# Stochastic Gradient MCMC for Gaussian Process Inference on Massive Geostatistical Data

**Mohamed A. Abba**[1] **Brian J. Reich**[1] **Reetam Majumder**[2*] **Brandon Feng**[1]

[1]Department of Statistics, North Carolina State University

[2]Department of Mathematical Sciences, University of Arkansas

*reetamm@uark.edu

## Abstract

Gaussian processes (GPs) are the workhorses of spatial data analyses, but are difficult to scale to large spatial datasets. The Vecchia approximation induces sparsity in the dependence structure and is one of several methods proposed to scale GP inference. We develop a stochastic gradient Markov chain Monte Carlo framework for efficient computation in GPs for spatial data. At each step, the algorithm subsamples a minibatch of locations and subsequently updates process parameters through stochastic gradient Riemannian Langevin dynamics (SGRLD) on a Vecchia-approximated GP likelihood. We are able to conduct full Bayesian analysis for GPs with up to 100,000 locations using our spatial SGRLD, and demonstrate its efficacy through numerical studies and an application using ocean temperature data.

## 1 Introduction

Gaussian process (GP) modeling is a powerful statistical and machine learning tool used to tackle a variety of tasks including regression, classification, and optimization. Within spatial statistics, in particular, GPs have become the primary tool for inference [11], with their main advantage being the ability to provide predictions at unobserved locations along with uncertainty quantification. However, handling large datasets with GPs poses computational challenges due to the cubic time complexity and quadratic memory requirements to evaluate the joint likelihood. This is compounded in the course of Bayesian inference, where thousands of Markov chain Monte Carlo (MCMC) iterations are needed to accurately approximate the posterior distribution. Scalable computation for GPs is therefore necessary for inference on large spatial datasets.

Stochastic gradient (SG) based optimization, where gradient information is used to sample the posterior efficiently, has emerged as an attractive alternative to regular MCMC for scalable computation. Instead of computing a costly gradient based on the full dataset, SG methods only need an unbiased and possibly noisy estimate using a subsample of the data. Although SGMCMC is widely used for *iid* data [24, 30, 6, 21, 9, 5], a naive application in the correlated setting would overlook critical dependencies in the data during subsampling. Moreover, the gradient estimate from the subsamples are not guaranteed to be unbiased. SGMCMC has been used for certain classes of dependent data [20, 22, 2, 3, 7], but to the best of our knowledge, subsampling methods for spatial data that result in unbiased gradient estimates have not been explored. In this work, we develop an SGMCMC algorithm based on Langevin dynamics (SGLD) for large spatial datasets, assumed to have a Matérn correlation structure [27]. We extend the SGLD method to the case of non-*iid* data using the Vecchia approximation that substantially reduces the computational cost to provide a method that takes account of the local curvature to improve convergence.

Workshop on Bayesian Decision-making and Uncertainty, 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

# 2 Methodology

## 2.1 The Matérn GP and the Vecchia approximation

Let $Y_i$ for $i \in \{1, ..., n\}$ be the observation at a spatial location $\mathbf{s}_i = (s_{i1}, s_{i2})$, and let $\mathbf{X}_i = (X_{i1}, ..., X_{ip})$ be a corresponding vector of covariates. The data-generation model for GP regression in the case of Gaussian data is

$$Y_i = \mathbf{X}_i\boldsymbol{\beta} + Z_i + \varepsilon_i, \tag{1}$$

with covariate effects $\boldsymbol{\beta}$, spatial process $Z_i \equiv Z(\mathbf{s}_i)$, and measurement error $\varepsilon_i \overset{iid}{\sim} \text{Normal}(0, \tau^2)$ with a nugget $\tau^2$. The process $Z(\mathbf{s})$ is an isotropic spatial Gaussian process with mean $\text{E}\{Z(\mathbf{s})\} = 0$, spatial variance $\text{Var}\{Z(\mathbf{s})\} = \sigma^2$ and spatial correlation $\text{Cor}\{Z_i, Z_j\} = \text{K}(d_{ij})$ for distance $d_{ij} = ||\mathbf{s}_i - \mathbf{s}_j||$. Specifically, we assume that the process has a Matérn correlation function [27] with range $\rho$ and smoothness $\nu$:

$$\text{K}(d) = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{d}{\rho}\right)^{\nu} \mathcal{K}_{\nu}\left(\frac{d}{\rho}\right), \tag{2}$$

where $\mathcal{K}_\nu$ is the modified Bessel function of the second kind. Let $\boldsymbol{\theta} = (\sigma^2, \rho, \nu, \tau^2)$ be the collection of covariance parameters. The marginal distribution (over $Z$) of $\mathbf{Y} = \{Y_1, \ldots, Y_n\}$ is multivariate normal with mean $\mathbb{E}[\mathbf{Y}] = \mathbf{X}\boldsymbol{\beta}$, for $\mathbf{X} \in \mathbf{R}^{n\times p}$ covariate matrix with the $\text{i}^{\text{th}}$ row $\mathbf{X}_i$, and covariance matrix $\mathbb{E}[(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^{\text{T}} \mid \boldsymbol{\theta}] = \Sigma(\boldsymbol{\theta})$ with

$$\begin{aligned} \Sigma(\boldsymbol{\theta}) &= \sigma^2\mathbf{K} + \tau^2\mathbf{I}_n, \\ \mathbf{K}_{i,j} &= \text{K}(d_{ij}). \end{aligned} \tag{3}$$

The full log-likelihood for the process is given by:

$$\ell_{\text{full}}(\boldsymbol{\beta}, \boldsymbol{\theta}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log\det\Sigma(\boldsymbol{\theta}) - \frac{1}{2}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^{\text{T}}\Sigma(\boldsymbol{\theta})^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}). \tag{4}$$

Evaluating the full likelihood of the process involves computing the determinant and inverse of $\Sigma(\boldsymbol{\theta})$ which generally requires $O(n^3)$ operations, and becomes prohibitive for large spatial datasets. To alleviate this, we write the joint distribution of $\mathbf{Y}$ as a product of univariate conditional distributions, which can then be approximated by a Vecchia approximation [29, 28, 8, 16]:

$$f(Y_1, ..., Y_n) = \prod_{i=1}^{n} f(Y_i|Y_1, ..., Y_{i-1}) \approx \prod_{i=1}^{n} f_i(Y_i|Y_{(i)}), \tag{5}$$

for $Y_{(i)} = \{Y_j; j \in \mathcal{N}_i\}$ and conditioning set $\mathcal{N}_i \subseteq \{1, ..., i-1\}$, e.g., the indices of the $m_i \leq m$ locations in $\mathcal{N}_i$ that are closest to $\mathbf{s}_i$ according to some ordering of the data. Conditioning on $\mathcal{N}_i$ leads to substantial computational savings when $m$ is small, *i.e.*, $m << n$. Let $p(\boldsymbol{\beta}, \boldsymbol{\theta})$ be the prior distribution on the regression and covariance parameters. Using (5) we can write the posterior $p(\boldsymbol{\beta}, \boldsymbol{\theta} \mid \mathbf{Y})$ (ignoring a constant that does not depend on the parameters) as:

$$\begin{aligned} \ell(\boldsymbol{\beta}, \boldsymbol{\theta}) &= \sum_{i=1}^{n} \log f(Y_i \mid Y_{(i)}, \boldsymbol{\beta}, \boldsymbol{\theta}), \\ \log p(\boldsymbol{\beta}, \boldsymbol{\theta} \mid \mathbf{Y}) &= \ell(\boldsymbol{\beta}, \boldsymbol{\theta}) + \log p(\boldsymbol{\beta}, \boldsymbol{\theta}). \end{aligned} \tag{6}$$

The log-likelihood and log-posterior of the parameters $\{\boldsymbol{\beta}, \boldsymbol{\theta}\}$ can consequently be written as a sum of conditional normal log-densities, where the conditioning set is at most of size $m$.

## 2.2 SGLD and SGRLD for spatial data

The Vecchia approximation reduces the computational cost for evaluating the full likelihood and the posterior from $O(n^3)$ to $O(nm^3)$; however, this can still pose challenges for very large $n$. We can further reduce the cost of Bayesian inference by using subsampling strategies. Note that sampling the summands of (6) with equal probability and without replacement leads to an unbiased estimate of the gradient. Let $\mathcal{B} \subset \{1, \ldots, n\}$ be a subsample, *i.e.*, a minibatch index set of size $n_{\mathcal{B}}$, and let

$$\bar{\ell}_{\mathcal{B}}(\boldsymbol{\beta}, \boldsymbol{\theta}) = \frac{n}{n_{\mathcal{B}}} \sum_{i\in\mathcal{B}} \log f(Y_i \mid Y_{(i)}, \boldsymbol{\beta}, \boldsymbol{\theta}). \tag{7}$$

**Theorem 1.** *The gradient of $\bar{\ell}_{\mathcal{B}}$ is an unbiased estimator of the gradient of the Vecchia likelihood $\ell(\boldsymbol{\beta}, \boldsymbol{\theta})$.*

*Proof.*

$$\begin{aligned}
\mathbb{E}_{\mathcal{B}}[\nabla \bar{\ell}_{\mathcal{B}}(\boldsymbol{\beta}, \boldsymbol{\theta})] &= \nabla\, \mathbb{E}_{\mathcal{B}}\left[\frac{n}{n_{\mathcal{B}}} \sum_{i=1}^{n} \log f(Y_i \mid Y_{(i)}, \boldsymbol{\beta}, \boldsymbol{\theta}) \delta_{i \in \mathcal{B}}\right] \\
&= \nabla \sum_{i=1}^{n} \log f(Y_i \mid Y_{(i)}, \boldsymbol{\beta}, \boldsymbol{\theta}) \\
&= \nabla \ell(\boldsymbol{\beta}, \boldsymbol{\theta}). \qquad (8)
\end{aligned}$$

□

Using (8), we can construct an unbiased estimate of the gradient of the Vecchia log-posterior based on a minibatch of the data:

$$\bar{g}_{\mathcal{B}}(\boldsymbol{\beta}, \boldsymbol{\theta}) = \nabla \bar{\ell}_{\mathcal{B}}(\boldsymbol{\beta}, \boldsymbol{\theta}) + \nabla \log p(\boldsymbol{\beta}, \boldsymbol{\theta}), \qquad (9)$$

reducing the cost of learning iterations to be linear in $n_{\mathcal{B}}$ instead of $n$, *i.e.*, $O(m^3 n_{\mathcal{B}})$.

SGMCMC proceeds by simulating continuous dynamics of a potential energy, namely the negative log-posterior, $-\log p(\boldsymbol{\beta}, \boldsymbol{\theta} \mid \mathbf{Y})$, in a manner that generates samples from the posterior distribution. Let $\boldsymbol{\phi} = (\boldsymbol{\beta}^{\mathrm{T}}, \boldsymbol{\theta}^{\mathrm{T}})^{\mathrm{T}}$ be the vector of all parameters for the GP regression model. The Langevin diffusion over $\log p(\boldsymbol{\phi} \mid \mathbf{Y})$ is given by the stochastic differential equation

$$d(\boldsymbol{\phi}_t) = \nabla \log p(\boldsymbol{\phi}_t \mid \mathbf{Y}) dt + \sqrt{2} dW_t, \qquad (10)$$

where $dW_t$ is Brownian motion and the index $t$ represents time. The distribution of samples $\boldsymbol{\phi}_t$ converges to the true posterior as $t \to \infty$ [26]. Since simulating a continuous time process is infeasible in practice, we use the Euler discretization method to approximate the Langevin dynamics:

$$\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + h_t \nabla \log p(\boldsymbol{\phi}_t \mid \mathbf{Y}) + \sqrt{2h_t} e_t, \qquad (11)$$

where $h_t$ is the step size at time $t$, $\boldsymbol{\phi}_t$ the current value of the parameter, and $e_t$ is random white noise. This recursive sampling approach is known as the Langevin Monte Carlo algorithm. Often, a Metropolis-Hastings (MH) correction step is added to account for the discretization error.

Computing the gradient of the log-posterior for large $n$ represents a computational bottleneck. To overcome this problem, the key idea of stochastic gradient Langevin dynamics (SGLD) is to replace $\nabla \log p(\boldsymbol{\phi} \mid \mathbf{Y})$ with an unbiased gradient estimate, *i.e.*, $\bar{g}_{\mathcal{B}}(\boldsymbol{\phi})$ in (9), that is computationally cheaper to compute and uses a decreasing step size $h_t$ to avoid the costly MH correction steps,

$$\text{SGLD}: \qquad \boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + h_t \bar{g}_{\mathcal{B}}(\boldsymbol{\phi}_t) + \sqrt{2h_t} e_t, \qquad (12)$$

for positive step sizes that satisfy the Robbins-Monro conditions [25]. Note that (12) updates all parameters using the same step size, which can cause slow mixing when different parameters have different curvature or scales. Stochastic gradient Reimannian Langevin dynamics (SGRLD) accounts for differences in curvature and scale by using an appropriate Riemannian metric $G(\boldsymbol{\phi})$ and preconditioning the unbiased gradient and noise in (12) using $G^{-1}(\boldsymbol{\phi})$. Commonly used metrics for $G(\boldsymbol{\phi})$ include the Fisher information matrix and estimates of the Hessian of the log-posterior. Given a preconditioning matrix $G(\boldsymbol{\phi})$, the SGRLD step is

$$\text{SGRLD}: \qquad \boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t + h_t \left(G^{-1}(\boldsymbol{\phi}_t) \bar{g}_{\mathcal{B}}(\boldsymbol{\phi}_t) + \Gamma(\boldsymbol{\phi}_t)\right) + \sqrt{2h_t} G^{-1/2}(\boldsymbol{\phi}_t) e_t, \qquad (13)$$

where the term $\Gamma(\boldsymbol{\phi}_t)$ represents the drift term that describes how the preconditioner $G(\boldsymbol{\phi}_t)$ changes with respect to $\boldsymbol{\phi}_t$. The drift term is given by

$$\Gamma(\boldsymbol{\phi}_t)_i = \sum_j \frac{\partial G(\boldsymbol{\phi}_t)^{-1}_{ij}}{\partial \boldsymbol{\phi}_{tj}}. \qquad (14)$$

The drift term vanishes in the SGLD step since the preconditioner is assumed to be the identity matrix. The SGRLD algorithm in (13) takes steps in the steepest ascent on the manifold defined by the metric $G(\boldsymbol{\phi}_t)$. While the Fisher information matrix is often intractable, our use of the Vecchia approximation facilitates computation of the Fisher information and its inverse without incurring a high computational cost; derivations of the expressions are provided in Appendix A. Code for our approach is available in the form of an R package on GitHub [1].

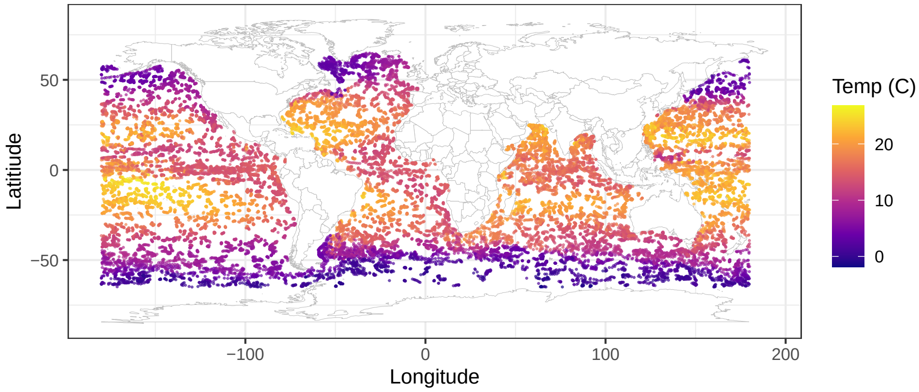

Figure 1: Argo ocean temperature measurements at a depth of 100 meters.

| | MSE | Coverage | $R^2$ | Time (in minutes) |
|---|---|---|---|---|
| NNGP | 6.41 | 0.88 | 0.89 | 218.55 |
| SGRLD | 1.47 | 0.93 | 0.94 | 7.01 |

Table 1: Prediction MSE, squared correlation between predicted and observed ($R^2$) and coverage rate of the $95\%$ predictive credible intervals on the test set and the correlation between the predicted temperatures and true observed values. The last column gives the total training time in minutes. We take $8000$ and $40000$ samples using the NNGP and SGRLD method respectively.

## 3 Results

We tested the efficacy of our proposed SGRLD method in (13) using a numerical study and assessed its performance against four state-of-the-art Bayesian methods. The first three are SG methods with adaptive drifts. The last method is the nearest neighbor Gaussian process (NNGP) [8] that uses the full dataset to sample the posterior distribution using the Vecchia approximation. The methods were compared for datasets with $n = \left\{10^4, 10^5, 10^6\right\}$ locations. Study details are provided in the Appendix B.1. SGRLD outperformed the competing methods with very low MSE across parameters. Additionally, all SG methods outperformed NNGP. SGRLD had consistently high coverage for $95\%$ credible intervals and overall the highest expected sample size per minute (ESS/min) among all methods.

We also applied the proposed method to the ocean temperature data provided by the Argo Program [4] made available through the `GpGp` package [13]in `R`. Each of the $n = 32,436$ observations are taken on buoys in the spring of 2016, and measures ocean temperature (C) at depths of roughly 100, 150, and 200 meters. The data are plotted in Figure 1 for a depth of 100 meters. As an illustrative example, the mean function is taken to be quadratic in latitude and longitude. All prior distributions and MCMC settings are the same as in the numerical study in Section B.1.

We set aside $20\%$ of all observations as the testing set, and train the models using $8000$ and $40000$ MCMC iterations for the NNGP and SGRLD methods respectively. Table 1 gives the MSE and coverage rate on the testing set, and total training time respectively. SGRLD results in less than a quarter of the MSE of NNGP while also requiring less than a twentieth of the time. For the coverage of the $95\%$ prediction intervals, the NNGP method's average coverage on the testing set is significantly lower than the nominal value, while our proposed method achieves $93\%$ coverage.

Table 2 gives the posterior mean, $95\%$ interval and the effective sample size [(ESS), 14] per minute for the covariance parameters for SGRLD and NNGP. The posterior means and credible intervals for $\rho$, and to a lesser extent $\sigma^2$, vary substantially across methods. The range estimates from SGRLD are almost three orders of magnitude higher than the NNGP estimate. Given the prediction results in Table 1, this indicates that the NNGP is underestimating $\rho$. Furthermore, for NNGP, the credible interval for $\rho$ has a total width of $10^{-2}$, perhaps indicating poor convergence. We also see from Table 2 that our SGRLD method allows fast exploration of the posterior and leads to massively higher ESS

| Method | Parameter | Posterior mean | 95% CI | ESS/min |
|---|---|---|---|---|
| NNGP | $\sigma^2$ | 6.72 | $(6.32, 7.08)$ | 0.17 |
| | $\rho$ | 0.10 | $(0.10, 0.11)$ | 3.42 |
| | $\nu$ | 0.33 | $(0.32, 0.34)$ | 0.04 |
| | $\tau^2$ | 0.08 | $(0.08, 0.09)$ | 0.08 |
| SGRLD | $\sigma^2$ | 10.64 | $(7.41, 13.57)$ | 52.21 |
| | $\rho$ | 48.93 | $(22.94, 68.46)$ | 115.41 |
| | $\nu$ | 0.25 | $(0.23, 0.27)$ | 18.68 |
| | $\tau^2$ | 0.04 | $(0.03, 0.05)$ | 39.13 |

Table 2: Posterior mean, $95\%$ credible intervals and ESS per minute for all covariance parameters.

per minute, while giving reasonable convergence (Figure 2). Additional plots and results, including a sensitivity study for the hyperparameters $m$ and $n_{\mathcal{B}}$, are provided in the Appendix B.2.

## 4 Discussion

SG methods offer considerable speed-ups when the data size is very large. This enables fast exploration of the posterior in significantly less time. GPs however fall within the correlated setting case where SGMCMC methods have received limited attention. Spatial correlation is a critical component of GPs and naive subsampling during parameter estimation would lead to random divisions of the spatial domain at each iteration. By leveraging the form of the Vecchia approximation, we derive unbiased gradient estimates based on minibatches of the data. We developed a new stochastic gradient based MCMC algorithm for scalable Bayesian inference in large spatial data settings. Without the Vecchia approximation, subsampling strategies would always lead to biased gradient estimates. The proposed method also uses the exact Fisher information to speed up convergence and explore the parameter space efficiently. Our work contributes to the literature on scalable methods for Gaussian process, and can be extended to non Gaussian models, e.g., to classification problems.

## Acknowledgements

This research was partially supported by National Science Foundation grants DMS2152887 and DMR-2022254, and by grants from the Southeast National Synthesis Wildfire and the United States Geological Survey's National Climate Adaptation Science Center (G21AC10045).

## References

[1] M. A. Abba, R. Majumder, B. Feng, and B. J. Reich. *spSGMCMC: spatial Stochastic Gradient MCMC*, 2024. R package version 0.1.0.

[2] C. Aicher, Y.-A. Ma, N. J. Foti, and E. B. Fox. Stochastic gradient MCMC for state space models. *SIAM Journal on Mathematics of Data Science*, 1(3):555–587, 2019.

[3] C. Aicher, S. Putcha, C. Nemeth, P. Fearnhead, and E. B. Fox. Stochastic gradient MCMC for nonlinear state space models. *arXiv preprint arXiv:1901.10568*, 2021.

[4] Argo. Argo Program Office. `https://argo.ucsd.edu/`, 2023. Accessed: 2023-11-26.

[5] J. Baker, P. Fearnhead, E. B. Fox, and C. Nemeth. Control variates for stochastic gradient MCMC. *Statistics and Computing*, 29:599–615, 2019.

[6] C. Chen, N. Ding, and L. Carin. On the convergence of stochastic gradient MCMC algorithms with high-order integrators. In *Neural Information Processing Systems*, 2015.

[7] H. Chen, L. Zheng, R. Al Kontar, and G. Raskutti. Stochastic gradient descent in correlated settings: A study on gaussian processes. *Advances in neural information processing systems*, 33:2722–2733, 2020.

[8] A. Datta, S. Banerjee, A. O. Finley, and A. E. Gelfand. Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association*, 111:800–812, 2016.

[9] K. A. Dubey, S. J Reddi, S. A. Williamson, B. Poczos, A. J. Smola, and E. P. Xing. Variance reduction in stochastic gradient Langevin dynamics. *Advances in neural information processing systems*, 29, 2016.

[10] A. O. Finley, A. Datta, and S. Banerjee. spNNGP R package for nearest neighbor Gaussian process models. *Journal of Statistical Software*, 103(5):1–40, 2022.

[11] A. E. Gelfand and E. M. Schliep. Spatial statistics and Gaussian processes: A beautiful marriage. *Spatial Statistics*, 18:86–104, 2016. Spatial Statistics Avignon: Emerging Patterns.

[12] J. Guinness. Gaussian process learning via fisher scoring of vecchia's approximation, 2019.

[13] J. Guinness, M. Katzfuss, and Y. Fahmy. Gpgp: fast Gaussian process computation using Vecchia's approximation. *R package version 0.1. 0*, 2018.

[14] P. Heidelberger and P. D. Welch. A spectral method for confidence interval generation and run length control in simulations. *Communications of the ACM*, 24(4):233–245, 1981.

[15] G. Hinton, N. Srivastava, and K. Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.

[16] M. Katzfuss and J. Guinness. A general framework for Vecchia approximations of Gaussian processes. *Statistical Science*, 36(1):124–141, 2021.

[17] S. Kim, Q. Song, and F. Liang. Stochastic gradient Langevin dynamics with adaptive drifts. *Journal of statistical computation and simulation*, 92(2):318–336, 2022.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] C. Li, C. Chen, D. Carlson, and L. Carin. Preconditioned stochastic gradient Langevin dynamics for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[20] W. Li, S. Ahn, and M. Welling. Scalable MCMC for mixed membership stochastic blockmodels. In *Artificial Intelligence and Statistics*, pages 723–731. PMLR, 2016.

[21] Y. Ma, Y.-A. Ma, T. Chen, and E. B. Fox. A complete recipe for stochastic gradient MCMC. In *Neural Information Processing Systems*, 2015.

[22] Y.-A. Ma, N. J. Foti, and E. B. Fox. Stochastic gradient MCMC methods for hidden Markov models. In *International Conference on Machine Learning*, pages 2265–2274. PMLR, 2017.

[23] K. V. Mardia and R. J. Marshall. Maximum likelihood estimation of models for residual covariance in spatial regression. *Biometrika*, 71(1):135–146, 1984.

[24] C. Nemeth and P. Fearnhead. Stochastic gradient Markov chain Monte Carlo. *Journal of the American Statistical Association*, 116(533):433–450, 2021.

[25] H. Robbins and S. Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951.

[26] G. O. Roberts and J. S. Rosenthal. Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60, 1998.

[27] M. L. Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 1999.

[28] M. L. Stein, Z. Chi, and L. J. Welty. Approximating likelihoods for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(2):275–296, 2004.

[29] A. V. Vecchia. Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):297–312, 1988.

[30] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning*, 2011.

# A Derivation of Gradients and Fisher Information for SGRLD

Given an index set for a mini-batch subset of the data $\mathcal{B}$, the log-likelihood in (7) decomposes as the sum of log-conditional densities of the $Y_i$ given the conditioning points $Y_{(i)}$. Computing the gradient of these conditional densities is analytically complicated and not computationally tractable. We follow [12] to first rewrite the log-conditional densities in terms of marginal densities, and then compute the gradients and Fisher information. Let $u_i = Y_{(i)}$, the set of neighbours, and $v_i = (Y_{(i)}, Y_i)$, the vector of concatenating the $i^{\text{th}}$ observation and its neighbours. Let $\mathbf{Q}_i$ and $\mathbf{R}_i$ be the covariate matrices for $u_i$ and $v_i$ respectively, and let $\mathbf{A}_i$ and $\mathbf{B}_i$ denote the covariance matrices of $u_i$ and $v_i$. The minibatch log-likelihood in (7) can thus be written as

$$\begin{aligned}\bar{\ell}_{\mathcal{B}}(\boldsymbol{\phi}) &= \sum_{i\in\mathcal{B}} \log f(v_i \mid \boldsymbol{\phi}) - \log f(u_i \mid \boldsymbol{\phi}) \\ &= -\frac{1}{2}\sum_{i\in\mathcal{B}} \log\det \mathbf{B}_i - \log\det \mathbf{A}_i \qquad (15) \\ &\quad -\frac{1}{2}\sum_{i\in\mathcal{B}} [(v_i - \mathbf{R}_i\boldsymbol{\beta})^{\mathrm{T}}\mathbf{B}_i^{-1}(v_i - \mathbf{R}_i\boldsymbol{\beta}) - (u_i - \mathbf{Q}_i\boldsymbol{\beta})^{\mathrm{T}}\mathbf{A}_i^{-1}(u_i - \mathbf{Q}_i\boldsymbol{\beta})] - \frac{n_{\mathcal{B}}}{2}\log(2\pi).\end{aligned}$$

In order to compute the log-likelihood, we need the following quantities

$$p^1_{\mathcal{B}}(\boldsymbol{\theta}) = \sum_{i\in\mathcal{B}} \log\det \mathbf{B}_i - \log\det \mathbf{A}_i \qquad (16)$$

$$p^2_{\mathcal{B}}(\boldsymbol{\theta}) = \sum_{i\in\mathcal{B}} (v_i^{\mathrm{T}}\mathbf{B}_i^{-1}v_i - u_i^{\mathrm{T}}\mathbf{A}_i^{-1}u_i) \qquad (17)$$

$$p^3_{\mathcal{B}}(\boldsymbol{\theta}) = \sum_{i\in\mathcal{B}} (\mathbf{R}_i^{\mathrm{T}}\mathbf{B}_i^{-1}v_i - \mathbf{Q}_i^{\mathrm{T}}\mathbf{A}_i^{-1}u_i) \qquad (18)$$

$$p^4_{\mathcal{B}}(\boldsymbol{\theta}) = \sum_{i\in\mathcal{B}} (\mathbf{R}_i^{\mathrm{T}}\mathbf{B}_i^{-1}\mathbf{R}_i - \mathbf{Q}_i^{\mathrm{T}}\mathbf{A}_i^{-1}\mathbf{Q}_i). \qquad (19)$$

The quantities in (16) – (19) only depend on the covariance parameters $\boldsymbol{\theta}$ via $\mathbf{A}_i$ and $\mathbf{B}_i$ and not the mean parameters $\boldsymbol{\beta}$. We can now write the minibatch log-likelihood as

$$\bar{\ell}_{\mathcal{B}}(\boldsymbol{\phi}) = -\frac{n_{\mathcal{B}}}{2}\log(2\pi) - \frac{1}{2}\left[p^1_{\mathcal{B}}(\boldsymbol{\theta}) + p^2_{\mathcal{B}}(\boldsymbol{\theta}) - 2\boldsymbol{\beta}^{\mathrm{T}}p^3_{\mathcal{B}}(\boldsymbol{\theta}) + \boldsymbol{\beta}^{\mathrm{T}}p^4_{\mathcal{B}}(\boldsymbol{\theta})\boldsymbol{\beta}\right]. \qquad (20)$$

## A.1 Mean parameters

The gradient of the minibatch log-likelihood with respect to the mean parameters $\boldsymbol{\beta}$ is

$$\frac{\partial\bar{\ell}_{\mathcal{B}}(\boldsymbol{\beta},\boldsymbol{\theta})}{\partial\boldsymbol{\beta}} = p^3_{\mathcal{B}}(\boldsymbol{\theta}) - p^4_{\mathcal{B}}(\boldsymbol{\theta})\boldsymbol{\beta}. \qquad (21)$$

For the Fisher information, recall that if a random vector follows a multivariate normal model with mean and variance parameterized by two different parameter vectors, *i.e.*, $W \sim \mathbf{N}(\mu(\boldsymbol{\beta}), \Sigma(\boldsymbol{\theta}))$, then the Fisher information is block diagonal $\mathcal{I}(\boldsymbol{\phi}) = \operatorname{diag}(\mathcal{I}(\boldsymbol{\beta}), \mathcal{I}(\boldsymbol{\theta}))$. Furthermore, let $J_{\boldsymbol{\beta}}$ be the Jacobian of $\mu(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$. Then the Fisher information matrix is analytically available [23] and takes the form

$$\mathcal{I}(\boldsymbol{\beta}) = J_{\boldsymbol{\beta}}\Sigma^{-1}J_{\boldsymbol{\beta}}^{\mathrm{T}} \qquad (22)$$

$$\mathcal{I}(\boldsymbol{\theta})_{jk} = \frac{1}{2}\operatorname{Tr}\left(\Sigma^{-1}\frac{\partial\Sigma}{\partial\boldsymbol{\theta}_j}\Sigma^{-1}\frac{\partial\Sigma}{\partial\boldsymbol{\theta}_k}\right). \qquad (23)$$

Using (22) and the chain rule property of the Fisher information, $\mathcal{I}_{Y(s_i)|u_i}(\boldsymbol{\phi}) = \mathcal{I}_{v_i}(\boldsymbol{\phi}) - \mathcal{I}_{u_i}(\boldsymbol{\phi})$, and summing over the components of the log-likelihood, we get

$$\mathcal{I}_{\mathcal{B}}(\boldsymbol{\beta}) = \sum_{i\in\mathcal{B}} (\mathbf{R}_i^{\mathrm{T}}\mathbf{B}_i^{-1}\mathbf{R}_i - \mathbf{Q}_i^{\mathrm{T}}\mathbf{A}_i^{-1}\mathbf{Q}_i) = p^4_{\mathcal{B}}(\boldsymbol{\theta}). \qquad (24)$$

Hence the Fisher information of $\boldsymbol{\beta}$ is constant with respect to the mean parameters. In addition, since $\mathcal{I}(\boldsymbol{\phi})$ is block diagonal, the drift term which represents how $\mathcal{I}(\boldsymbol{\beta})$ changes with respect to $\boldsymbol{\phi}$ is $\Gamma_{\mathcal{B}}(\boldsymbol{\beta}) = \mathbf{0}_p$. The SGRLD step for regression parameters is thus

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t + h_t p^4_{\mathcal{B}}(\boldsymbol{\theta}_t)^{-1}\left(p^3_{\mathcal{B}}(\boldsymbol{\theta}_t) - p^4_{\mathcal{B}}(\boldsymbol{\theta}_t)\boldsymbol{\beta}_t\right) + \sqrt{2h_t}p^4_{\mathcal{B}}(\boldsymbol{\theta})^{-1/2}e_t. \qquad (25)$$

### A.2 Covariance parameters

For the covariance parameters, we first start by computing the partial derivatives of the quantities defined in (16) – (19) with respect to the components of $\boldsymbol{\theta}$, $p_j^k(\boldsymbol{\theta}) = \partial p_{\mathcal{B}}^k(\boldsymbol{\theta})/\partial\theta_j$ for $j \in \{1, \ldots, 4\}$,

$$p_j^1(\boldsymbol{\theta}) = \sum_{i\in\mathcal{B}} \left( \mathrm{Tr}(\mathbf{B}_i^{-1}\frac{\partial \mathbf{B}_i}{\partial\theta_j}) - \mathrm{Tr}(\mathbf{A}_i^{-1}\frac{\partial \mathbf{A}_i}{\partial\theta_j}) \right) \tag{26}$$

$$p_j^2(\boldsymbol{\theta}) = \sum_{i\in\mathcal{B}} \left( v_i^{\mathrm{T}}\mathbf{B}_i^{-1}\frac{\partial \mathbf{B}_i}{\partial\theta_j}\mathbf{B}_i^{-1}v_i - u_i^{\mathrm{T}}\mathbf{A}_i^{-1}\frac{\partial \mathbf{A}_i}{\partial\theta_j}\mathbf{A}_i^{-1}u_i \right) \tag{27}$$

$$p_j^3(\boldsymbol{\theta}) = \sum_{i\in\mathcal{B}} \left( \mathbf{R}_i^{\mathrm{T}}\mathbf{B}_i^{-1}\frac{\partial \mathbf{B}_i}{\partial\theta_j}\mathbf{B}_i^{-1}v_i - \mathbf{Q}_i^{\mathrm{T}}\mathbf{A}_i^{-1}\frac{\partial \mathbf{A}_i}{\partial\theta_j}\mathbf{A}_i^{-1}u_i \right) \tag{28}$$

$$p_j^4(\boldsymbol{\theta}) = \sum_{i\in\mathcal{B}} \left( \mathbf{R}_i^{\mathrm{T}}\mathbf{B}_i^{-1}\frac{\partial \mathbf{B}_i}{\partial\theta_j}\mathbf{B}_i^{-1}\mathbf{R}_i - \mathbf{Q}_i^{\mathrm{T}}\mathbf{A}_i^{-1}\frac{\partial \mathbf{A}_i}{\partial\theta_j}\mathbf{A}_i^{-1}\mathbf{Q}_i \right) \tag{29}$$

$$\frac{\partial \bar{\ell}_{\mathcal{B}}(\boldsymbol{\beta}, \boldsymbol{\theta})}{\partial\theta_j} = -\frac{1}{2}\left[ p_j^1(\boldsymbol{\theta}) + p_j^2(\boldsymbol{\theta}) - 2p_j^3(\boldsymbol{\theta})\boldsymbol{\beta} + \boldsymbol{\beta}^{\mathrm{T}} p_j^4(\boldsymbol{\theta})\boldsymbol{\beta} \right]. \tag{30}$$

Using (23) and the chain rule decomposition of the Fisher information, we derive the analytic form of the Fisher information and drift term for the covariance parameters

$$\mathcal{I}_{\mathcal{B}}(\boldsymbol{\theta})_{jk} = \frac{1}{2}\sum_{i\in\mathcal{B}} \mathrm{Tr}\left( \mathbf{B}_i^{-1}\frac{\partial \mathbf{B}_i}{\partial\theta_j}\mathbf{B}_i^{-1}\frac{\partial \mathbf{B}_i}{\partial\theta_k} \right) - \mathrm{Tr}\left( \mathbf{A}_i^{-1}\frac{\partial \mathbf{A}_i}{\partial\theta_j}\mathbf{A}_i^{-1}\frac{\partial \mathbf{A}_i}{\partial\theta_k} \right) \tag{31}$$

$$\begin{aligned} \frac{\partial \mathcal{I}_{\mathcal{B}}(\boldsymbol{\theta})_{jk}}{\partial\theta_k} =& \sum_{i\in\mathcal{B}} \mathrm{Tr}\left( \mathbf{B}_i^{-1}\frac{\partial^2 \mathbf{B}_i}{\partial\theta_j\partial\theta_k}\mathbf{B}_i^{-1}\frac{\partial \mathbf{B}_i}{\partial\theta_k} \right) - \mathrm{Tr}\left( \mathbf{B}_i^{-1}\frac{\partial \mathbf{B}_i}{\partial\theta_j}\mathbf{B}_i^{-1}\frac{\partial \mathbf{B}_i}{\partial\theta_k}\mathbf{B}_i^{-1}\frac{\partial \mathbf{B}_i}{\partial\theta_k} \right) \\ & - \sum_{i\in\mathcal{B}} \mathrm{Tr}\left( \mathbf{A}_i^{-1}\frac{\partial^2 \mathbf{A}_i}{\partial\theta_j\partial\theta_k}\mathbf{A}_i^{-1}\frac{\partial \mathbf{A}_i}{\partial\theta_k} \right) - \mathrm{Tr}\left( \mathbf{A}_i^{-1}\frac{\partial \mathbf{A}_i}{\partial\theta_j}\mathbf{A}_i^{-1}\frac{\partial \mathbf{A}_i}{\partial\theta_k}\mathbf{A}_i^{-1}\frac{\partial \mathbf{A}_i}{\partial\theta_k} \right) \end{aligned} \tag{32}$$

$$\Gamma_{\mathcal{B}}(\boldsymbol{\theta})_j = -\sum_k \mathcal{I}_{\mathcal{B}}(\boldsymbol{\theta})^{-1}_{j\cdot}\frac{\partial \mathcal{I}_{\mathcal{B}}(\boldsymbol{\theta})}{\partial\theta_k}\mathcal{I}_{\mathcal{B}}(\boldsymbol{\theta})^{-1}_{\cdot k}. \tag{33}$$

## B Additional Numerical Results

### B.1 Numerical study

In this section, we test our proposed SGRLD method in (13) on synthetic data and assess its performance against state-of-the-art Bayesian methods. We use Mean Squared Error (MSE) and coverage of credible intervals of posterior MCMC estimators to evaluate the estimation of the spatial covariance parameters, and we use the effective sample sizes (ESS) per minute to gauge computational efficiency of MCMC algorithms. We present results only for the spatial covariance parameters $\boldsymbol{\theta}$ since the results are similar across methods for $\boldsymbol{\beta}$.

### B.1.1 Data generation

We generate data on a regular rectangular grid formed with $n_1$ locations on the x-axis and $n_2$ on the y-axis, with a total number of points $n = n_1 n_2$ and grid spacing one. We consider $n = \left\{10^4, 10^5, 10^6\right\}$ for $n_1 = \{100, 300, 1000\}$ and $n_2 = n/n_1$. We generate the Gaussian process $Z(\mathbf{s})$ from a Matérn kernel with possible smoothness values $\nu \in \{0.5, 1.0, 1.5\}$. The range parameter $\rho$ is chosen such that the correlation function is approximately $10^{-4}$ for the maximum distance between two points in the grid. We fix the spatial variance $\sigma^2 = 5$, and consider different scenarios for the observation noise based on the proportion of variance $\kappa = \tau^2/\sigma^2 \in \{0.2, 1.0, 5.0\}$. Let $\mathbf{X}_i = (1, x_i)$, the covariate for the $i^{\text{th}}$ site; the mean of the Gaussian process will take the form $\mathbb{E}[Y_i] = \beta_0 + \beta_1 \cos(x_i)$, where $\beta_0 = -3$, and $\beta_1 = 5$, and $x_i \overset{iid}{\sim}$ Uniform$(-3, 3)$. For $n = 10^6$, generating a Gaussian process is computationally infeasible, thus we generate a Vecchia approximated Gaussian process with $m = 120$ neighbors for each site. For each $n$, we generate 100 datasets and record the posterior mean and posterior credible intervals for parameters.

### B.1.2 Competing methods and metrics

We compare our SGRLD method with four different MCMC methods. The first three are SGMCMC methods that all use momentum and past gradient information to estimate the curvature and accelerate the convergence. These methods extend the momentum methods used in SG optimization methods for faster exploration of the posterior. The first method is preconditioned SGLD [(PSGLD), 19] that uses the Root Mean Square Propagation [(RMSPROP), 15] algorithm to estimate a diagonal preconditioner for the minibatch gradient and injected noise. The second method is ADAMSGLD [17] that extends the widely used Adam (adaptive moment) optimizer [18] to the SGLD setting. ADAMSGLD approximates the first-order and second-order moments of the minibatch gradients to construct a preconditioner. Finally, we also include the performance of momentum SGLD (MSGLD) where no preconditioner is used but past gradient information is used to accelerate the exploration of the posterior. The details of the above algorithms are included in the Appendix C. Finally, the NNGP method [8] is a standard MCMC method based on the Vecchia approximation and is implemented in the R package spNNGP [10]. For this method, the initial values are set to the true values and the Metropolis-Hastings proposal distribution is chosen adaptively using the default settings.

For the SGMCMC methods, the batch size is set to $250$ when the number of locations is $10^4$ and $500$ for the other two cases. We noticed during our experiments that batch sizes in the order of 200 perform better than smaller size ones, with very similar performance to larger ones. The number of epochs will depend on the size of data, and is chosen such that the total number of iterations is $20,000$, of which a quarter are discarded as burn-in. The learning rate is divided by a factor of $2$ every $5$ epochs, so the final learning rate is set at $1\%$ of the initial value. A first tentative value of the learning rate is set at $1/n$, then reduced until the norm of the first step is less than one. We noticed that the appropriate learning rate for our SGRLD method is within one to two orders of magnitude large than the learning rate for the other SG sampling methods. For all the methods, the size of the conditioning set is fixed at $m = 15$. The conditioning sets were selected using the max-min ordering [16] for $n < 10^6$, and random ordering otherwise. [16] showed that the max-min ordering results in significant improvements over other coordinate based orderings. However, when $n$ is very large, the cost of max-min ordering becomes prohibitive. For the NNGP method, we take 2000 samples when $n < 10^5$ and 1000 otherwise. For all the methods we use a non-informative flat prior on the regression parameters. For the covariance parameters, we set the following priors:

$$\begin{aligned} \rho &\sim \text{Gamma}(9.0, 2.0) \\ \nu &\sim \text{Log-Normal}(1.0, 1.0) \\ \tau^2,\, \sigma^2 &\sim \text{Gamma}(0.1, 0.1) \end{aligned}$$

The prior $90\%$ credible intervals for $\rho$ and $\nu$ are $(2.06, 7.88)$ and $(0.52, 14.08)$ respectively, which represent weakly informative priors.

### B.1.3 Results

Table 3 gives the MSE results. Our SGRLD method outperforms all the others with very low MSE across parameters. In particular, the SGMCMC methods all outperform the NNGP method. In our experiments, we noticed that the NNGP method suffers from very slow mixing due to the MH step necessary for sampling the covariance parameters. In fact, even if we start the NNGP sampling process at true values of the covariance parameters, and reduce the variance of the proposal distribution, the acceptance rate of the MH step stays below $15\%$. None of the SGMCMC methods requires any such step as long as the learning rate is kept small.

Table 4 summarizes the results for the coverage of the $95\%$ credible intervals. Our SGRLD method again outperforms the other methods. One exception is that the PSGLD algorithm surpasses the SGRLD in the coverage of the variance parameter. Across methods, the smoothness parameter consistently has the lowest coverage, followed by the range parameter. Even for $n = 10^6$, MSGLD, ADAMSGLD and NNGP fail to attain attain a $90\%$ coverage rate. Whilst the SGRLD coverage rate for both parameters is higher than $90\%$ even for $n = 10^4$.

For the ESS results in Table 5, the SGRLD method offers superior ESS per minute for all the parameters. The PSGLD and MSGLD method seem to adapt to the curvature of the variance parameter, with PSGLD offering higher effective samples than SGRLD. This suggests that the computed preconditioner in PSGLD adapts mainly to the curvature of the variance term, but fails to measure the

Table 3: Mean squared error (Monte Carlo standard errors) of covariance parameters computed using 100 simulations, each having sample size $n$. The proposed SGRLD method compared with other SGMCMC methods (PSGLD, ADAMSGLD, MSGLD) and the full likelihood NNGP method.

| $n$ | Algorithm | Variance ($\sigma^2$) | Range ($\rho$) | Smoothness ($\nu$) | Nugget ($\tau^2$) |
|---|---|---|---|---|---|
| $10^4$ | PSGLD | $0.074(0.013)$ | $0.039(0.008)$ | $0.103(0.017)$ | $0.002(4 \cdot 10^{-4})$ |
| | ADAMSGLD | $0.075(0.017)$ | $0.036(0.008)$ | $0.129(0.023)$ | $0.002(6 \cdot 10^{-4})$ |
| | MSGLD | $0.066(0.014)$ | $0.034(0.008)$ | $0.108(0.0196)$ | $0.002(6 \cdot 10^{-4})$ |
| | NNGP | $0.414(0.131)$ | $0.095(0.071)$ | $0.162(0.106)$ | $0.093(2.4 \cdot 10^{-2})$ |
| | SGRLD | $0.056(0.016)$ | $0.031(0.006)$ | $0.077(0.013)$ | $0.001(10^{-4})$ |
| $10^5$ | PSGLD | $0.008(0.001)$ | $0.002(0.0003)$ | $0.011(0.0019)$ | $1 \cdot 10^{-4}(2 \cdot 10^{-5})$ |
| | ADAMSGLD | $0.014(0.005)$ | $0.008(0.002)$ | $0.031(0.008)$ | $1 \cdot 10^{-4}(2 \cdot 10^{-4})$ |
| | MSGLD | $0.017(0.001)$ | $0.003(5 \cdot 10^{-4)}$ | $0.019(0.002)$ | $2 \cdot 10^{-4}(4 \cdot 10^{-5})$ |
| | NNGP | $0.116(0.030)$ | $0.024(0.01)$ | $0.118(0.08)$ | $4 \cdot 10^{-2}(0.01)$ |
| | SGRLD | $0.005(8 \cdot 10^{-4})$ | $0.001(1.0 \cdot 10^{-4})$ | $0.008(1.8 \cdot 10^{-3})$ | $10^{-4}(2 \cdot 10-5)$ |
| $10^6$ | PSGLD | $0.003(0.001)$ | $0.003(0.0008)$ | $0.002(0.0014)$ | $3.1 \cdot 10^{-4}(6 \cdot 10^{-5})$ |
| | ADAMSGLD | $0.009(0.002)$ | $0.006(0.002)$ | $0.026(0.007)$ | $2 \cdot 10^{-4}(9 \cdot 10^{-5})$ |
| | MSGLD | $0.011(1.8 \cdot 10^{-3})$ | $0.003(5 \cdot 10^{-4)}$ | $0.019(0.002)$ | $1 \cdot 10^{-5}(3 \cdot 10^{-5})$ |
| | NNGP | $0.078(0.055)$ | $0.016(0.009)$ | $0.126(0.086)$ | $0.08(0.049)$ |
| | SGRLD | $0.002(3 \cdot 10^{-4})$ | $0.001(1 \cdot 10^{-4})$ | $0.004(6.1 \cdot 10^{-3})$ | $0.4 \cdot 10^{-4}(1 \cdot 10-5)$ |

Table 4: Coverage of the $95\%$ credible intervals (Monte Carlo standard errors) for the covariance parameters computed using 100 simulations, each having sample size $n$. The proposed SGRLD method is compared with other SGMCMC methods (PSGLD, ADAMSGLD, MSGLD) and the full likelihood NNGP method.

| $n$ | Algorithm | Variance, $\sigma^2$ | Range, $\rho$ | Smoothness, $\nu$ | Nugget, $\tau^2$ |
|---|---|---|---|---|---|
| $10^4$ | PSGLD | $0.977(0.02)$ | $0.845(0.06)$ | $0.815(0.06)$ | $0.931(0.05)$ |
| | ADAMSGLD | $0.886(0.05)$ | $0.791(0.08)$ | $0.647(0.08)$ | $0.636(0.05)$ |
| | MSGLD | $0.793(0.03)$ | $0.847(0.07)$ | $0.709(0.07)$ | $0.683(0.05)$ |
| | NNGP | $0.783(0.06)$ | $0.776(0.05)$ | $0.614(0.07)$ | $0.812(0.01)$ |
| | SGRLD | $0.955(0.03)$ | $0.924(0.05)$ | $0.909(0.04)$ | $0.935(0.01)$ |
| $10^5$ | PSGLD | $0.991(0.03)$ | $0.913(0.04)$ | $0.862(0.05)$ | $0.965(0.02)$ |
| | ADAMSGLD | $0.861(0.03)$ | $0.754(0.07)$ | $0.814(0.03)$ | $0.738(0.05)$ |
| | MSGLD | $0.896(0.04)$ | $0.881(0.07)$ | $0.774(0.08)$ | $0.872(0.07)$ |
| | NNGP | $0.826(0.05)$ | $0.758(0.04)$ | $0.714(0.03)$ | $0.872(0.02)$ |
| | SGRLD | $0.957(0.01)$ | $0.964(0.01)$ | $0.948(0.01)$ | $0.932(5 \cdot 10^{-3})$ |
| $10^6$ | PSGLD | $0.987(6 \cdot 10^{-3})$ | $0.934(0.02)$ | $0.901(0.03)$ | $0.961(0.01)$ |
| | ADAMSGLD | $0.902(0.01)$ | $0.824(10^{-3})$ | $0.838(0.02)$ | $0.781(0.03)$ |
| | MSGLD | $0.884(10^{-3})$ | $0.918(0.02)$ | $0.846(0.01)$ | $0.926(0.01)$ |
| | NNGP | $0.866(0.03)$ | $0.818(0.06)$ | $0.834(0.04)$ | $0.862(0.01)$ |
| | SGRLD | $0.968(6 \cdot 10^{-3})$ | $0.941(8 \cdot 10^{-3})$ | $0.929(5 \cdot 10^{-3})$ | $0.941(2 \cdot 10^{-3})$ |

curvature of the smoothness and range. A similar behavior is also observed in the other two methods, MSGLD and ADAMSGLD. On the other hand, the ESS for SGRLD is of the same order for all the parameters. We believe this indicates that using the Fisher information matrix as a Riemannian metric provides an accurate measure of the curvature and results in higher effective samples for all the parameters. The NNGP method provides low effective sample sizes compared to the other three methods due to the low acceptance rate from the MH correction step.

### B.2 Additional Argo results

Figure 2 shows trace plots for the posterior sampling of the covariance parameters using SGRLD. We conduct a sensitivity analysis to assess the effect of the mini batch size and conditioning set size on the results. We compare the SGRLD results with mini-batch size $n_{\mathcal{B}} \in \{100, 250, 500\}$ and conditioning set size $m \in \{10, 15, 30\}$. Table 6 show the posterior mean and $95\%$ credible intervals of the covariance parameters for all combinations of the two hyperparameters. The posterior mean of the spatial variance, smoothness and nugget vary little across these combinations of tuning parameters. For the range parameter, we notice a sensitivity to small batch sizes, *e.g.*, $n_{\mathcal{B}} = 100$ resulting in wide credible intervals and larger estimates compared to the other cases. For batch sizes $\{250, 500\}$ the estimates are similar across values of $m$.

Table 5: Effective sample size per minute (Monte Carlo standard errors) of covariance parameters computed using 100 simulations, each having sample size $n$. The proposed SGRLD method is compared with other SGMCMC methods (PSGLD, ADAMSGLD, MSGLD) and the full likelihood NNGP method.

| $n$ | Algorithm | Variance, $\sigma^2$ | Range, $\rho$ | Smoothness, $\nu$ | Nugget, $\tau^2$ |
|---|---|---|---|---|---|
| $10^4$ | PSGLD | 42.97(1.57) | 8.43(0.54) | 4.33(0.26) | 9.82(0.79) |
| | ADAMSGLD | 9.12(0.45) | 4.22(0.33) | 2.85(0.28) | 3.80(0.48) |
| | MSGLD | 15.68(0.95) | 6.48(0.70) | 3.65(0.44) | 5.11(0.78) |
| | NNGP | 1.02(0.33) | 0.99(0.24) | 1.11(0.75) | 0.51(0.14) |
| | SGRLD | 23.8(1.15) | 23.9(1.19) | 25.2(1.25) | 30.5(1.55) |
| $10^5$ | PSGLD | 66.87(2.09) | 10.06(0.65) | 3.59(0.21) | 11.3(0.79) |
| | ADAMSGLD | 7.87(0.38) | 2.37(0.27) | 1.15(0.13) | 1.64(0.24) |
| | MSGLD | 12.92(0.67) | 3.15(0.36) | 1.206(0.11) | 1.71(0.13) |
| | NNGP | 0.89(0.08) | 0.75(0.31) | 1.02(0.14) | 0.47(0.07) |
| | SGRLD | 22.7(0.33) | 22.44(0.27) | 22.69(0.13) | 23.23(0.34) |
| $10^6$ | PSGLD | 96.49(3.37) | 13.68(0.81) | 3.04(0.11) | 9.74(0.42) |
| | ADAMSGLD | 6.17(0.13) | 4.56(0.52) | 1.98(0.62) | 2.36(0.83) |
| | MSGLD | 15.07(1.01) | 3.78(0.81) | 2.06(0.30) | 5.01(0.97) |
| | NNGP | 0.81(0.16) | 1.01(0.34) | 0.28(0.05) | 0.52(0.03) |
| | SGRLD | 25.8(0.14) | 26.05(0.18) | 29.62(0.28) | 24.07(0.27) |

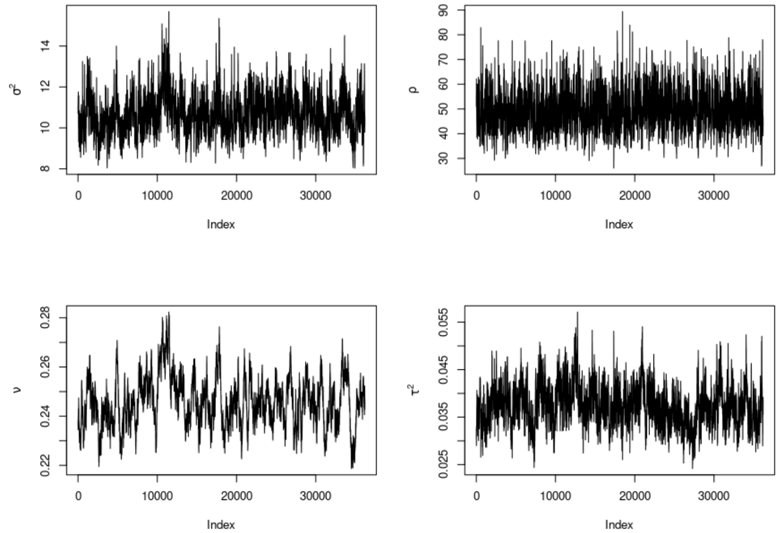

Figure 2: Evolution of SGRLD sampling from the posterior distribution of the covariance parameters.

| $n_{\mathcal{B}}$ | $m$ | $\sigma^2$ | $\rho$ | $\nu$ | $\tau^2$ |
|---|---|---|---|---|---|
| 100 | 10 | $10.18_{(8.79,11.89)}$ | $53.67_{(41.11,66.87)}$ | $0.24_{(0.22,0.25)}$ | $0.04_{(0.03,0.04)}$ |
| | 15 | $11.29_{(9.08,13.53)}$ | $54.95_{(39.18,72.83)}$ | $0.25_{(0.22,0.27)}$ | $0.04_{(0.03,0.04)}$ |
| | 30 | $9.60_{(5.09,13.24)}$ | $46.41_{(13.34,74.52)}$ | $0.24_{(0.20,0.26)}$ | $0.04_{(0.03,0.07)}$ |
| 250 | 10 | $10.52_{(9.08,12.25)}$ | $49.55_{(37.79,62.85)}$ | $0.25_{(0.22,0.26)}$ | $0.04_{(0.03,0.05)}$ |
| | 15 | $10.64_{(7.41,13.57)}$ | $48.93_{(22.94,68.46)}$ | $0.25_{(0.23,0.27)}$ | $0.04_{(0.03,0.05)}$ |
| | 30 | $10.59_{(7.41,13.57)}$ | $46.08_{(22.95,69.46)}$ | $0.25_{(0.23,0.27)}$ | $0.04_{(0.03,0.05)}$ |
| 500 | 10 | $11.02_{(9.74,12.69)}$ | $49.23_{(39.56,60.36)}$ | $0.25_{(0.23,0.27)}$ | $0.04_{(0.03,0.04)}$ |
| | 15 | $11.41_{(9.75,13.20)}$ | $48.42_{(37.98,60.26)}$ | $0.25_{(0.24,0.27)}$ | $0.04_{(0.03,0.04)}$ |
| | 30 | $11.52_{(9.72,13.61)}$ | $48.39_{(36.61,62.35)}$ | $0.26_{(0.24,0.28)}$ | $0.04_{(0.03,0.04)}$ |

Table 6: Sensitivity analysis to the choice of the conditioning set size $m$ and the mini-batch size $n_{\mathcal{B}}$. Posterior mean and 95% credible intervals are displayed for each combination of $n_{\mathcal{B}}$ and $m$.

## C Competing Algorithms

Here we give the detailed algorithms of the SG methods with adaptive drifts. The RMSprop (Root Mean Square Propagation) algorithm is an optimization algorithm originally developed for training neural network models. It adapts the learning rates of each parameter based on the historical gradient information. This can be seen as adaptive preconditioning method.

```
**Algorithm 1:** RMSprop Algorithm
Input: Initial parameter values $\theta_0$, learning rate $h_0$, decay rate $\rho$, small constant $\epsilon$
Output: Optimized parameter values $\theta$
Initialize square gradient accumulator $r_0 = 0$;
while not converged do
    Sample minibatch without repetition; Compute gradient $\bar{g}$ on mini-batch;
    Accumulate squared gradient: $r_t \leftarrow \rho r_{t-1} + (1-\rho)\bar{g} \odot \bar{g}$;
    Update parameters: $\theta_{t+1} \leftarrow \theta_t - h_t \bar{g} \oslash \sqrt{r_t + \epsilon}$;
```

Momentum SGD is an optimization algorithm that uses a Neseterov momentum term to accelerate the convergence in the presence of high curvature or noisy gradients. Momentum SGD proceeds as follows

```
Algorithm 2: Momentum SGD Algorithm
Input: Initial parameter values $\theta_0$, learning rate $h_0$, momentum term $\alpha$
Output: Optimized parameter values $\theta$
Initialize velocity $v_0 = 0$;
while not converged do
    Sample minibatch without repetition; Compute gradient $\bar{g}_t$ on mini-batch;
    Update velocity: $v_t \leftarrow \alpha v_{t-1} - h_t \bar{g}$;
    Update parameters: $\theta_{t+1} \leftarrow \theta_t + v_t$;
```

The Adam algorithm combines ideas from RMSprop and momentum to adaptively adjust learning rates.

```
Algorithm 3: Adam Algorithm
Input: Initial parameter values $\theta_0$, learning rate $h_0$, exponential decay rates for moments $\alpha_1$,
        $\alpha_2$, small constant $\epsilon$
Output: Optimized parameter values $\theta$
Initialize moment estimates $m_0 = 0$, $v_0 = 0$, time step $t = 0$;
while not converged do
    Sample minibatch without repetition; Compute gradient $\bar{g}$ on mini-batch;
    Update biased first moment estimate: $m_{t+1} \leftarrow \alpha_1 m_t + (1-\alpha_1)\bar{g}$;
    Update biased second raw moment estimate: $v_{t+1} \leftarrow \beta_2 v + (1-\alpha_2)\bar{g} \odot \bar{g}$;
    Correct bias in moment estimates: $\hat{m}_t \leftarrow m_t/(1-\alpha_1^t)$, $\hat{v}_t \leftarrow v_t/(1-\alpha_2^t)$;
    Update parameters: $\theta_{t+1} \leftarrow \theta_t - \alpha \hat{m}_t \oslash (\sqrt{\hat{v}_t} + \epsilon)$;
```