# Stochastic Gradient MCMC with Repulsive Forces

**Victor Gallego**
ICMAT-CSIC
victor.gallego@icmat.es

**David Rios Insua**
ICMAT-CSIC
david.rios@icmat.es

## Abstract

We propose a unifying view of two different families of Bayesian inference algorithms, SG-MCMC and SVGD. We show that SVGD plus a noise term can be framed as a multiple chain SG-MCMC method. Instead of treating each parallel chain independently from others, the proposed algorithm implements a repulsive force between particles, avoiding collapse. Experiments in both synthetic distributions and real datasets show the benefits of the proposed scheme.

## 1 Introduction

Bayesian computation lies at the heart of many machine learning models in both academia and industry. Thus, it is of major importance to develop more efficient approximation techniques that tackle the intractable integrals that arise in large scale Bayesian problems. Recent developments in Bayesian techniques applied to large scale datasets or deep models include variational approaches such as *Automatic Differentiation Variational Inference* (ADVI) [1] and *Stein Variational Gradient Descent* (SVGD) [2], or sampling approaches such as *Stochastic Gradient Markov Chain Monte Carlo* (SG-MCMC) [3]. While variational techniques enjoy faster computations than the latter approaches, they rely on optimizing a family of posterior approximates that may not contain the actual posterior distribution. In this work, we draw on a similitude between SG-MCMC and SVGD in order to propose an efficient sampling algorithm.

Any competing MCMC algorithm should verify the following list of properties: *scalability* (we resort to SG-MCMC methods since at each iteration they may be approximated to just require a minibatch of the dataset), *convergence to the true posterior* and *flexibility* (since we provide a parametric formulation of the transition kernel it is possible to adapt other methods such as *Hamiltonian Monte Carlo* [4] or the Nosé-Hoover thermostat method [5]).

Our contributions are summarized in what follows:

- We provide a unifying hybrid scheme of SG-MCMC and SVGD algorithms, satisfying the previous list of requirements.
- Based on it, we can develop new SG-MCMC schemes that include repulsive forces between particles.

## 2 Background and related work

The celebrated work of [3] proposed a general formulation of a continuous-time Markov process that converges to a target distribution $\pi \propto \exp(-H(\boldsymbol{z}))$ with $\boldsymbol{z} \in \mathbb{R}^d$. It is based on the Euler-Maruyama discretization of the generalized Langevin dynamics:

$$\boldsymbol{z}_{t+1} \leftarrow \boldsymbol{z}_t - \epsilon_t \left[(\mathbf{D}(\boldsymbol{z}_t) + \mathbf{Q}(\boldsymbol{z}_t))\nabla H(\boldsymbol{z}_t) + \boldsymbol{\Gamma}(\boldsymbol{z}_t)\right] + \mathcal{N}(\mathbf{0}, 2\epsilon_t \mathbf{D}(\boldsymbol{z}_t)), \tag{1}$$

where $\epsilon_t$ is the stepsize, $\mathbf{D}(\boldsymbol{z})$ is a diffusion matrix, $\mathbf{Q}(\boldsymbol{z})$ is a curl matrix and $\boldsymbol{\Gamma}(\boldsymbol{z})_i = \sum_{j=1}^{d} \frac{\partial}{\partial \boldsymbol{z}_j}(\mathbf{D}_{ij}(\boldsymbol{z}) + \mathbf{Q}_{ij}(\boldsymbol{z}))$ is a correction term. Hence, to obtain any valid SG-MCMC algorithm

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

we simply have to choose the dimensionality of $\boldsymbol{z}$ (i.e., if we augment the space with auxiliary variables as in HMC), $\mathbf{D}$ and $\mathbf{Q}$. For instance, the popular Stochastic gradient Langevin dynamics (SGLD) [6] is obtained when $\mathbf{D} = \mathbf{I}$ and $\mathbf{Q} = \mathbf{0}$. In addition, the Hamiltonian variant can also be recovered if we augment the state space with a $d-$dimensional momentum term $\bar{\boldsymbol{z}} = (\boldsymbol{z}, \boldsymbol{p})$. Then, we set $\mathbf{D} = \mathbf{0}$ and $\mathbf{Q} = \begin{pmatrix} \mathbf{0} & -\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{pmatrix}$.

On the other hand, SVGD [2] frames posterior sampling as an optimization process, in which a set of $K$ particles $\{\boldsymbol{z}_i\}_{i=1}^K$ is evolved iteratively via the velocity field $\boldsymbol{z}_{i,t+1} \leftarrow \boldsymbol{z}_{i,t} + \epsilon\phi(\boldsymbol{z}_{i,t})$. Assuming $q$ is the particle distribution at iteration $t$ and $q_{[\epsilon\phi]}$ is the distribution after an update $(t+1)$, then, the optimal choice of the velocity field $\phi$ can be framed into the optimization problem $\phi^* = \arg\max_{\phi \in \mathcal{F}} \{-\frac{d}{d\epsilon}\mathrm{KL}(q_{[\epsilon\phi]} \| p)\}$, i.e., $\phi$ is chosen so as to maximize the decreasing rate on the KL divergence between the particle distribution and the target. When $\mathcal{F}$ is a RKHS, [2] showed that the optimal velocity field is given by

$$\boldsymbol{z}_{i,t+1} \leftarrow \boldsymbol{z}_{i,t} - \epsilon_t \frac{1}{K} \sum_{j=1}^{K} \left[ k(\boldsymbol{z}_{j,t}, \boldsymbol{z}_{i,t}) \nabla H(\boldsymbol{z}_{j,t}) + \nabla_{\boldsymbol{z}_{j,t}} k(\boldsymbol{z}_{j,t}, \boldsymbol{z}_{i,t}) \right], \tag{2}$$

where the RBF kernel $k(\boldsymbol{z}, \boldsymbol{z}') = \exp(-\frac{1}{h}\|\boldsymbol{z} - \boldsymbol{z}'\|^2)$ is typically adopted. Note that the gradient term $\nabla_{\boldsymbol{z}_{j,t}} k(\boldsymbol{z}_{j,t}, \boldsymbol{z}_{i,t})$ acts as a repulsive force that prevents particles from collapsing. [7] started to consider similitudes between SG-MCMC and SVGD, though in this work we propose the first hybrid scheme between both methods.

## 3 Proposed scheme

We use the framework introduced in [3] in an augmented state space $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_K)$ to obtain a valid posterior sampler that runs multiple Markov chains with interaction. This multiple particle version of SG-MCMC is given by the following equation

$$\boldsymbol{z}_{t+1} \leftarrow \boldsymbol{z}_t - \epsilon_t \left[ (\mathbf{K} + \mathbf{Q}) \boldsymbol{\nabla} + \boldsymbol{\Gamma} \right] + \boldsymbol{\eta}_t, \qquad \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, 2\epsilon_t \mathbf{K}). \tag{3}$$

Now, $\boldsymbol{z}_t = (\boldsymbol{z}_{1,t} \ldots \boldsymbol{z}_{K,t})^\top$ is a $Kd$-dimensional vector defined by the concatenation of $K$ particles. $\boldsymbol{\nabla} \in \mathbb{R}^{K \times d \times 1}$ so that $(\boldsymbol{\nabla})_{i,:} = \nabla H(\boldsymbol{z}_{i,t})$[1]. $\mathbf{K} \in \mathbb{R}^{Kd \times Kd}$ is an expansion of the $D$ matrix, accounting for distance between the particles and $\mathbf{Q} \in \mathbb{R}^{Kd \times Kd}$ might be used if a Hamiltonian variant is to be adopted. $\boldsymbol{\Gamma}$ is the correction term from the framework of [3]. Note that $\mathbf{K}$, $\mathbf{Q}$ and $\boldsymbol{\Gamma}$ can depend on the state $\boldsymbol{z}_t$ (an example will be given below), but we do not make it explicit in order to ease the notation.

Recall that, in matrix notation, the update rule for SVGD can be expressed as

$$\overline{\boldsymbol{z}}_{t+1} \leftarrow \overline{\boldsymbol{z}}_t - \frac{\epsilon_t}{K} \left( \overline{\mathbf{K}\boldsymbol{\nabla}} + \overline{\boldsymbol{\Gamma}} \right) \tag{4}$$

where $\overline{\mathbf{K}} \in \mathbb{R}^{K \times K}$ so that $(\overline{\mathbf{K}})_{ij} = k(\boldsymbol{z}_i, \boldsymbol{z}_j)$, $\overline{\boldsymbol{\nabla}} \in \mathbb{R}^{K \times d}$ and $\overline{\boldsymbol{z}}_t \in \mathbb{R}^{K \times d}$. Casting the later matrix as a tensor $\boldsymbol{\nabla} \in \mathbb{R}^{K \times d \times 1}$ and the former one as a tensor $\mathbf{K} \in \mathbb{R}^{(K \times d) \times (K \times d)}$ by broadcasting along the second and fourth axes, we may associate $\mathbf{K}$ with SG-MCMC's diffusion matrix $\mathbf{D}$ over a $Kd-$dimensional space. Appendix A shows the precise definition of the matrix $\mathbf{K}$.

From this perspective, Eq. (4) (SVGD) can be seen as a special case of Eq. (3) when the curl matrix $\mathbf{Q} = \mathbf{0}$ and the noise term is added. We shall refer to this perturbed variant of SVGD as *Parallel SGLD plus repulsion* (SGLD+R):

$$\boldsymbol{z}_{t+1} \leftarrow \boldsymbol{z}_t - \frac{\epsilon_t}{K} \left( \mathbf{K}\boldsymbol{\nabla} + \boldsymbol{\Gamma} \right) + \boldsymbol{\eta}_t, \qquad \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, 2\epsilon_t \mathbf{K}/K). \tag{5}$$

Since matrix $\mathbf{K}$ is definite positive (it was constructed from the RBF kernel), we may now use the key result from [3] (Theorem 1) to derive the following property:

**Proposition 1.** *SGLD+R (or its general form, Eq.* (3)*) has* $\pi(\boldsymbol{z}) = \prod_{k=1}^{K} \pi(\boldsymbol{z}_k)$ *as stationary distribution, and the proposed discretizations are asymptotically exact as* $\epsilon_t \to 0$.

[1] Though $\boldsymbol{\nabla} \in \mathbb{R}^{Kd \times 1}$ to allow multiplication by $\mathbf{K} + \mathbf{Q}$, we reshape it as $\boldsymbol{\nabla} \in \mathbb{R}^{K \times d \times 1}$ to better illustrate how it is defined.

Shown that SVGD plus a noise term can be framed as a SG-MCMC method, we may now explore the design spaces of the $\mathbf{K}$ and $\mathbf{Q}$ matrices. However, for the rest of the paper we resort to the case $\mathbf{Q} = \mathbf{0}$ (i.e., we will just experimentally study SGLD with repulsion between multiple particles).

Algorithm 1 shows how to set it up. Finally, our proposed method is amenable to parallelization, since the mini-batch setting from SG-MCMC can be adopted (see Appendix B).

## 4 Experiments

This Section describes the experiments developed to empirically test the proposed scheme. Code will be released at `https://github.com/vicgalle/sgmcmc-force` and we rely on Tensorflow-probability [8] as the main package.

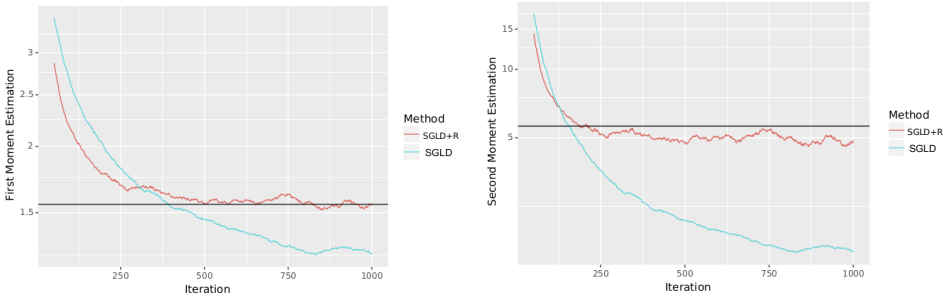
**Synthetic distributions**. We test our proposed scheme in the following distributions.

- **Mixture of Exponentials (MoE)**. Two exponential distributions with different scale parameters $\lambda_1 = 1.5, \lambda_2 = 0.5$ and mixture proportions $\pi_1 = 1/3, \pi_2 = 2/3$.
- **Mixture of 2D Gaussians (MoG)**. A grid of $3\times3$ equally distributed isotropic 2D Gaussians, see Figure 2(d) for the density plot.

We compare two sampling methods, SGLD with $K$ parallel chains, and our proposed scheme, SGLD+R. Note that the only difference in these two sampling algorithms is that for the former $\mathbf{K} = \mathbf{I}$ whereas the latter accounts for repulsion between particles. Table 1 reports the effective sampling size metrics for each method using $K = 10$ particles. Note that while ESS/s are similar, the repulsive forces in SGLD+R makes for a more efficient exploration, resulting in much lower estimation errors. Figures 1 and 2 seem to confirm this fact. In addition, even when increasing the number of particles $K$, SGLD+R achieves lower errors than SGLD (see Fig. 3).

| | ESS | | ESS/s | | Error of $\mathbb{E}[X]$ | |
|---|---|---|---|---|---|---|
| **Distribution** | **SGLD** | **SGLD+R** | **SGLD** | **SGLD+R** | **SGLD** | **SGLD+R** |
| MoE | 44.3 | **59.1** | 51.5 | **61.0** | 0.39 | **0.14** |
| MoG | 151.3 | **169.5** | **36.3** | 32.5 | 1.42 | **1.19** |

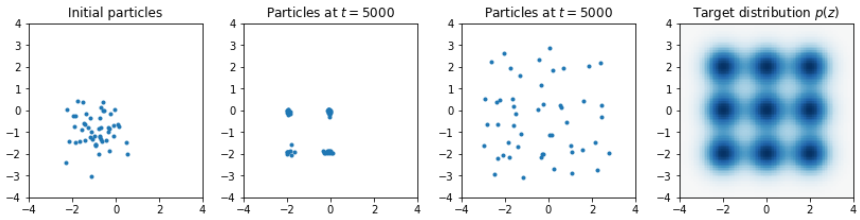
Table 1: Results for the two synthetic distributions task




(a) Estimation of $\mathbb{E}[X]$ (b) Estimation of $\mathbb{E}\left[X^2\right]$

Figure 1: Evolution of estimation during the MoE experiment. 10 particles are used and black line depicts the exact value to be estimated

**Bayesian Neural Network**. We test the proposed scheme in a suite of regression tasks using a feed-forward neural network with 50 hidden units and ReLu activations. The datasets are taken from the UCI repository. We use minibatches of size 100, but while we use the same experimental setting as [2], we simply use SGD instead of Adagrad, since it is not clear that Proposition 1 can be extended to the non-SGD case. As before, we compare SGLD and SGLD+R, reporting average root mean squared error and log-likelihood over a test set in Table 2. We observe that SGLD+R typically outperforms SGLD. During the experiments, we noted that in order to reduce computation time, during the last half of training we could disable the repulsion between particles without incurring in performance cost.




(a) Particles from prior (b) SGLD dynamics (c) SGLD+R dyn. (d) Density of MoG

Figure 2: Evolution of the particles during the MoG experiment

| | Avg. Test RMSE | | Avg. Test LL | |
|---|---|---|---|---|
| **Dataset** | **SGLD** | **SGLD+R** | **SGLD** | **SGLD+R** |
| Boston | $2.392 \pm 0.018$ | $\mathbf{2.295 \pm 0.017}$ | $-2.551 \pm 0.018$ | $-2.575 \pm 0.007$ |
| Kin8nm | $0.104 \pm 0.001$ | $0.104 \pm 0.001$ | $0.826 \pm 0.005$ | $0.831 \pm 0.006$ |
| Naval | $0.008 \pm 0.000$ | $0.008 \pm 0.000$ | $3.379 \pm 0.011$ | $\mathbf{3.428 \pm 0.019}$ |
| Protein | $4.810 \pm 0.003$ | $\mathbf{4.794 \pm 0.003}$ | $-2.991 \pm 0.000$ | $\mathbf{-2.987 \pm 0.001}$ |
| Wine | $0.522 \pm 0.004$ | $\mathbf{0.514 \pm 0.004}$ | $-0.765 \pm 0.008$ | $\mathbf{-0.750 \pm 0.007}$ |
| Yacht | $0.942 \pm 0.015$ | $\mathbf{0.894 \pm 0.029}$ | $-1.211 \pm 0.020$ | $-1.172 \pm 0.026$ |

Table 2: Results for the BNN experiments

## 5 Conclusions and further work

This paper shows how to generate new SG-MCMC methods consisting in multiple chains plus repulsion between the particles. Instead of the naïve parallelization, in which a particle from a chain is agnostic to the others, we showed how it is possible to adapt another method from the literature, SVGD, in order to account for better exploration of the space, avoiding collapse between particles. Further work shall explore different matrices $\mathbf{K}$ and $\mathbf{Q}$ in order to further accelerate the sampling process.

**Acknowledgments** Authors thank Roi Naveiro from the ICMAT Datalab group for interesting discussions. Authors also thank the anonymous reviewers for their suggestions. VG acknowledges support from grant FPU16-05034. DRI is grateful to the MINECO MTM2014-56949-C3-1-R project and the AXA-ICMAT Chair in Adversarial Risk Analysis. All authors acknowledge support from the Severo Ochoa Excellence Programme SEV-2015-0554.

## References

[1] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[2] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pages 2378–2386, 2016.

[3] Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient mcmc. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2917–2925. Curran Associates, Inc., 2015.

[4] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International Conference on Machine Learning*, pages 1683–1691, 2014.

[5] Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D Skeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. In *Advances in Neural Information Processing Systems*, pages 3203–3211, 2014.

[6] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

[7] Qiang Liu. Stein variational gradient descent as gradient flow. In *Advances in Neural Information Processing Systems*, pages 3115–3123, 2017.

**Algorithm 1** Bayesian Inference via SGLD+R

**Input:** A target distribution with density function $\pi(\boldsymbol{z}) \propto \exp(-H(\boldsymbol{z}))$.
**Output:** A set of particles $\{\boldsymbol{z}_i\}_{i=1}^{MK}$ that approximates the target distribution.
Sample initial set of particles from prior: $\boldsymbol{z}_1^0, \boldsymbol{z}_2^0, \dots \boldsymbol{z}_K^0 \sim \pi(\boldsymbol{z})$.
**for** each iteration $t$ **do**

$$\boldsymbol{z}_i^{t+1} \leftarrow \boldsymbol{z}_i^t - \epsilon_t \frac{1}{K} \sum_{j=1}^{K} \left[ k(\boldsymbol{z}_j^t, \boldsymbol{z}_i^t) \nabla_{\boldsymbol{z}_j^t} H(\boldsymbol{z}_j^t) + \nabla_{\boldsymbol{z}_j^t} k(\boldsymbol{z}_j^t, \boldsymbol{z}_i^t) \right] + \boldsymbol{\eta}_i^t \tag{6}$$

where $\boldsymbol{\eta}_i^t$ is the noise for particle $i$ defined as in Eq (5).

After a burn-in period, start collecting particles: $\{\boldsymbol{z}_i\}_{i=1}^{NK} \leftarrow \{\boldsymbol{z}_i\}_{i=1}^{(N-1)K} \cup \{\boldsymbol{z}_1^{t+1}, \dots, \boldsymbol{z}_K^{t+1}\}$
**end for**

[8] Joshua V Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A Saurous. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

## A Construction of $\mathbf{K}$

The big matrix $\mathbf{K}$ from Eq. (5) is defined as a permuted block-diagonal matrix consisting of $d$ repeating kernel matrices $\overline{\mathbf{K}}$:

$$\mathbf{K} = \begin{bmatrix} \overline{\mathbf{K}} & & & \\ & \overline{\mathbf{K}} & & \\ & & \ddots & \\ & & & \overline{\mathbf{K}} \end{bmatrix} \mathbf{P},$$

with $\mathbf{P}$ being the $Kd \times Kd$ permutation matrix

$$\mathbf{P} = \left[\begin{array}{c|c|c|c} 1 & & & \\ & 1 & \ddots & \\ & & & 1 \\ \hline 1 & & & \\ & 1 & \ddots & \\ & & & 1 \\ \hline \ddots & \ddots & \ddots & \ddots \\ \hline & 1 & & \\ & & 1 & \ddots \\ & & & 1 \end{array}\right].$$

The permutation matrix $\mathbf{P}$ rearranges the block-diagonal kernel matrix in order to match with the dimension ordering of the state space $\boldsymbol{z}_t = (\boldsymbol{z}_{1,t} \dots \boldsymbol{z}_{K,t})^\top$. Note that with this convention, $\mathbf{K}\boldsymbol{\nabla}$ is equivalent to $\overline{\mathbf{K}\boldsymbol{\nabla}}$, only differing in the shape of the resulting matrix. This allows us to frame SVGD plus the noise term as a valid scheme from the SG-MCMC framework of [3], using the $Kd$-dimensional augmented state space.

## B Implementation details

Algorithm 1 details SGLD+R. Note the proposed scheme preserves the efficient properties of the original SG-MCMC and SVGD algorithms. The main computational bottleneck lies in the evaluation of the gradient $\nabla_{\boldsymbol{z}} \log \pi(\boldsymbol{z})$, which can be troublesome in the big data setting such as when $\pi(\boldsymbol{z}|\mathcal{D}) = p(\boldsymbol{z}) \Pi_{i=1}^{N} p(D_k|\boldsymbol{z})$. We then may approximate the true gradient with an unbiased estimator taken along a minibatch of datapoints $\Omega \subset \{1, 2, \dots, N\}$ in the usual way:

$$\nabla_{\boldsymbol{z}} \log \pi(\boldsymbol{z}|\mathcal{D}) \approx \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}) + \frac{N}{|\Omega|} \sum_{i \in \Omega} \nabla_{\boldsymbol{z}} \log p(D_i|\boldsymbol{z}).$$

# C Experiment details

### Synthetic distributions

For the Mixture of Exponentials experiment, the pdf used is

$$p(z) = \sum_{i=1}^{2} \pi_i \lambda_i \exp(-\lambda_i z).$$

Note that the exact value of the first and second moments can be computed using

$$\mathbb{E}\left[z^n\right] = \sum_{i=1}^{2} \pi_i \frac{n!}{\lambda_i^n}$$

with $n \in \mathbb{N}$. Since $z > 0$, in order to use the proposed scheme, we reparameterize using the $\log$ function. The pdf of the transformation $y = \log(z)$ can be computed using

$$p(y) = p(\log^{-1}(y))|\mathrm{D} \log^{-1}(y))|.$$

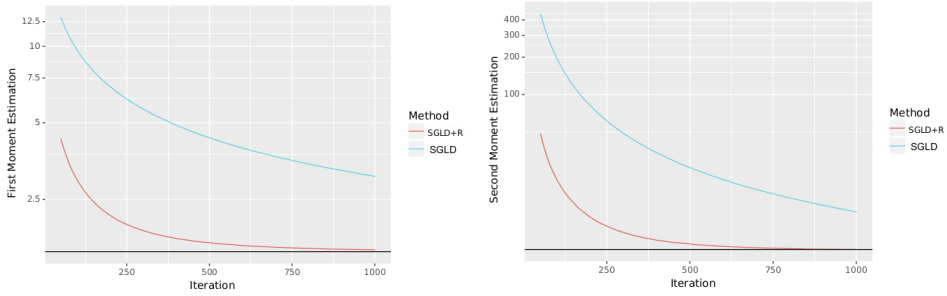
For the Mixture of Gaussians experiment, we set $\Sigma = \mathrm{diag}(0.1, 0.1)$ and place each of the nine Gaussians centered at each each of the following points $\{(-2,-2),(-2,0),(-2,2),(0,-2),(0,0),(0,2),(2,-2),(2,0),(2,2)\}$.

For the computation of the error of $\mathbb{E}\left[X\right]$ in Table 1, we sample for 500 iterations after discarding the first 500 iterations as burn-in, and we collected samples every 10 iterations to reduce correlation between samples. For the MoE case we used 10 particles whereas for the MoG task we used 20 particles due to the increased number of modes.

### Bayesian Neural Network

Learning rate was chosen from a grid $\{1e-5, \ldots, 1e-3\}$ validated on another fold. The number of iterations was set to 2000 in every experiment. As before, to make predictions we collect samples every 10 iterations after a burn-in period. 20 particles were used for each of the tested datasets.

# D Additional results




(a) Estimation of $\mathbb{E}\left[X\right]$ (b) Estimation of $\mathbb{E}\left[X^2\right]$

Figure 3: Evolution of estimation during the MoE experiment. 100 particles are used and black line depicts the exact value to be estimated