log q(x) \rangle \\ & + k(x, y) \langle \nabla_x \log q(x), \nabla_y \log q(y) \rangle. \end{aligned}$$

When  $k(x, y)$  is the inverse multiquadric kernel, the gradient-free KSD offers convergence control (Theorem 2 in Fisher and Oates (2024)).

# Table of Contents

- 1 Background
  - Markov Chain Monte Carlo (MCMC)
  - Challenges of running MCMC
  - Stein thinning
  - Gradient-free kernel Stein discrepancy
- 2 Methodology
  - Proposed algorithm
  - Evaluation
- 3 Results
  - Bivariate Gaussian mixture
  - Lotka-Volterra inverse problem
- 4 Conclusions
- 5 Further Research
- 6 References

# Gradient-free Stein thinning

We modify the algorithm of Riabiz et al. (2022) to minimise the gradient-free KSD of Fisher and Oates (2024):

---

**Algorithm 2:** Gradient-free Stein thinning.

---

**Data:**

- sample  $(x_i)_{i=1}^n$  from MCMC,
- target log-densities  $(\log p(x_i))_{i=1}^n$
- auxiliary log-densities  $(\log q(x_i))_{i=1}^n$
- auxiliary gradients  $(\nabla \log q(x_i))_{i=1}^n$
- desired cardinality  $m \in \mathbb{N}$

**Result:** Indices  $\pi$  of a sequence  $(x_{\pi(j)})_{j=1}^m$  where  $\pi(j) \in \{1, \dots, n\}$ .

**for**  $j = 1, \dots, m$  **do**

$$\pi(j) \in \arg \min_{i=1, \dots, n} \frac{k_{P,Q}(x_i, x_i)}{2} + \sum_{j'=1}^{j-1} k_{P,Q}(x_{\pi(j')}, x_i)$$

**end**

---

**Algorithm 3:** Optimised gradient-free Stein thinning.

---

**Data:**

sample  $(x_i)_{i=1}^n$  from MCMC,  
target log-densities  $(\log p(x_i))_{i=1}^n$   
auxiliary log-densities  $(\log q(x_i))_{i=1}^n$   
auxiliary gradients  $(\nabla \log q(x_i))_{i=1}^n$   
desired cardinality  $m \in \mathbb{N}$ .

**Result:** Indices  $\pi$  of a sequence  $(x_{\pi(j)})_{j=1}^m$  where  $\pi(j) \in \{1, \dots, n\}$ .

Initialise an array  $A[i]$  of size  $n$

Set  $A[i] = k_{P,Q}(x_i, x_i)$  for  $i = 1, \dots, n$

Set  $\pi(1) = \arg \min_i A[i]$

**for**  $j = 2, \dots, m$  **do**

    Update  $A[i] = A[i] + 2k_{P,Q}(x_{\pi(j-1)}, x_i)$  for  $i = 1, \dots, n$

    Set  $\pi(j) = \arg \min_i A[i]$

**end**

---

The following protocol was used in the evaluating the new method:

- ① obtain a sample from the target distribution (depending on the test case, the sampling is done either i.i.d. or via MCMC),
- ② apply naïve thinning, Stein thinning and the proposed algorithm to get a thinned sample of a given cardinality,
- ③ evaluate the result of thinning using an impartial metric.

# Energy distance

In order to assess how well the selected sample approximates the target distribution, we use the energy distance.

## Energy distance (Rizzo and Székely (2016))

The squared energy distance is defined for two distributions  $P$  and  $Q$  as

$$D_e^2(P, Q) := 2\mathbb{E}\|X - Y\| - \mathbb{E}\|X - X'\| - \mathbb{E}\|Y - Y'\|,$$

where  $X, X' \sim P$  and  $Y, Y' \sim Q$ .

For samples  $x_1, \dots, x_n$  and  $y_1, \dots, y_m$  from  $X$  and  $Y$ , respectively, the corresponding statistic is given by

$$\mathcal{E}_{n,m}(P, Q) := \frac{2}{nm} \sum_{i=1}^n \sum_{j=1}^m \|x_i - y_j\| - \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \|x_i - x_j\| - \frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m \|y_i - y_j\|.$$



# Table of Contents

- 1 Background
  - Markov Chain Monte Carlo (MCMC)
  - Challenges of running MCMC
  - Stein thinning
  - Gradient-free kernel Stein discrepancy
- 2 Methodology
  - Proposed algorithm
  - Evaluation
- 3 Results
  - Bivariate Gaussian mixture
  - Lotka-Volterra inverse problem
- 4 Conclusions
- 5 Further Research
- 6 References

# Bivariate Gaussian mixture

We use the bivariate Gaussian mixture with means

$$\mu_1 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \quad \mu_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

covariance matrices

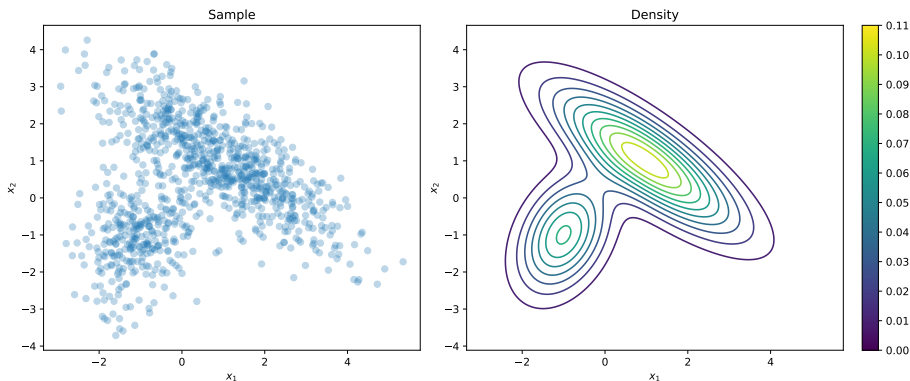
$$\Sigma_1 = \begin{pmatrix} 0.5 & 0.25 \\ 0.25 & 1 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 2 & -0.8\sqrt{3} \\ -0.8\sqrt{3} & 1.5 \end{pmatrix}$$

and weights

$$w = \begin{pmatrix} 0.3 \\ 0.7 \end{pmatrix}.$$

# Bivariate Gaussian mixture: sample

We obtain 1000 points by directly drawing from the target distribution:

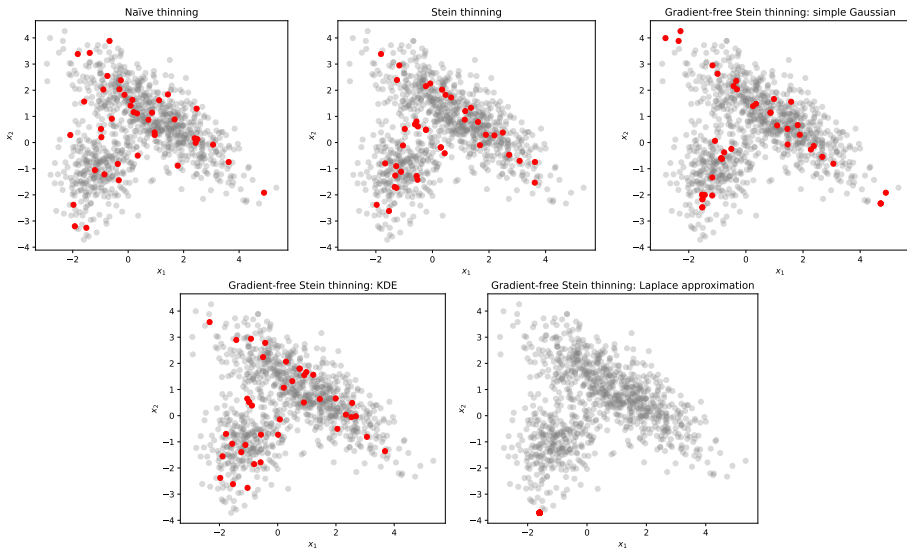


# Bivariate Gaussian mixture: thinning approaches

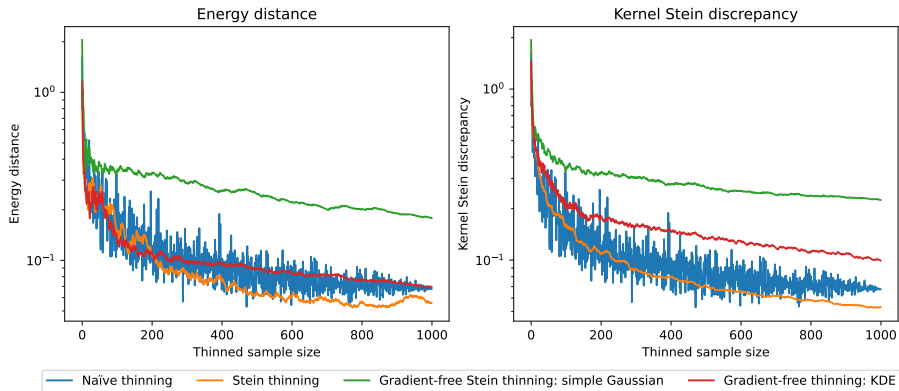
We evaluate the following approaches:

- naïve thinning,
- standard Stein thinning,
- gradient-free Stein thinning with different choices of  $Q$ :
  - multivariate Gaussian using the sample mean and covariance,
  - Laplace approximation,
  - KDE approximation.

# Bivariate Gaussian mixture: thinning results



# Bivariate Gaussian mixture: comparison of approaches



# Lotka-Volterra inverse problem

The Lotka-Volterra model describes the evolution of an idealised ecosystem with two species: predator and prey.

Let  $u_1$  be the size of the prey population and  $u_2$  the size of the predator population. The model then postulates the following dynamic:

$$\begin{aligned}\frac{du_1}{dt} &= \theta_1 u_1 - \theta_2 u_1 u_2, \\ \frac{du_2}{dt} &= -\theta_3 u_2 + \theta_4 u_1 u_2,\end{aligned}$$

with  $\theta_1, \dots, \theta_4 > 0$ .

The inverse problem: given a noisy realisation

$$y(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} + \varepsilon(t),$$

infer  $\theta = (\theta_1, \dots, \theta_4)^T$  that best describes the observed behaviour.

# Lotka-Volterra inverse problem: synthetic data

## Lotka-Volterra model

$$\begin{aligned}\frac{du_1}{dt} &= \theta_1 u_1 - \theta_2 u_1 u_2, \\ \frac{du_2}{dt} &= -\theta_3 u_2 + \theta_4 u_1 u_2,\end{aligned}$$

We solve the model with parameters

$$\theta^* = (0.67, 1.33, 1, 1)^T$$

and initial values

$$u(0) = (1, 1)^T$$

for  $t \in [0, 25]$  discretised into  $N = 2400$  points.

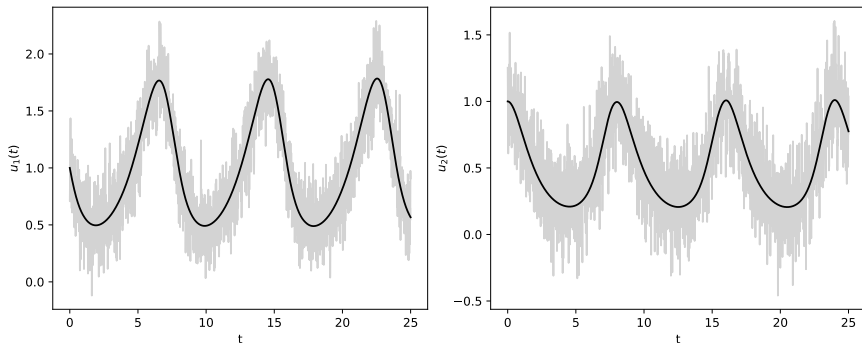
We then add bivariate i.i.d. Gaussian noise:

$$\varepsilon(t) \sim \mathcal{N}(0, \text{diag}(0.2^2, 0.2^2)).$$



# Lotka-Volterra inverse problem: synthetic data

The resulting noisy data is used as the input for the inverse problem:



# Lotka-Volterra inverse problem: Bayesian inference

Assuming independent observations, we take the likelihood to be

$$\mathcal{L}(\theta) = \prod_{i=1}^N \phi_i(u(t_i; \theta)),$$

where

$$\phi_i(u(t_i; \theta)) \propto \exp \left( -\frac{1}{2} (y(t_i) - u(t_i; \theta))^T C^{-1} (y(t_i) - u(t_i; \theta)) \right)$$

with  $C = \text{diag}(0.2^2, 0.2^2)$ .

Since  $\theta_k > 0$ , we put independent log-normal priors on each  $\theta_k$ :

$$\pi(\theta) \propto \exp \left( -\frac{1}{2} (\log \theta)^T (\log \theta) \right).$$

By the Bayes theorem, the posterior is then

$$p(\theta) \propto \mathcal{L}(\theta) \pi(\theta).$$

# Lotka-Volterra inverse problem: MCMC

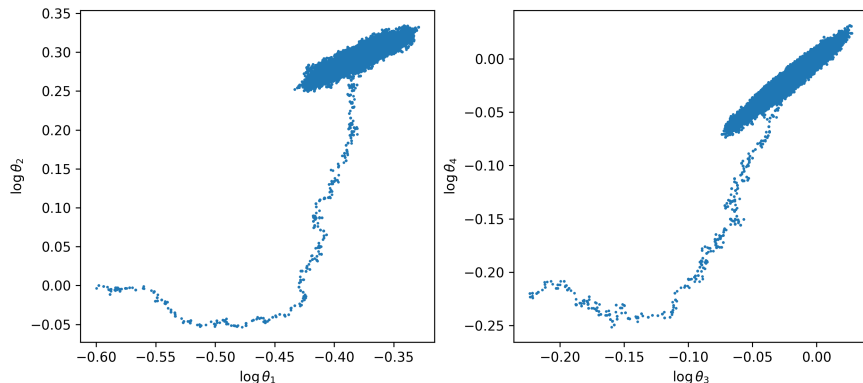
The inference is performed using the Metropolis-Hastings algorithm with the starting points taken from Riabiz et al. (2022):

Chain	$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$
1	0.55	1	0.8	0.8
2	1.5	1	0.8	0.8
3	1.3	1.33	0.5	0.8
4	0.55	3	3.	0.8
5	0.55	1	1.5	1.5

Since the parameters  $\theta_k$  of the Lotka-Volterra model are positive, we run MCMC in the log-space by applying the reparameterisation  $\zeta_k = \log \theta_k$ . We run 500,000 iterations of the algorithm for each chain.

# Lotka-Volterra inverse problem: MCMC sample

The sample from the first chain is shown here for illustration:



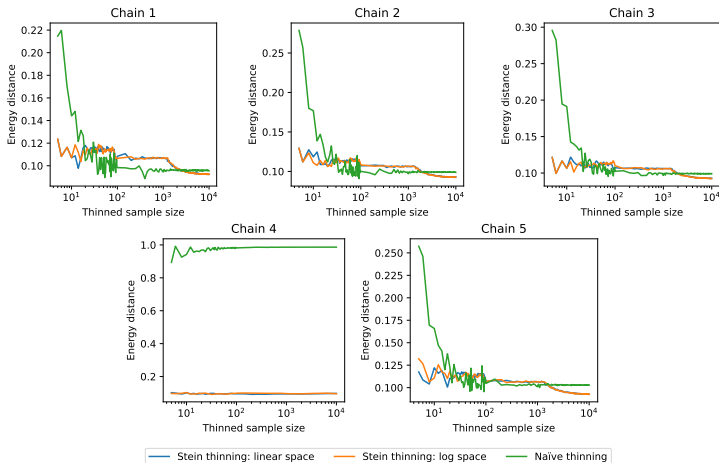
# Lotka-Volterra inverse problem: thinning approaches

We evaluate the following approaches:

- naïve thinning,
- standard Stein thinning,
- gradient-free Stein thinning with different choices of  $Q$ :
  - multivariate Gaussian using the sample mean and covariance,
  - Laplace approximation - **fails as in bivariate Gaussian case**,
  - KDE approximation - **computationally infeasible**.

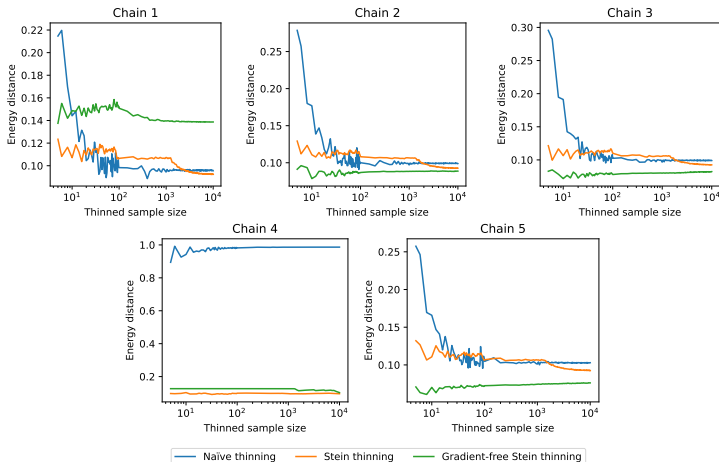
# Lotka-Volterra inverse problem: Stein thinning

We have a choice of applying thinning in linear or logarithmic space, however the energy distance comparison indicates no discernible difference, so we proceed to use the logarithmic space.



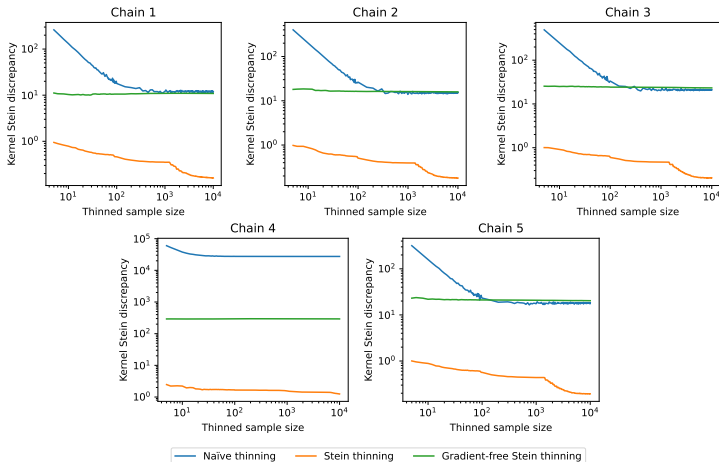
# Lotka-Volterra inverse problem: comparison of approaches

Gradient-free thinning using a multivariate Gaussian with the sample mean and covariance matrix performs comparably with the gradient-based approach, notably for chain 4, when compared using the energy distance:



# Lotka-Volterra inverse problem: comparison of approaches

Gradient-free thinning offers a significant improvement over naïve thinning in terms of KSD:





# Table of Contents

- 1 Background
  - Markov Chain Monte Carlo (MCMC)
  - Challenges of running MCMC
  - Stein thinning
  - Gradient-free kernel Stein discrepancy
- 2 Methodology
  - Proposed algorithm
  - Evaluation
- 3 Results
  - Bivariate Gaussian mixture
  - Lotka-Volterra inverse problem
- 4 Conclusions
- 5 Further Research
- 6 References

The project makes three contributions:

- implementation of the gradient-free Stein thinning algorithm in the Python library `stein-thinning`,
- evaluation of the performance of the proposed algorithm,
- improvement of the computational efficiency of the existing Stein thinning algorithm from  $O(nm^2)$  to  $O(nm)$ , where  $n$  is the input sample size and  $m$  is the desired thinned sample size.

# Conclusions

- The gradient-free approach is feasible and performs similarly to the Stein thinning algorithm of Riabiz et al. (2022) for small thinned sample sizes,
- The performance of the algorithm depends crucially on the choice of the auxiliary distribution. For example, even in the highly favourable setting of i.i.d. samples from a Gaussian mixture, choosing the auxiliary distribution based on the Laplace approximation fails to produce a thinned sample.
- The simple multivariate Gaussian distribution using the sample mean and covariance offered a good starting point in our experiments, however bespoke treatment might be required for more complex problems.
- In deciding whether to use the new algorithm as opposed to the gradient-based approach, the effort involved in selecting a good auxiliary distribution must be weighed against the computational cost of obtaining gradients.

# Table of Contents

- 1 Background
  - Markov Chain Monte Carlo (MCMC)
  - Challenges of running MCMC
  - Stein thinning
  - Gradient-free kernel Stein discrepancy
- 2 Methodology
  - Proposed algorithm
  - Evaluation
- 3 Results
  - Bivariate Gaussian mixture
  - Lotka-Volterra inverse problem
- 4 Conclusions
- 5 Further Research
- 6 References

# Further Research

- Evaluate the choices of KDE kernels other than Gaussian for constructing the auxiliary distribution.
- Parallelise the computation of KDE.
- Perform thinning in a lower-dimensional space.
- Investigate the behaviour of Stein thinning for large thinned sample sizes.
- Compare the performance of the approaches in terms of estimating the true parameters of the Lotka-Volterra model.
- Run an experiment with randomised starting points.

# Further Research (continued)

- Repeat the experiments with more advanced MCMC algorithms.
- Check how running a gradient-free MCMC sampling algorithm (such the random-walk Metropolis-Hastings) followed by Stein thinning of the sample compares to running a gradient-based sampling algorithm (e.g. HMC).
- Provide theoretical justification for gradient-free Stein thinning.
- Explore other gradient-free alternatives.

# Table of Contents

- 1 Background
  - Markov Chain Monte Carlo (MCMC)
  - Challenges of running MCMC
  - Stein thinning
  - Gradient-free kernel Stein discrepancy
- 2 Methodology
  - Proposed algorithm
  - Evaluation
- 3 Results
  - Bivariate Gaussian mixture
  - Lotka-Volterra inverse problem
- 4 Conclusions
- 5 Further Research
- 6 References

# References I

- W. Y. Chen, A. Barp, F.-X. Briol, J. Gorham, M. Girolami, L. Mackey, and Chris. J. Oates. Stein Point Markov Chain Monte Carlo. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1011–1021, Long Beach, California, 2019. PLMR.
- M. A. Fisher and C. J. Oates. Gradient-Free Kernel Stein Discrepancy. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 23855–23885, May 2024.
- J. Gorham and L. Mackey. Measuring Sample Quality with Stein’s Method. In *Advances in Neural Information Processing Systems*, volume 28, pages 226–234. MIT Press, 2015.
- J. Gorham and L. Mackey. Measuring Sample Quality with Kernels. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1292–1301, Sydney, Australia, 2017. PLMR.
- M. Riabiz, W. Y. Chen, J. Cockayne, P. Swietach, S. A. Niederer, L. Mackey, and Chris. J. Oates. Optimal Thinning of MCMC Output. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(4):1059–1081, September 2022.



M. L. Rizzo and G. J. Székely. Energy Distance. *WIREs Computational Statistics*, 8(1):27–38, January 2016.