
Quasi-Monte Carlo Variational Inference

Alexander Buchholz^{*1} Florian Wenzel^{*2} Stephan Mandt³

Abstract

Many machine learning problems involve Monte Carlo gradient estimators. As a prominent example, we focus on Monte Carlo variational inference (MCVI) in this paper. The performance of MCVI crucially depends on the variance of its stochastic gradients. We propose variance reduction by means of Quasi-Monte Carlo (QMC) sampling. QMC replaces N i.i.d. samples from a uniform probability distribution by a deterministic sequence of samples of length N . This sequence covers the underlying random variable space more evenly than i.i.d. draws, reducing the variance of the gradient estimator. With our novel approach, both the score function and the reparameterization gradient estimators lead to much faster convergence. We also propose a new algorithm for Monte Carlo objectives, where we operate with a constant learning rate and increase the number of QMC samples per iteration. We prove that this way, our algorithm can converge asymptotically at a faster rate than SGD. We furthermore provide theoretical guarantees on QMC for Monte Carlo objectives that go beyond MCVI, and support our findings by several experiments on large-scale data sets from various domains.

1. Introduction

In many situations in machine learning and statistics, we encounter objective functions which are expectations over continuous distributions. Among other examples, this situation occurs in reinforcement learning (Sutton and Barto, 1998) and variational inference (Jordan et al., 1999). If the expectation cannot be computed in closed form, an approximation can often be obtained via Monte Carlo (MC) sampling from the underlying distribution. As most optimization pro-

cedures rely on the gradient of the objective, a MC gradient estimator has to be built by sampling from this distribution. The finite number of MC samples per gradient step introduces noise. When averaging over multiple samples, the error in approximating the gradient can be decreased, and thus its variance reduced. This guarantees stability and fast convergence of stochastic gradient descent (SGD).

Certain objective functions require a large number of MC samples per stochastic gradient step. As a consequence, the algorithm gets slow. It is therefore desirable to obtain the same degree of variance reduction with fewer samples. This paper proposes the idea of using Quasi-Monte Carlo (QMC) samples instead of i.i.d. samples to achieve this goal.

A QMC sequence is a deterministic sequence which covers a hypercube $[0, 1]^d$ more regularly than random samples. When using a QMC sequence for Monte Carlo integration, the mean squared error (MSE) decreases asymptotically with the number of samples N as $O(N^{-2}(\log N)^{2d-2})$ (Leobacher and Pillichshammer, 2014). In contrast, the naive MC integration error decreases as $O(N^{-1})$. Since the cost of generating N QMC samples is $O(N \log N)$, this implies that a much smaller number of operations per gradient step is required in order to achieve the same precision (provided that N is large enough). Alternatively, we can achieve a larger variance reduction with the same number of samples, allowing for larger gradient steps and therefore also faster convergence. This paper investigates the benefits of this approach both experimentally and theoretically.

Our ideas apply in the context of Monte Carlo variational inference (MCVI), a set of methods which make approximate Bayesian inference scalable and easy to use. Variance reduction is an active area of research in this field. Our algorithm has the advantage of being very general; it can be easily implemented in existing software packages such as STAN and Edward (Carpenter et al., 2017; Tran et al., 2016). In Appendix D we show how our approach can be easily implemented in your existing code.

The main contributions are as follows:

- We investigate the idea of using QMC sequences for Monte Carlo variational inference. While the usage of QMC for VI has been suggested in the outlook section of Ranganath et al. (2014), to our knowledge, we are the first to actually investigate this approach both

^{*}Equal contribution ¹ENSAE-CREST, Paris ²TU Kaiserslautern, Germany ³Disney Research, Los Angeles, USA. Correspondence to: Alexander Buchholz <alexander.buchholz@ensae.fr>, Florian Wenzel <wenzelfl@hu-berlin.de>, Stephan Mandt <stephan.mandt@gmail.com>.

theoretically and experimentally.

- We show that when using a randomized version of QMC (RQMC), the resulting stochastic gradient is unbiased and its variance is asymptotically reduced. We also show that when operating SGD with a constant learning rate, the stationary variance of the iterates is reduced by a factor of N , allowing us to get closer to the optimum.
- We propose an algorithm which operates at a constant learning rate, but increases the number of RQMC samples over iterations. We prove that this algorithm has a better asymptotic convergence rate than SGD.
- Based on three different experiments and for two popular types of gradient estimators we illustrate that our method allows us to train complex models several orders of magnitude faster than with standard MCVI.

Our paper is structured as follows. Section 2 reviews related work. Section 3 explains our method and exhibits our theoretical results. In Section 4 we describe our experiments and show comparisons to other existing approaches. Finally, Section 5 concludes and lays out future research directions.

2. Related Work

Monte Carlo Variational Inference (MCVI) Since the introduction of the score function (or REINFORCE) gradient estimator for variational inference (Paisley et al., 2012; Ranganath et al., 2014), Monte Carlo variational inference has received an ever-growing attention, see Zhang et al. (2017a) for a recent review. The introduction of the gradient estimator made VI applicable to non-conjugate models but highly depends on the variance of the gradient estimator. Therefore various variance reduction techniques have been introduced; for example Rao-Blackwellization and control variates, see Ranganath et al. (2014) and importance sampling, see Ruiz et al. (2016a); Burda et al. (2016).

At the same time the work of Kingma and Welling (2014); Rezende et al. (2014) introduced reparameterization gradients for MCVI, which typically exhibits lower variance but are restricted to models where the variational family can be reparametrized via a differentiable mapping. In this sense MCVI based on score function gradient estimators is more general but training the algorithm is more difficult. A unifying view is provided by Ruiz et al. (2016b). Miller et al. (2017) introduce a modification of the reparametrized version, but relies itself on assumptions on the underlying variational family. Roeder et al. (2017) propose a lower variance gradient estimator by omitting a term of the ELBO. The idea of using QMC in order to reduce the variance has been suggested by Ranganath et al. (2014) and Ruiz et al. (2016a) and used for a specific model by Tran et al. (2017), but without

a focus on analyzing or benchmarking the method.

Quasi-Monte Carlo and Stochastic Optimization Besides the generation of random samples for approximating posterior distributions (Robert and Casella, 2013), Monte Carlo methods are used for calculating expectations of intractable integrals via the law of large numbers. The error of the integration with random samples goes to zero at a rate of $O(N^{-1})$ in terms of the MSE. For practical application this rate can be too slow. Faster rates of convergence in reasonable dimensions can be obtained by replacing the randomness by a deterministic sequence, also called Quasi-Monte Carlo.

Compared to Monte Carlo and for sufficiently regular functions, QMC reaches a faster rate of convergence of the approximation error of an integral. Niederreiter (1992); L’Ecuyer and Lemieux (2005); Leobacher and Pillichshammer (2014); Dick et al. (2013) provide excellent reviews on this topic. From a theoretical point of view, the benefits of QMC vanish in very high dimensions. Nevertheless, the error bounds are often too pessimistic and in practice, gains are observed up to dimension 150, see Glasserman (2013).

QMC has frequently been used in financial applications (Glasserman, 2013; Joy et al., 1996; Lemieux and L’Ecuyer, 2001). In statistics, some applications include particle filtering (Gerber and Chopin, 2015), approximate Bayesian computation (Buchholz and Chopin, 2017), control functionals (Oates and Girolami, 2016) and Bayesian optimal design (Drovandi and Tran, 2018). Yang et al. (2014) used QMC in the context of large scale kernel methods.

Stochastic optimization has been pioneered by the work of Robbins and Monro (1951). As stochastic gradient descent suffers from noisy gradients, various approaches for reducing the variance and adapting the step size have been introduced (Johnson and Zhang, 2013; Kingma and Ba, 2015; Defazio et al., 2014; Duchi et al., 2011; Zhang et al., 2017b). Extensive theoretical results on the convergence of stochastic gradients algorithms are provided by Moulines and Bach (2011). Mandt et al. (2017) interpreted stochastic gradient descent with constant learning rates as approximate Bayesian inference. Some recent reviews are for example Bottou et al. (2016); Nesterov (2013). Naturally, concepts from QMC can be beneficial to stochastic optimization. Contributions on exploiting this idea are e.g. Gerber and Bornn (2017) and Drew and Homem-de Mello (2006).

3. Quasi-Monte Carlo Variational Inference

In this Section, we introduce Quasi-Monte Carlo Variational Inference (QMCVI), using randomized QMC (RQMC) for variational inference. We review MCVI in Section 3.1. RQMC

and the details of our algorithm are exposed in Section 3.2. Theoretical results are given in Section 3.3.

3.1. Background: Monte Carlo Variational Inference

Variational inference (VI) is key to modern probabilistic modeling and Bayesian deep learning (Jordan et al., 1999; Blei et al., 2017; Zhang et al., 2017a). In Bayesian inference, the object of interest is a posterior distribution of latent variables \mathbf{z} given observations \mathbf{x} . VI approximates Bayesian inference by an optimization problem which we can solve by (stochastic) gradient ascent (Jordan et al., 1999; Hoffman et al., 2013).

In more detail, VI builds a tractable approximation of the posterior $p(\mathbf{z}|\mathbf{x})$ by minimizing the KL-divergence between a variational family $q(\mathbf{z}|\lambda)$, parametrized by free parameters $\lambda \in \mathbb{R}^d$, and $p(\mathbf{z}|\mathbf{x})$. This is equivalent to maximizing the so-called evidence lower bound (ELBO):

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\mathbf{z}|\lambda)}[\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}|\lambda)]. \quad (1)$$

In classical variational inference, the expectations involved in (1) are carried out analytically (Jordan et al., 1999). However, this is only possible for the fairly restricted class of so-called conditionally conjugate exponential family models (Hoffman et al., 2013). More recently, black-box variational methods have gained momentum, which make the analytical evaluation of these expectation redundant, and which shall be considered in this paper.

Maximizing the objective (1) is often based on a gradient ascent scheme. However, a direct differentiation of the objective (1) with respect to λ is not possible, as the measure of the expectation depends on this parameter. The two major approaches for overcoming this issue are the score function estimator and the reparameterization estimator.

Score Function Gradient The score function gradient (also called REINFORCE gradient) (Ranganath et al., 2014) expresses the gradient as expectation with respect to $q(\mathbf{z}|\lambda)$ and is given by

$$\begin{aligned} \nabla_{\lambda} \mathcal{L}(\lambda) &= \mathbb{E}_{q(\mathbf{z}|\lambda)}[\nabla_{\lambda} \log q(\mathbf{z}|\lambda) (\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}|\lambda))]. \quad (2) \end{aligned}$$

The gradient estimator is obtained by approximating the expectation with independent samples from the variational distribution $q(\mathbf{z}|\lambda)$. This estimator applies to continuous and discrete variational distributions.

Reparameterization Gradient The second approach is based on the reparameterization trick (Kingma and Welling, 2014), where the distribution over \mathbf{z} is expressed as a deterministic transformation of another distribution over a noise

variable ε , hence $\mathbf{z} = g_{\lambda}(\varepsilon)$ where $\varepsilon \sim p(\varepsilon)$. Using the reparameterization trick, the ELBO is expressed as expectation with respect to $p(\varepsilon)$ and the derivative is moved inside the expectation:

$$\begin{aligned} \nabla_{\lambda} \mathcal{L}(\lambda) &= \mathbb{E}_{p(\varepsilon)}[\nabla_{\lambda} \log p(\mathbf{x}, g_{\lambda}(\varepsilon)) - \nabla_{\lambda} \log q(g_{\lambda}(\varepsilon)|\lambda)]. \quad (3) \end{aligned}$$

The expectation is approximated using a MC sum of independent samples from $p(\varepsilon)$. In its basic form, the estimator is restricted to distributions over continuous variables.

MCVI In the general setup of MCVI considered here, the gradient of the ELBO is represented as an expectation $\nabla_{\lambda} \mathcal{L}(\lambda) = \mathbb{E}[g_{\tilde{\mathbf{z}}}(\lambda)]$ over a random variable $\tilde{\mathbf{z}}$. For the score function estimator we choose g according to Equation (2) with $\tilde{\mathbf{z}} = \mathbf{z}$ and for the reparameterization gradient according to Equation (3) with $\tilde{\mathbf{z}} = \varepsilon$, respectively. This allows us to obtain a stochastic estimator of the gradient by an average over a finite sample $\{\tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_N\}$ as $\hat{g}_N(\lambda_t) = (1/N) \sum_{i=1}^N g_{\tilde{\mathbf{z}}_i}(\lambda_t)$. This way, the ELBO can be optimized by stochastic optimization. This is achieved by iterating the SGD updates with decreasing step sizes α_t :

$$\lambda_{t+1} = \lambda_t + \alpha_t \hat{g}_N(\lambda_t). \quad (4)$$

The convergence of the gradient ascent scheme in (4) tends to be slow when gradient estimators have a high variance. Therefore, various approaches for reducing the variance of both gradient estimators exist; e.g. control variates (CV), Rao-Blackwellization and importance sampling. However these variance reduction techniques do not improve the $O(N^{-1})$ rate of the MSE of the estimator, except under some restrictive conditions (Oates et al., 2017). Moreover, the variance reduction schemes must often be tailored to the problem at hand.

3.2. Quasi-Monte Carlo Variational Inference

Quasi Monte Carlo Low discrepancy sequences, also called QMC sequences, are used for integrating a function ψ over the $[0, 1]^d$ hypercube. When using standard i.i.d. samples on $[0, 1]^d$, the error of the approximation is $O(N^{-1})$. QMC achieves a rate of convergence in terms of the MSE of $O(N^{-2}(\log N)^{2d-2})$ if ψ is sufficiently regular (Leobacher and Pillichshammer, 2014). This is achieved by a deterministic sequence that covers $[0, 1]^d$ more evenly.

On a high level, QMC sequences are constructed such that the number of points that fall in a rectangular volume is proportional to the volume. This idea is closely linked to stratification. Halton sequences e.g. are constructed using coprime numbers (Halton, 1964). Sobol sequences are based on the reflected binary code (Antonov and Saleev, 1979). The exact construction of QMC sequences is quite involved and we refer to Niederreiter (1992); Leobacher

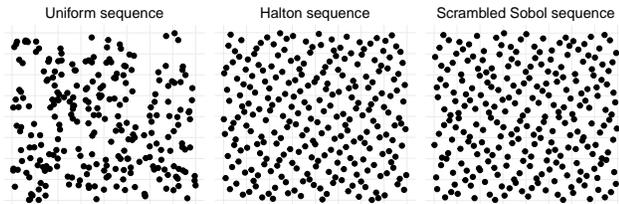


Figure 1: MC (left), QMC (center) and RQMC (right) sequences of length $N = 256$ on $[0, 1]^2$. QMC and RQMC tend to cover the target space more evenly.

and Pillichshammer (2014); Dick et al. (2013) for more details.

The approximation error of QMC increases with the dimension, and it is difficult to quantify. Carefully reintroducing randomness while preserving the structure of the sequence leads to randomized QMC. RQMC sequences are unbiased and the error can be assessed by repeated simulation. Moreover, under slightly stronger regularity conditions on F we can achieve rates of convergence of $O(N^{-2})$ (Gerber, 2015). For illustration purposes, we show different sequences in Figure 1. In Appendix A we provide more technical details.

QMC or RQMC can be used for integration with respect to arbitrary distributions by transforming the initial sequence on $[0, 1]^d$ via a transformation Γ to the distribution of interest. Constructing the sequence typically costs $O(N \log N)$ (Gerber and Chopin, 2015).

QMC and VI We suggest to replace N independent MC samples for computing $\hat{g}_N(\lambda_t)$ by an RQMC sequence of the same length. With our approach, the variance of the gradient estimators becomes $O(N^{-2})$, and the costs for creating the sequence is $O(N \log N)$. The incorporation of RQMC in VI is straightforward: instead of sampling \mathbf{z} as independent MC samples, we generate a uniform RQMC sequence $\mathbf{u}_1, \dots, \mathbf{u}_N$ and transform this sequence via a mapping Γ to the original random variable $\mathbf{z} = \Gamma(\mathbf{u})$. Using this transformation we obtain the RQMC gradient estimator

$$\hat{g}_N(\lambda_t) = (1/N) \sum_{i=1}^N g_{\Gamma(\mathbf{u}_i)}(\lambda). \quad (5)$$

From a theoretical perspective, the function $\mathbf{u} \mapsto g_{\Gamma(\mathbf{u})}(\lambda)$ has to be sufficiently smooth for all λ . For commonly used variational families this transformation is readily available. Although evaluating these transforms adds computational overhead, we found this cost negligible in practice. For example, in order to sample from a multivariate Gaussian $\mathbf{z}_n \sim \mathcal{N}(\mu, \Sigma)$, we generate an RQMC sequence \mathbf{u}_n and apply the transformation $\mathbf{z}_n = \Phi^{-1}(\mathbf{u}_n)\Sigma^{1/2} + \mu$, where $\Sigma^{1/2}$ is the Cholesky decomposition of Σ and Φ^{-1} is the component-wise inverse cdf of a standard normal distribution. Similar

procedures are easily obtained for exponential, Gamma, and other distributions that belong to the exponential family. Algorithm 1 summarizes the procedure.

Algorithm 1: Quasi-Monte Carlo Variational Inference

Input: Data \mathbf{x} , model $p(\mathbf{x}, \mathbf{z})$, variational family $q(\mathbf{z}|\lambda)$

Result: Variational parameters λ^*

```

1 while not converged do
2   Generate uniform RQMC sequence  $\mathbf{u}_{1:N}$ 
3   Transform the sequence via  $\Gamma$ 
4   Estimate the gradient  $\hat{g}_N(\lambda_t) = \frac{1}{N} \sum_{i=1}^N g_{\Gamma(\mathbf{u}_i)}(\lambda_t)$ 
5   Update  $\lambda_{t+1} = \lambda_t + \alpha_t \hat{g}_N(\lambda_t)$ 
    
```

RQMC samples can be generated via standard packages such as `randtoolbox` (Christophe and Petr, 2015), available in R. Existing MCVI algorithms are adapted by replacing the random variable sampler by an RQMC version. Our approach reduces the variance in MCVI and applies in particular to the reparametrization gradient estimator and the score function estimator. RQMC can in principle be combined with additional variance reduction techniques such as CV, but care must be taken as the optimal CV for RQMC are not the same as for MC (Hickernell et al., 2005).

3.3. Theoretical Properties of QMCVI

In what follows we give a theoretical analysis of using RQMC in stochastic optimization. Our results apply in particular to VI but are more general.

QMCVI leads to faster convergence in combination with Adam (Kingma and Ba, 2015) or Adagrad (Duchi et al., 2011), as we will show empirically in Section 4. Our analysis, presented in this section, underlines this statement for the simple case of SGD with fixed step size in the Lipschitz continuous (Theorem 1) and strongly convex case (Theorem 2). We show that for N sufficiently large, SGD with RQMC samples reaches regions closer to the true optimizer of the ELBO. Moreover, we obtain a faster convergence rate than SGD when using a fixed step size and increasing the sample size over iterations (Theorem 3).

RQMC for Optimizing Monte Carlo Objectives We step back from black box variational inference and consider the more general setup of optimizing Monte Carlo objectives. Our goal is to minimize a function $F(\lambda)$, where the optimizer has only access to a noisy, unbiased version $\hat{F}_N(\lambda)$, with $\mathbb{E}[\hat{F}_N(\lambda)] = F(\lambda)$ and access to an unbiased noisy estimator of the gradients $\hat{g}_N(\lambda)$, with $\mathbb{E}[\hat{g}_N(\lambda)] = \nabla F(\lambda)$. The optimum of $F(\lambda)$ is λ^* .

We furthermore assume that the gradient estimator $\hat{g}_N(\lambda)$ has the form as in Eq. 5, where Γ is a reparameterization function that converts uniform samples from the hypercube

into samples from the target distribution. In this paper, $\mathbf{u}_1, \dots, \mathbf{u}_N$ is an RQMC sequence.

In the following theoretical analysis, we focus on SGD with a constant learning rate α . The optimal value λ^* is approximated by SGD using the update rule

$$\lambda_{t+1} = \lambda_t - \alpha \hat{g}_N(\lambda_t). \quad (6)$$

Starting from λ_1 the procedure is iterated until $|\hat{F}_N(\lambda_t) - \hat{F}_N(\lambda_{t+1})| \leq \epsilon$, for a small threshold ϵ . The quality of the approximation $\lambda_T \approx \lambda^*$ crucially depends on the variance of the estimator \hat{g}_N (Johnson and Zhang, 2013).

Intuitively, the variance of $\hat{g}_N(\lambda)$ based on an RQMC sequence will be $O(N^{-2})$ and thus for N large enough, the variance will be smaller than for the MC counterpart, that is $O(N^{-1})$. This will be beneficial to the optimization procedure defined in (6). Our following theoretical results are based on standard proof techniques for stochastic approximation, see e.g. Bottou et al. (2016).

Stochastic Gradient Descent with Fixed Step Size In the case of functions with Lipschitz continuous derivatives, we obtain the following upper bound on the norm of the gradients.

Theorem 1 *Let F be a function with Lipschitz continuous derivatives, i.e. there exists $L > 0$ s.t. $\forall \lambda, \bar{\lambda} \|\nabla F(\lambda) - \nabla F(\bar{\lambda})\|_2^2 \leq L\|\lambda - \bar{\lambda}\|_2^2$, let $U_N = \{\mathbf{u}_1, \dots, \mathbf{u}_N\}$ be an RQMC sequence and let $\forall \lambda, G : \mathbf{u} \mapsto g_{\Gamma(\mathbf{u})}(\lambda)$ has cross partial derivatives of up to order d . Let the constant learning rate $\alpha < 2/L$ and let $\mu = 1 - \alpha L/2$. Then $\forall \lambda$, $\text{tr Var}_{U_N}[\hat{g}_N(\lambda)] \leq M_V \times r(N)$, where $M_V < \infty$ and $r(N) = O(N^{-2})$ and*

$$\frac{\sum_{t=1}^T \mathbb{E}\|\nabla F(\lambda_t)\|_2^2}{T} \leq \frac{1}{2\mu} \alpha L M_V r(N) + \frac{F(\lambda_1) - F(\lambda^*)}{\alpha \mu T},$$

where λ_t is iteratively defined in (6). Consequently,

$$\lim_{T \rightarrow \infty} \frac{\sum_{t=1}^T \mathbb{E}\|\nabla F(\lambda_t)\|_2^2}{T} \leq \frac{1}{2\mu} \alpha L M_V r(N). \quad (7)$$

Equation (7) underlines the dependence of the sum of the norm of the gradients on the variance of the gradients. The better the gradients are estimated, the closer one gets to the optimum where the gradient vanishes. As the dependence on the sample size becomes $O(N^{-2})$ for an RQMC sequence instead of $1/N$ for a MC sequence, the gradient is more precisely estimated for N large enough.

We now study the impact of a reduced variance on SGD with a fixed step size and strongly convex functions. We obtain an improved upper bound on the optimality gap.

Theorem 2 *Let F have Lipschitz continuous derivatives and be a strongly convex function, i.e. there exists a constant $c > 0$ s.t. $\forall \lambda, \bar{\lambda} F(\bar{\lambda}) \geq F(\lambda) + \nabla F(\lambda)^T (\bar{\lambda} - \lambda) + \frac{1}{2}c\|\lambda - \bar{\lambda}\|_2^2$, let $U_N = \{\mathbf{u}_1, \dots, \mathbf{u}_N\}$ be an RQMC sequence and let $\forall \lambda, G : \mathbf{u} \mapsto g_{\Gamma(\mathbf{u})}(\lambda)$ be as in Theorem 1. Let the constant learning rate $\alpha < \frac{1}{2c}$ and $\alpha < \frac{2}{L}$. Then the expected optimality gap satisfies, $\forall t \geq 0$,*

$$\begin{aligned} \mathbb{E}[F(\lambda_{t+1}) - F(\lambda^*)] &\leq \left[\left(\frac{\alpha^2 L}{2} - \alpha \right) 2c + 1 \right] \times \mathbb{E}[F_N(\lambda_t) - F(\lambda^*)] \\ &\quad + \frac{1}{2} L \alpha^2 [M_V r(N)]. \end{aligned}$$

Consequently,

$$\lim_{T \rightarrow \infty} \mathbb{E}[F(\lambda_T) - F(\lambda^*)] \leq \frac{\alpha L}{4c - \alpha L c} [M_V r(N)].$$

The previous result has the following interpretation. The expected optimality gap between the last iteration λ_T and the true minimizer λ^* is upper bounded by the magnitude of the variance. The smaller this variance, the closer we get to λ^* . Using RQMC we gain a factor $1/N$ in the bound.

Increasing Sample Size Over Iterations While SGD with a fixed step size and a fixed number of samples per gradient step does not converge, convergence can be achieved when increasing the number of samples used for estimating the gradient over iterations. As an extension of Theorem 2, we show that a linear convergence is obtained while increasing the sample size at a slower rate than for MC sampling.

Theorem 3 *Assume the conditions of Theorem 2 with the modification $\alpha \leq \min\{1/c, 1/L\}$. Let $1 - \alpha c/2 < \xi^2 = \frac{1}{\tau^2} < 1$. Use an increasing sample size $N_t = \underline{N} + \lceil \tau^t \rceil$, where $\underline{N} < \infty$ is defined in Appendix B.3. Then $\forall t \in \mathbb{N}$, $\exists \hat{M}_V < \infty$,*

$$\text{tr Var}_{U_N}[\hat{g}_{N_t}(\lambda)] \leq \hat{M}_V \times \frac{1}{\tau^{2t}}$$

and

$$\mathbb{E}[F(\lambda_{t+1}) - F(\lambda^*)] \leq \omega \xi^{2t},$$

where $\omega = \max\{\alpha L \hat{M}_V / c, F(\lambda_1) - F(\lambda^*)\}$.

This result compares favorably with a standard result on the linear convergence of SGD with fixed step size and strongly convex functions (Bottou et al., 2016). For MC sampling one obtains a different constant $\tilde{\omega}$ and an upper bound with ξ^t and not ξ^{2t} . Thus, besides the constant factor, RQMC samples allow us to close the optimality gap faster for the same geometric increase in the sample size τ^t or to use $\tau^{t/2}$ to obtain the same linear rate of convergence as MC based estimators.

Other Remarks The reduced variance in the estimation of the gradients should allow us to make larger moves in the parameter space. This is for example achieved by using adaptive step size algorithms as Adam (Kingma and Ba, 2015), or Adagrad (Duchi et al., 2011). However, the theoretical analysis of these algorithms is beyond the scope of this paper.

Also, note that it is possible to relax the smoothness assumptions on G while supposing only square integrability. Then one obtains rates in $o(N^{-1})$. Thus, RQMC yields always a faster rate than MC, regardless of the smoothness. See Appendix A for more details.

In the previous analysis, we have assumed that the entire randomness in the gradient estimator comes from the sampling of the variational distribution. In practice, additional randomness is introduced in the gradient via mini batch sampling. This leads to a dominating term in the variance of $O(K^{-1})$ for mini batches of size K . Still, the part of the variance related to the variational family itself is reduced and so is the variance of the gradient estimator as a whole.

4. Experiments

We study the effectiveness of our method in three different settings: a hierarchical linear regression, a multi-level Poisson generalized linear model (GLM) and a Bayesian neural network (BNN). Finally, we confirm the result of Theorem 3, which proposes to increase the sample size over iterations in QMCVI for faster asymptotic convergence.

Setup In the first three experiments we optimize the ELBO using the Adam optimizer (Kingma and Ba, 2015) with the initial step size set to 0.1, unless otherwise stated. The RQMC sequences are generated through a python interface to the R package `randtoolbox` (Christophe and Petr, 2015). In particular we use scrambled Sobol sequences. The gradients are calculated using an automatic differentiation toolbox. The ELBO values are computed by using 10,000 MC samples, the variance of the gradient estimators is estimated by resampling the gradient 1000 times in each optimization step and computing the empirical variance.

Benchmarks The first benchmark is the vanilla MCVI algorithm based on ordinary MC sampling. Our method QMCVI replaces the MC samples by RQMC sequences and comes at almost no computational overhead (Section 3).

Our second benchmark in the second and third experiment is the control variate (cv) approach of Miller et al. (2017), where we use the code provided with the publication. In the first experiment, this comparison is omitted since the method of Miller et al. (2017) does not apply in this setting due to the non-Gaussian variational distribution.

Main Results We find that our approach generally leads to a faster convergence compared to our baselines due to a decreased gradient variance. For the multi-level Poisson GLM experiment, we also find that our RQMC algorithm converges to a better local optimum of the ELBO. As proposed in Theorem 3, we find that increasing the sample size over iteration in QMCVI leads to a better asymptotic convergence rate than in MCVI.

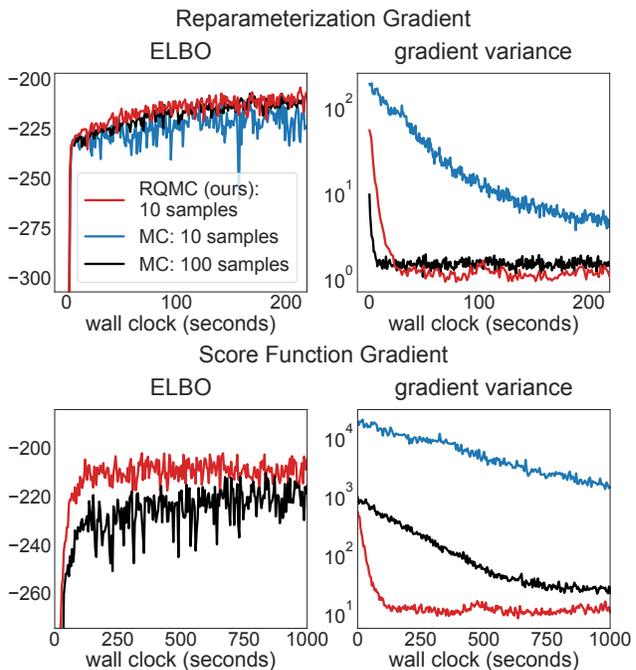


Figure 2: Toy: experiment 4.1. ELBO optimization path using Adam and variance of the stochastic gradient using the RQMC based gradient estimator using 10 samples (ours, in red) and the MC based estimator using 10 samples (blue) and 100 samples (black), respectively. The upper panel corresponds to the reparameterization gradient and the lower panel to the score function gradient estimator¹. For both versions of MCVI, using RQMC samples (proposed) leads to variance reduction and faster convergence.

4.1. Hierarchical Linear Regression

We begin the experiments with a toy model of hierarchical linear regression with simulated data. The sampling process for the outputs y_i is $y_i \sim \mathcal{N}(\mathbf{x}_i^T \mathbf{b}_i, \epsilon)$, $\mathbf{b}_i \sim \mathcal{N}(\mu_\beta, \sigma_\beta)$. We place lognormal hyper priors on the variance of the intercepts σ_β and on the noise ϵ ; and a Gaussian hyper prior on μ_β . Details on the model are provided in Appendix C.1. We set the dimension of the data points to be 10 and simulated 100 data points from the model. This results in a 1012-

¹Using only 10 samples for the MC based score function estimator leads to divergence and the ELBO values are out of the scope of the plot.

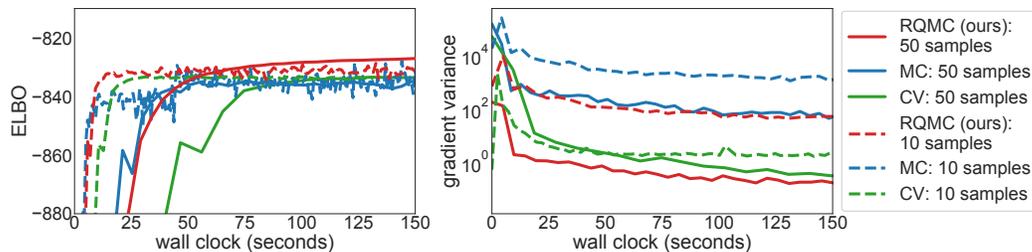


Figure 3: Frisk: experiment 4.2. Left, the optimization path of the ELBO is shown using Adam with the RQMC, MC and CV based reparameterization gradient estimator, respectively. Right, the gradient variances as function of time are reported. In the case of using 10 samples (dashed lines) RQMC (ours) outperforms the baselines in terms of speed while the CV based method exhibits lowest gradient variance. When increasing the sample size to 50 (solid lines) for all methods, RQMC converges closer to the optimum than the baselines while having lowest gradient variance.

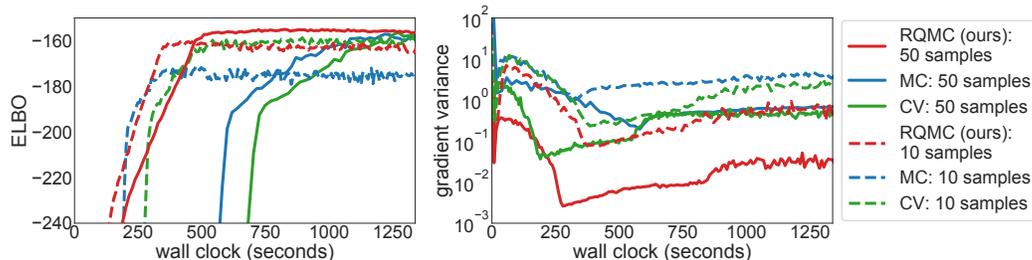


Figure 4: BNN: experiment 4.3. Left, the optimization path of the ELBO is shown using Adam with the RQMC, MC and CV based reparameterization gradient estimator, respectively. Right, the gradient variances as function of time are reported. RQMC (ours) based on 10 samples outperforms the baselines in terms of speed. RQMC with 50 samples is bit slower but converges closer to the optimum as its gradient variance is up to 3 orders of magnitude lower than for the baselines.

dimensional posterior, which we approximate by a variational distribution that mirrors the prior distributions.

We optimize the ELBO using Adam (Kingma and Ba, 2015) based on the score function as well as the reparameterization gradient estimator. We compare the standard MC based approach using 10 and 100 samples with our RQMC based approach using 10 samples, respectively. The CV based estimator cannot be used in this setting since it only supports Gaussian variational distributions and the variational family includes a lognormal distribution. For the score function estimator, we set the initial step size of Adam to 0.01.

The results are shown in Figure 2. We find that using RQMC samples decreases the variance of the gradient estimator substantially. This applies both to the score function and the reparameterization gradient estimator. Our approach substantially improves the standard score function estimator in terms of convergence speed and leads to a decreased gradient variance of up to three orders of magnitude. Our approach is also beneficial in the case of the reparameterization gradient estimator, as it allows for reducing the sample size from 100 MC samples to 10 RQMC samples, yielding a similar gradient variance and optimization speed.

4.2. Multi-level Poisson GLM

We use a multi-level Poisson generalized linear model (GLM), as introduced in (Gelman and Hill, 2006) as an example of multi-level modeling. This model has a 37-dim posterior, resulting from its hierarchical structure.

As in (Miller et al., 2017), we apply this model to the *frisk* data set (Gelman et al., 2006) that contains information on the number of stop-and-frisk events within different ethnicity groups. The generative process of the model is described in Appendix C.2. We approximate the posterior by a diagonal Gaussian variational distribution.

The results are shown in Figure 3. When using a small number of samples ($N = 10$), all three methods have comparable convergence speed and attain a similar optimum. In this setting, the CV based method has lowest gradient variance. When increasing the sample size to 50, our proposed RQMC approach leads to substantially decreased gradient variance and allows Adam to convergence closer to the optimum than the baselines. This agrees with the fact that RQMC improves over MC for sufficiently large sample sizes.

4.3. Bayesian Neural Network

As a third example, we study QMCVI and its baselines in the context of a Bayesian neural network. The network consists of a 50-unit hidden layer with ReLU activations. We place a normal prior over each weight, and each weight prior has an inverse Gamma hyper prior. We also place an inverse Gamma prior over the observation variance. The model exhibits a posterior of dimension $d = 653$ and is applied to a 100-row subsample of the wine dataset from the UCI repository². The generative process is described in Appendix C.3. We approximate the posterior by a variational diagonal Gaussian.

The results are shown in Figure 4. For $N = 10$, both the RQMC and the cv version converge to a comparable value of the ELBO, whereas the ordinary MC approach converges to a lower value. For $N = 50$, all three algorithms reach approximately the same value of the ELBO, but our RQMC method converges much faster. In both settings, the variance of the RQMC gradient estimator is one to three orders of magnitude lower than the variance of the baselines.

4.4. Increasing the Sample Size Over Iterations

Along with our new Monte Carlo variational inference approach QMCVI, Theorem 3 gives rise to a new stochastic optimization algorithm for Monte Carlo objectives. Here, we investigate this algorithm empirically, using a constant learning rate and an (exponentially) increasing sample size schedule. We show that, for strongly convex objective functions and some mild regularity assumptions, our RQMC based gradient estimator leads to a faster asymptotic convergence rate than using the ordinary MC based gradient estimator.

In our experiment, we consider a two-dimensional factorizing normal target distribution with zero mean and standard deviation one. Our variational distribution is also a normal distribution with fixed standard deviation of 1, and with a variational mean parameter, i.e., we only optimize the mean parameter. In this simple setting, the ELBO is strongly convex and the variational family includes the target distribution. We optimize the ELBO with an increasing sample size, using the SGD algorithm described in Theorem 3. We initialize the variational parameter to $(0.1, 0.1)$. Results are shown in Figure 5.

We considered both RQMC (red) and MC (blue) based gradient estimators. We plot the difference between the optimal ELBO value and the optimization trace in logarithmic scale. The experiment confirms the theoretical result of Theorem 3 as our RQMC based method attains a faster asymptotic convergence rate than the ordinary MC based approach. This means that, in the absence of additional noise due to data

²<https://archive.ics.uci.edu/ml/datasets/Wine+Quality>

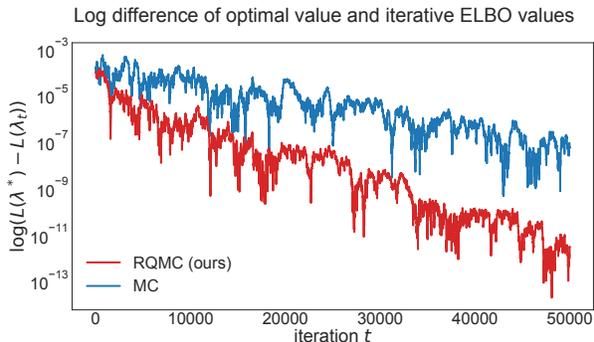


Figure 5: Constant SGD: experiment 4.4. We exemplify the consequences of Theorem 3 and optimize a simple concave ELBO using SGD with fixed learning rate $\alpha = 0.001$ while the number of samples are iteratively increased. We use an exponential sample size schedule (starting with one sample and 50,000 samples in the final iteration). The logarithmic difference of the ELBO to the optimum is plotted. We empirically confirm the result of Theorem 3 and observe a faster asymptotic convergence rate when using RQMC samples over MC samples.

subsampling, optimizing Monte Carlo objectives with RQMC can drastically outperform SGD.

5. Conclusion

We investigated randomized Quasi-Monte Carlo (RQMC) for stochastic optimization of Monte Carlo objectives. We termed our method Quasi-Monte Carlo Variational Inference (QMCVI), currently focusing on variational inference applications. Using our method, we showed that we can achieve faster convergence due to variance reduction.

QMCVI has strong theoretical guarantees and provably gets us closer to the optimum of the stochastic objective. Furthermore, in absence of additional sources of noise such as data subsampling noise, QMCVI converges at a faster rate than SGD when increasing the sample size over iterations.

QMCVI can be easily integrated into automated inference packages. All one needs to do is replace a sequence of uniform random numbers over the hypercube by an RQMC sequence, and perform the necessary reparameterizations to sample from the target distributions.

An open question remains as to which degree QMCVI can be combined with control variates, as RQMC may introduce additional unwanted correlations between the gradient and the cv. We will leave this aspect for future studies. We see particular potential for QMCVI in the context of reinforcement learning, which we consider to investigate.

Acknowledgements

We would like to thank Pierre E. Jacob, Nicolas Chopin, Rajesh Ranganath, Jaan Altsaar and Marius Kloft for their valuable feedback on our manuscript. This work was partly funded by the German Research Foundation (DFG) award KL 2698/2-1 and a GENES doctoral research scholarship.

References

- Antonov, I. A. and Saleev, V. (1979). An economic method of computing LP τ -sequences. *USSR Computational Mathematics and Mathematical Physics*, 19(1):252–256.
- Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.
- Bottou, L., Curtis, F. E., and Nocedal, J. (2016). Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*.
- Buchholz, A. and Chopin, N. (2017). Improving approximate Bayesian computation via quasi Monte Carlo. *arXiv preprint arXiv:1710.01057*.
- Burda, Y., Grosse, R., and Salakhutdinov, R. (2016). Importance weighted autoencoders. *Proceedings of the International Conference on Learning Representations*.
- Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software, Articles*, 76(1):1–32.
- Christophe, D. and Petr, S. (2015). *randtoolbox: Generating and Testing Random Numbers*. R package version 1.17.
- Defazio, A., Bach, F., and Lacoste-Julien, S. (2014). Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654.
- Dick, J., Kuo, F. Y., and Sloan, I. H. (2013). High-dimensional integration: The quasi-monte carlo way. *Acta Numerica*, 22:133–288.
- Drew, S. S. and Homem-de Mello, T. (2006). Quasi-monte carlo strategies for stochastic optimization. In *Proceedings of the 38th conference on Winter simulation*, pages 774–782. Winter Simulation Conference.
- Drovandi, C. C. and Tran, M.-N. (2018). Improving the Efficiency of Fully Bayesian Optimal Design of Experiments Using Randomised Quasi-Monte Carlo. *Bayesian Anal.*, 13(1):139–162.
- Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.
- Gelman, A. and Hill, J. (2006). *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.
- Gelman, A., Kiss, A., and Fagan, J. (2006). An analysis of the nypd’s stop-and-frisk policy in the context of claims of racial bias. *Columbia Public Law & Legal Theory Working Papers*, page 0595.
- Gerber, M. (2015). On integration methods based on scrambled nets of arbitrary size. *Journal of Complexity*, 31(6):798–816.
- Gerber, M. and Bornn, L. (2017). Improving simulated annealing through derandomization. *Journal of Global Optimization*, 68(1):189–217.
- Gerber, M. and Chopin, N. (2015). Sequential quasi monte carlo. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(3):509–579.
- Glasserman, P. (2013). *Monte Carlo methods in financial engineering*, volume 53. Springer Science & Business Media.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.
- Halton, J. H. (1964). Algorithm 247: Radical-inverse quasi-random point sequence. *Communications of the ACM*, 7(12):701–702.
- Hardy, G. H. (1905). On double *Fourier* series, and especially those which represent the double zeta-function with real and incommensurable parameters. *Quart. J.*, 37:53–79.
- Hickernell, F. J. (2006). *Koksma-Hlawka Inequality*. American Cancer Society.
- Hickernell, F. J., Lemieux, C., Owen, A. B., et al. (2005). Control variates for quasi-monte carlo. *Statistical Science*, 20(1):1–31.
- Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.
- Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323.
- Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.
- Joy, C., Boyle, P. P., and Tan, K. S. (1996). Quasi-monte carlo methods in numerical finance. *Management Science*, 42(6):926–938.
- Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. *Proceedings of the International Conference on Learning Representations*.
- Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. *Proceedings of the International Conference on Learning Representations*.
- Kuipers, L. and Niederreiter, H. (2012). *Uniform distribution of sequences*. Courier Corporation.
- L’Ecuyer, P. and Lemieux, C. (2005). Recent advances in randomized quasi-Monte Carlo methods. In *Modeling uncertainty*, pages 419–474. Springer.

- Lemieux, C. and L'Ecuyer, P. (2001). On the use of quasi-monte carlo methods in computational finance. In *International Conference on Computational Science*, pages 607–616. Springer.
- Leobacher, G. and Pillichshammer, F. (2014). *Introduction to quasi-Monte Carlo integration and applications*. Springer.
- Mandt, S., Hoffman, M. D., and Blei, D. M. (2017). Stochastic Gradient Descent as Approximate Bayesian Inference. *Journal of Machine Learning Research*, 18(134):1–35.
- Miller, A. C., Foti, N., D'Amour, A., and Adams, R. P. (2017). Reducing reparameterization gradient variance. In *Advances in Neural Information Processing Systems*.
- Moulines, E. and Bach, F. R. (2011). Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*.
- Nesterov, Y. (2013). *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.
- Niederreiter, H. (1992). *Random number generation and quasi-Monte Carlo methods*. SIAM.
- Oates, C. and Girolami, M. (2016). Control functionals for quasi-monte carlo integration. In *Artificial Intelligence and Statistics*, pages 56–65.
- Oates, C. J., Girolami, M., and Chopin, N. (2017). Control functionals for Monte Carlo integration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):695–718.
- Owen, A. B. (1997). Scrambled Net Variance for Integrals of Smooth Functions. *The Annals of Statistics*, 25(4):1541–1562.
- Owen, A. B. et al. (2008). Local antithetic sampling with scrambled nets. *The Annals of Statistics*, 36(5):2319–2343.
- Paisley, J., Blei, D., and Jordan, M. (2012). Variational Bayesian inference with stochastic search. *International Conference on Machine Learning*.
- Ranganath, R., Gerrish, S., and Blei, D. M. (2014). Black box variational inference. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*.
- Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the International Conference on Machine Learning*.
- Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.
- Robert, C. and Casella, G. (2013). *Monte Carlo Statistical Methods*. Springer Science & Business Media.
- Roeder, G., Wu, Y., and Duvenaud, D. K. (2017). Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In *Advances in Neural Information Processing Systems*.
- Rosenblatt, M. (1952). Remarks on a multivariate transformation. *Ann. Math. Statist.*, 23(3):470–472.
- Ruiz, F. J. R., Titsias, M. K., and Blei, D. M. (2016a). Overdispersed black-box variational inference. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.
- Ruiz, F. R., Titsias, M., and Blei, D. (2016b). The generalized reparameterization gradient. In *Advances in Neural Information Processing Systems*.
- Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT press Cambridge.
- Tran, D., Kucukelbir, A., Dieng, A. B., Rudolph, M., Liang, D., and Blei, D. M. (2016). Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*.
- Tran, M.-N., Nott, D. J., and Kohn, R. (2017). Variational bayes with intractable likelihood. *Journal of Computational and Graphical Statistics*, 26(4):873–882.
- Yang, J., Sindhwani, V., Avron, H., and Mahoney, M. (2014). Quasi-monte carlo feature maps for shift-invariant kernels. In *International Conference on Machine Learning*, pages 485–493.
- Zhang, C., Butepage, J., Kjellstrom, H., and Mandt, S. (2017a). Advances in variational inference. *arXiv preprint arXiv:1711.05597*.
- Zhang, C., Kjellström, H., and Mandt, S. (2017b). Determinantal point processes for mini-batch diversification. In *Uncertainty in Artificial Intelligence, UAI 2017*.

A. Additional Information on QMC

We provide some background on QMC sequences that we estimate necessary for the understanding of our algorithm and our theoretical results.

Quasi-Monte Carlo (QMC) Low discrepancy sequences (also called Quasi Monte Carlo sequences), are used to approximate integrals over the $[0, 1]^d$ hyper-cube: $\mathbb{E}\psi(U) = \int_{[0,1]^d} \psi(u)du$, that is the expectation of the random variable $\psi(U)$, where $U \sim \mathcal{U}[0, 1]^d$, is a uniform distribution on $[0, 1]^d$. The basic Monte Carlo approximation of the integral is $\hat{I}_N := \frac{1}{N} \sum_{n=1}^N \psi(\mathbf{u}_n)$, where each $\mathbf{u}_n \sim \mathcal{U}[0, 1]^d$, independently. The error of this approximation is $\mathcal{O}(N^{-1})$, since $\text{Var}[\hat{I}_N] = \text{Var}[\psi(U)]/N$.

This basic approximation may be improved by replacing the random variables \mathbf{u}_n by a low-discrepancy sequence; that is, informally, a deterministic sequence that covers $[0, 1]^d$ more regularly. The error of this approximation is assessed by the Koksma-Hlawka inequality (Hickernell, 2006):

$$\left| \int_{[0,1]^d} \psi(\mathbf{u})d\mathbf{u} - \frac{1}{N} \sum_{n=1}^N \psi(\mathbf{u}_n) \right| \leq V(\psi)D^*(\mathbf{u}_{1:N}), \quad (8)$$

where $V(\psi)$ is the total variation in the sense of Hardy and Krause (Hardy, 1905). This quantity is closely linked to the smoothness of the function ψ . $D^*(\mathbf{u}_{1:N})$ is called the star discrepancy, that measures how well the sequence covers the target space.

The general notion of discrepancy of a given sequence $\mathbf{u}_1, \dots, \mathbf{u}_N$ is defined as follows:

$$D(\mathbf{u}_{1:N}, \mathcal{A}) := \sup_{A \in \mathcal{A}} \left| \frac{1}{N} \sum_{n=1}^N \mathbb{1}\{\mathbf{u}_n \in A\} - \lambda_d(A) \right|,$$

where $\lambda_d(A)$ is the volume (Lebesgue measure on \mathbb{R}^d) of A and \mathcal{A} is a set of measurable sets. When we fix the sets $A = [0, \mathbf{b}] = \prod_{i=1}^d [0, b_i]$ with $0 \leq b_i \leq 1$ as a products set of intervals anchored at 0, the star discrepancy is then defined as follows

$$D^*(\mathbf{u}_{1:N}) := \sup_{[0, \mathbf{b}]} \left| \frac{1}{N} \sum_{n=1}^N \mathbb{1}\{\mathbf{u}_n \in [0, \mathbf{b}]\} - \lambda_d([0, \mathbf{b}]) \right|.$$

It is possible to construct sequences such that $D^*(\mathbf{u}_{1:N}) = \mathcal{O}((\log N)^{2d-2}/N^2)$. See also Kuipers and Niederreiter (2012) and Leobacher and Pillichshammer (2014) for more details.

Thus, QMC integration schemes are asymptotically more efficient than MC schemes. However, if the dimension d gets too large, the number of necessary samples N in order to reach the asymptotic regime becomes prohibitive. As the upper bound is rather pessimistic, in practice QMC integration outperforms MC integration even for small N in most applications, see e.g. the examples in Chapter 5 of Glasserman (2013). Popular QMC sequences are for example the Halton sequence or the Sobol sequence. See e.g. Dick et al. (2013) for details on the construction. A drawback of QMC is that it is difficult to assess the error and that the deterministic approximation is inherently biased.

Randomized Quasi Monte Carlo (RQMC). The reintroduction of randomness in a low discrepancy sequence while preserving the low discrepancy properties enables the construction of confidence intervals by repeated simulation. Moreover,

MC	QMC	RQMC
N^{-1}	$N^{-2}(\log N)^{2d-2}$	N^{-2}

Table 1: Best achievable rates for MC, QMC and RQMC in terms of the MSE of the approximation.

the randomization makes the approximation unbiased. The simplest method for this purpose is a randomly shifted sequence. Let $v \sim \mathcal{U}[0, 1]^d$. Then the sequence based on $\hat{\mathbf{u}}_n := \mathbf{u}_n + v \bmod 1$ preserves the properties of the QMC sequence with probability 1 and is marginally uniformly distributed.

Scrambled nets (Owen, 1997) represent a more sophisticated approach. Assuming smoothness of the derivatives of the function, Gerber (2015) showed recently, that rates of $\mathcal{O}(N^{-2})$ are achievable. We summarize the best rates in table 1.

Transforming QMC and RQMC sequences A generic recipe for using QMC / RQMC for integration is given by transforming a sequence with the inverse Rosenblatt transformation $\Gamma : \mathbf{u} \in [0, 1]^d \mapsto \mathbf{z} \in \mathbb{R}^d$, see Rosenblatt (1952) and Gerber and Chopin (2015), such that

$$\int \psi(\Gamma(\mathbf{u}))d\mathbf{u} = \int \psi(\mathbf{z})p(\mathbf{z})d\mathbf{z},$$

where $p(\mathbf{z})$ is the respective measure of integration. The inverse Rosenblatt transformation can be understood as the multivariate extension of the inverse cdf transform. For the procedure to be correct we have to make sure that $\psi \circ \Gamma$ is sufficiently regular.

Theoretical Results on RQMC In our analysis we mainly use the following result.

Theorem 4 (Owen et al., 2008) *Let $\psi : [0, 1]^d \rightarrow \mathbb{R}$ be a function such that its cross partial derivatives up to order d exist and are continuous, and let $(\mathbf{u}_n)_{n \in 1:N}$ be a relaxed scrambled (α, s, m, d) -net in base b with dimension d with uniformly bounded gain coefficients. Then,*

$$\text{Var} \left(\frac{1}{N} \sum_{n=1}^N \psi(\mathbf{u}_n) \right) = \mathcal{O}(N^{-3} \log(N)^{(d-1)}),$$

where $N = \alpha b^m$.

In words, $\forall \tau > 0$ the RQMC error rate is $\mathcal{O}(N^{-3+\tau})$ when a scrambled (α, s, m, d) -net is used. However, a more general result has recently been shown by Gerber (2015)[Corollary 1], where if ψ is square integrable and $(\mathbf{u}_n)_{n \in 1:N}$ is a scrambled (s, d) -sequence, then

$$\text{Var} \left(\frac{1}{N} \sum_{n=1}^N \psi(\mathbf{u}_n) \right) = o(N^{-1}).$$

This result shows that RQMC integration is always better than MC integration. Moreover, Gerber (2015)[Proposition 1] shows that rates $\mathcal{O}(N^{-2})$ can be obtained when the function ψ is regular in the sense of Theorem 4. In particular one gets

$$\text{Var} \left(\frac{1}{N} \sum_{n=1}^N \psi(\mathbf{u}_n) \right) = \mathcal{O}(N^{-2}).$$

B. Proofs

Our proof are deduced from standard results in the stochastic approximation literature, e.g. [Bottou et al. \(2016\)](#), when the variance of the gradient estimator is reduced due to RQMC sampling. Our proofs rely on scrambled (s, d) -sequences in order to use the result of [Gerber \(2015\)](#). The scrambled Sobol sequence, that we use in our simulations satisfies the required properties. We denote by \mathbb{E} the total expectation and by $\mathbb{E}_{U_{N,t}}$ the expectation with respect to the RQMC sequence U_N generated at time t . Note, that λ_t is not a random variable w.r.t. $U_{N,t}$ as it only depends on all the previous $U_{N,1}, \dots, U_{N,t-1}$ due to the update equation in (6). However, $\hat{F}_N(\lambda_t)$ is a random variable depending on $U_{N,t}$.

B.1. Proof of Theorem 1

Let us first prove that $\text{tr Var}[\hat{g}_N(\lambda)] \leq M_V \times r(N)$ for all λ . By assumption we have that $g_z(\lambda)$ with $z = \Gamma(\mathbf{u})$ is a function $G : \mathbf{u} \mapsto g_{\Gamma(\mathbf{u})}(\lambda)$ with continuous mixed partial derivatives of up to order d for all λ . Therefore, if $\mathbf{u}_1, \dots, \mathbf{u}_N$ is a RQMC sequence, then the trace of the variance of the estimator $\hat{g}_N(\lambda)$ is upper bounded by Theorem 4 and its extension by [Gerber \(2015\)](#)[Proposition 1] by a uniform bound M_V and the quantity $r(N) = \mathcal{O}(N^{-2})$, that goes to 0 faster than the Monte Carlo rate $1/N$.

By the Lipschitz assumption we have that $F(\lambda) \leq F(\bar{\lambda}) + \nabla F(\bar{\lambda})^T(\lambda - \bar{\lambda}) + \frac{1}{2}L\|\lambda - \bar{\lambda}\|_2^2, \forall \lambda, \bar{\lambda}$, see for example [Bottou et al. \(2016\)](#). By using the fact that $\lambda_{t+1} - \lambda_t = -\alpha \hat{g}_N(\lambda_t)$ we obtain

$$\begin{aligned} & F(\lambda_{t+1}) - F(\lambda_t) \\ & \leq \nabla F(\lambda_t)^T(\lambda_{t+1} - \lambda_t) + \frac{1}{2}L\|\lambda_{t+1} - \lambda_t\|_2^2, \\ & = -\alpha \nabla F(\lambda_t)^T \hat{g}_N(\lambda_t) + \frac{\alpha^2 L}{2} \|\hat{g}_N(\lambda_t)\|_2^2. \end{aligned}$$

After taking expectations with respect to $U_{N,t}$ we obtain

$$\begin{aligned} & \mathbb{E}_{U_{N,t}} F(\lambda_{t+1}) - F(\lambda_t) \\ & \leq -\alpha \nabla F(\lambda_t) \mathbb{E}_{U_{N,t}} \hat{g}_N(\lambda_t) + \frac{\alpha^2 L}{2} \mathbb{E}_{U_{N,t}} \|\hat{g}_N(\lambda_t)\|_2^2. \end{aligned}$$

We now use the fact that $\mathbb{E}_{U_{N,t}} \|\hat{g}_N(\lambda_t)\|_2^2 = \text{tr Var}_{U_{N,t}}[\hat{g}_N(\lambda_t)] + \|\mathbb{E}_{U_{N,t}} \hat{g}_N(\lambda_t)\|_2^2$ and after exploiting the fact that $\mathbb{E}_{U_{N,t}} \hat{g}_N(\lambda_t) = \nabla F(\lambda_t)$ we obtain

$$\begin{aligned} & \mathbb{E}_{U_{N,t}} F(\lambda_{t+1}) - F(\lambda_t) \\ & \leq -\alpha \|\nabla F(\lambda_t)\|_2^2 + \frac{\alpha^2 L}{2} \left[\text{tr Var}_{U_{N,t}}[\hat{g}_N(\lambda_t)] + \|\nabla F(\lambda_t)\|_2^2 \right], \\ & = \frac{\alpha^2 L}{2} \text{tr Var}_{U_{N,t}}[\hat{g}_N(\lambda_t)] + \left(\frac{\alpha^2 L}{2} - \alpha \right) \|\nabla F(\lambda_t)\|_2^2. \end{aligned}$$

The inequality is now summed for $t = 1, \dots, T$ and we take the total expectation:

$$\begin{aligned} & \mathbb{E} F(\lambda_T) - F(\lambda_1) \\ & \leq \frac{\alpha^2 L}{2} \sum_{t=1}^T \mathbb{E} \text{tr Var}_{U_{N,t}}[\hat{g}_N(\lambda_t)] \\ & \quad + \left(\frac{\alpha^2 L}{2} - \alpha \right) \sum_{t=1}^T \mathbb{E} \|\nabla F(\lambda_t)\|_2^2. \end{aligned}$$

We use the fact that $F(\lambda^*) - F(\lambda_1) \leq \mathbb{E} F(\lambda_T) - F(\lambda_1)$, where λ_1 is deterministic and λ^* is the true minimizer, and divide the inequality by T :

$$\begin{aligned} & \frac{1}{T} [F(\lambda^*) - F(\lambda_1)] \\ & \leq \frac{\alpha^2 L}{2} \frac{1}{T} \sum_{t=1}^T \mathbb{E} \text{tr Var}_{U_{N,t}}[\hat{g}_N(\lambda_t)] \\ & \quad + \left(\frac{\alpha^2 L}{2} - \alpha \right) \frac{1}{T} \sum_{t=1}^T \mathbb{E} \|\nabla F(\lambda_t)\|_2^2. \end{aligned}$$

By rearranging and using $\alpha < 2/L$ and $\mu = 1 - \alpha L/2$ we obtain

$$\begin{aligned} & \frac{1}{T} \sum_{t=1}^T \mathbb{E} \|\nabla F(\lambda_t)\|_2^2 \\ & \leq \frac{1}{T\alpha\mu} [F(\lambda_1) - F(\lambda^*)] \\ & \quad + \frac{\alpha L}{2\mu} \frac{1}{T} \sum_{t=1}^T \mathbb{E} \text{tr Var}_{U_{N,t}}[\hat{g}_N(\lambda_t)]. \end{aligned}$$

We now use $\text{tr Var}_{U_{N,t}}[\hat{g}_N(\lambda_t)] \leq M_V r(N)$ for all t . Equation (7) is obtained as $T \rightarrow \infty$.

B.2. Proof of Theorem 2

A direct consequence of strong convexity is the fact that the optimality gap can be upper bounded by the gradient in the current point λ , e.g. $2c(F(\lambda) - F(\lambda^*)) \leq \|\nabla F(\lambda)\|_2^2, \forall \lambda$. The following proof uses this result. Based on the previous proof we get

$$\begin{aligned} & \mathbb{E}_{U_{N,t}} F(\lambda_{t+1}) - F(\lambda_t) \\ & \leq \frac{\alpha^2 L}{2} \text{tr Var}_{U_{N,t}}[\hat{g}_N(\lambda_t)] \\ & \quad + \left(\frac{\alpha^2 L}{2} - \alpha \right) \|\nabla F(\lambda_t)\|_2^2 \\ & \leq \frac{\alpha^2 L}{2} \text{tr Var}_{U_{N,t}}[\hat{g}_N(\lambda_t)] \\ & \quad + \left(\frac{\alpha^2 L}{2} - \alpha \right) 2c(F(\lambda_t) - F(\lambda^*)), \end{aligned}$$

where we have used that $\left(\frac{\alpha L}{2} - 1\right) \leq 0$. By subtracting $F(\lambda^*)$ from both sides, taking total expectations and rearranging we obtain:

$$\begin{aligned} & \mathbb{E} F(\lambda_{t+1}) - F(\lambda^*) \\ & \leq \frac{\alpha^2 L}{2} \mathbb{E} \text{tr Var}_{U_{N,t}}[\hat{g}_N(\lambda_t)] \\ & \quad + \left[\left(\frac{\alpha^2 L}{2} - \alpha \right) 2c + 1 \right] (\mathbb{E} F(\lambda_t) - F(\lambda^*)). \end{aligned}$$

Define $\beta = \left[\left(\frac{\alpha^2 L}{2} - \alpha \right) 2c + 1 \right]$. We add

$$\frac{\alpha^2 L \mathbb{E} \text{tr Var}_{U_{N,t}}[\hat{g}_N(\lambda_t)]}{2(\beta - 1)}$$

to both sides of the equation. This yields

$$\begin{aligned}
 & \mathbb{E}F(\lambda_{t+1}) - F(\lambda^*) + \frac{\alpha^2 L \mathbb{E} \operatorname{tr} \operatorname{Var}_{U_{N_t}} [\hat{g}_N(\lambda_t)]}{2(\beta - 1)} \\
 & \leq \frac{\alpha^2 L}{2} \mathbb{E} \operatorname{tr} \operatorname{Var}_{U_{N_t}} [\hat{g}_N(\lambda_t)] \\
 & \quad + \beta (\mathbb{E}F(\lambda_t) - F(\lambda^*)) + \frac{\alpha^2 L \mathbb{E} \operatorname{tr} \operatorname{Var}_{U_{N_t}} [\hat{g}_N(\lambda_t)]}{2(\beta - 1)} \\
 & \leq \beta \left(\mathbb{E}F(\lambda_t) - F(\lambda^*) + \frac{\alpha^2 L \mathbb{E} \operatorname{tr} \operatorname{Var}_{U_{N_t}} [\hat{g}_N(\lambda_t)]}{2(\beta - 1)} \right).
 \end{aligned}$$

Let us now show that $\beta < 1$:

$$\beta \leq \left[\left(\frac{\alpha L}{2} - 1 \right) 2\alpha c + 1 \right]$$

And as $\frac{\alpha L}{2} < 1$ we get $\beta < 1 - 2\alpha c$. Using $\alpha < 1/2c$ we obtain $\beta < 1$ and thus get a contracting equation when iterating over t :

$$\begin{aligned}
 & \mathbb{E}F(\lambda_{t+1}) - F(\lambda^*) \\
 & \leq \beta^t \left(F(\lambda_1) - F(\lambda^*) + \frac{\alpha^2 L \mathbb{E} \operatorname{tr} \operatorname{Var}_{U_{N_t}} [\hat{g}_N(\lambda_t)]}{2(\beta - 1)} \right) \\
 & \quad - \frac{\alpha^2 L \mathbb{E} \operatorname{tr} \operatorname{Var}_{U_{N_t}} [\hat{g}_N(\lambda_t)]}{2(\beta - 1)}, \\
 & \leq \beta^t \left(F(\lambda_1) - F(\lambda^*) + \frac{\alpha^2 L M_V r(N)}{2(\beta - 1)} \right) \\
 & \quad + \frac{\alpha^2 L M_V r(N)}{2(1 - \beta)}.
 \end{aligned}$$

After simplification we get

$$\begin{aligned}
 & \mathbb{E}F(\lambda_{t+1}) - F(\lambda^*) \\
 & \leq \beta^t \left(F(\lambda_1) - F(\lambda^*) + \frac{\alpha L}{2\alpha Lc - 4c} M_V r(N) \right) \\
 & \quad + \frac{\alpha L}{4c - 2\alpha Lc} M_V r(N),
 \end{aligned}$$

where the first term of the r.h.s. goes to 0 as $t \rightarrow \infty$.

B.3. Proof of Theorem 3

We require $\underline{N} \geq b^{s+d}$, due to a remark of Gerber (2015), where d is the dimension and b, s are integer parameters of the RQMC sequence. As $\mathbf{u} \mapsto g_{\Gamma(\omega)}(\lambda)$ has continuous mixed partial derivatives of order d for all λ , $\operatorname{tr} \operatorname{Var} [\hat{g}_{N_t}(\lambda)] = \mathcal{O}(1/N_t^2)$ and consequently $\operatorname{tr} \operatorname{Var} [\hat{g}_{N_t}(\lambda)] \leq \hat{M}_V \times (1/N_t^2)$, where \hat{M}_V is an universal upper bound on the variance. We recall that $N_t = \underline{N} + \lceil \tau^t \rceil$. Consequently

$$\operatorname{tr} \operatorname{Var} [\hat{g}_{N_t}(\lambda)] \leq \hat{M}_V \times \frac{1}{(\underline{N} + \lceil \tau^t \rceil)^2} \leq \hat{M}_V \times \frac{1}{\tau^{2t}}.$$

Now we take an intermediate result from the previous proof:

$$\begin{aligned}
 & \mathbb{E}_{U_{N_t}} F(\lambda_{t+1}) - F(\lambda_t) \\
 & \leq \frac{\alpha^2 L}{2} \operatorname{tr} \operatorname{Var}_{U_{N_t}} [\hat{g}_{N_t}(\lambda_t)] + \left(\frac{\alpha^2 L}{2} - \alpha \right) \|\nabla F(\lambda_t)\|_2^2 \\
 & \leq \frac{\alpha^2 L}{2} \operatorname{tr} \operatorname{Var}_{U_{N_t}} [\hat{g}_{N_t}(\lambda_t)] - \frac{\alpha}{2} \|\nabla F(\lambda_t)\|_2^2 \\
 & \leq \frac{\alpha^2 L}{2} \operatorname{tr} \operatorname{Var}_{U_{N_t}} [\hat{g}_{N_t}(\lambda_t)] - \alpha c (F(\lambda_t) - F(\lambda^*)),
 \end{aligned}$$

where we have used $\alpha \leq \min\{1/c, 1/L\}$ as well as strong convexity. Adding $F(\lambda^*)$, rearranging and taking total expectations yields

$$\begin{aligned}
 & \mathbb{E}F(\lambda_{t+1}) - F(\lambda^*) \\
 & \leq \frac{\alpha^2 L}{2} \mathbb{E} \operatorname{tr} \operatorname{Var}_{U_{N_t}} [\hat{g}_{N_t}(\lambda_t)] \\
 & \quad + [1 - \alpha c] (\mathbb{E}F(\lambda_t) - F(\lambda^*)).
 \end{aligned}$$

We now use $\operatorname{tr} \operatorname{Var}_{U_{N_t}} [\hat{g}_{N_t}(\lambda_t)] \leq \hat{M}_V \xi^{2t}$ and get

$$\begin{aligned}
 & \mathbb{E}F(\lambda_{t+1}) - F(\lambda^*) \\
 & \leq \frac{\alpha^2 L}{2} \hat{M}_V \xi^{2t} + [1 - \alpha c] (\mathbb{E}F(\lambda_t) - F(\lambda^*)).
 \end{aligned}$$

We now use induction to prove the main result. The initialization for $t = 0$ holds true by the definition of ω . Then, for all $t \geq 1$,

$$\begin{aligned}
 & \mathbb{E}F(\lambda_{t+1}) - F(\lambda^*) \\
 & \leq [1 - \alpha c] \omega \xi^{2t} + \frac{\alpha^2 L}{2} \hat{M}_V \xi^{2t}, \\
 & \leq \omega \xi^{2t} \left(1 - \alpha c + \frac{\alpha^2 L \hat{M}_V}{2\omega} \right), \\
 & \leq \omega \xi^{2t} \left(1 - \alpha c + \frac{\alpha c}{2} \right), \\
 & \leq \omega \xi^{2t} \left(1 - \frac{\alpha c}{2} \right) \\
 & \leq \omega \xi^{2(t+1)},
 \end{aligned}$$

where we have used the definition of ω and that $(1 - \frac{\alpha c}{2}) \leq \xi^2$.

C. Details for the Models Considered in the Experiments

C.1. Hierarchical Linear Regression

The generative process of the hierarchical linear regression model is as follows.

$\mu_\beta \sim \mathcal{N}(0, 10^2)$	intercept hyper prior
$\sigma_\beta \sim \operatorname{LogNormal}(0.5)$	intercept hyper prior
$\epsilon \sim \operatorname{LogNormal}(0.5)$	noise
$\mathbf{b}_i \sim \mathcal{N}(\mu_\beta, \sigma_\beta)$	intercepts
$y_i \sim \mathcal{N}(\mathbf{x}_i^\top \mathbf{b}_i, \epsilon)$	output

The dimension of the parameter space is $d = I \times k + k + 2$, where k denotes the dimension of the data points x_i and I their number. We set $I = 100$ and $k = 10$. The dimension hence equals $d = 1012$.

C.2. Multi-level Poisson GLM

The generative process of the multi-level Poisson GLM is

$\mu \sim N(0, 10^2)$	mean offset
$\log \sigma_\alpha^2, \log \sigma_\beta^2 \sim N(0, 10^2)$	group variances
$\alpha_e \sim N(0, \sigma_\alpha^2)$	ethnicity effect
$\beta_p \sim N(0, \sigma_\beta^2)$	precinct effect
$\log \lambda_{ep} = \mu + \alpha_e + \beta_p + \log N_{ep}$	log rate
$Y_{ep} \sim \text{Poisson}(\lambda_{ep})$	stop-and-frisk events

Y_{ep} denotes the number of stop-and-frisk events within ethnicity group e and precinct p over some fixed period. N_{ep} represents the total number of arrests of group e in precinct p over the same period; α_e and β_p are the ethnicity and precinct effects.

C.3. Bayesian Neural Network

We study a Bayesian neural network which consists of a 50-unit hidden layer with ReLU activations.

The generative process is

$\alpha \sim \text{InvGamma}(1, 0.1)$	weight hyper prior
$\tau \sim \text{InvGamma}(1, 0.1)$	noise hyper prior
$w_i \sim N(0, 1/\alpha)$	weights
$y \sim N(\phi(\mathbf{x}, \mathbf{w}), 1/\tau)$	output distribution

Above, w is the set of weights, and $\phi(\mathbf{x}, \mathbf{w})$ is a multi-layer perceptron that maps input \mathbf{x} to output y as a function of parameters \mathbf{w} . We denote the set of parameters as $\theta := (\mathbf{w}, \alpha, \tau)$. The model exhibits a posterior of dimension $d = 653$.

D. Practical Advice for Implementing QMCVI in Your Code

It is easy to implement RQMC based stochastic optimization in your existing code. First, you have to look for all places where random samples are used. Then replace the ordinary random number generator by RQMC sampling. To replace an ordinarily sampled random variable \mathbf{z} by an RQMC sample we need a mapping Γ from a uniformly distributed random variable \mathbf{u} to $\mathbf{z} = \Gamma(\mathbf{u}|\lambda)$, where λ are the parameters of the distribution (e.g. mean and covariance parameter for a Gaussian random variable). Fortunately, such a mapping can often be found (see Appendix A).

In many recent machine learning models, such as variational auto encoders (Kingma and Ba, 2015) or generative adversarial networks (Goodfellow et al., 2014), the application of RQMC sampling is straightforward. In those models, all random variables are often expressed as transformations of Gaussian random variables via deep neural networks. To apply our proposed RQMC sampling approach we only have to replace the Gaussian random variables of the base distributions by RQMC sampled random variables.

In the following Python code snippet we show how to apply our proposed RQMC sampling approach in such settings.

```
import numpy.random as nr
import numpy as np
import rpy2.robjects.packages as rpackages
import rpy2.robjects as robjects
from scipy.stats import norm

randtoolbox = rpackages.importr('randtoolbox')

def random_sequence_rqmc(dim, i=0, n=1, random_seed=0):
    """
    generate uniform RQMC random sequence
    """
    dim = np.int(dim)
    n = np.int(n)
    u = np.array(randtoolbox.sobol(n=n, dim=dim, init=(
        i==0), scrambling=1, seed=random_seed)).reshape((n,
        dim))
    # randtoolbox for sobol sequence
    return(u)

def random_sequence_mc(dim, n=1, random_seed=0):
    """
    generate uniform MC random sequence
    """
    dim = np.int(dim)
    n = np.int(n)
    np.random.seed(seed=random_seed)
    u = np.asarray(nr.uniform(size=dim*n).reshape((n,
        dim)))
    return(u)

def transform_uniform_to_normal(u, mu, sigma):
    """
    generat a multivariate normal based on
    a unifrom sequence
    """
    l_cholesky = np.linalg.cholesky(sigma)
    epsilon = norm.ppf(u).transpose()
    res = np.transpose(l_cholesky.dot(epsilon))+mu
    return res

if __name__ == '__main__':
    # example in dimension 2
    dim = 2
    n = 1000
    mu = np.ones(dim)*2. # mean of the Gaussian
    sigma = np.array([[2.,1.],[1.,2.]]) # variance of
    the Gaussian

    # generate Gaussian random variables via RQMC
    u_random = random_sequence_rqmc(dim, i=0, n=n,
    random_seed=1)
    x_normal = transform_uniform_to_normal(u_random, mu
    , sigma)

    # here comes the existing code of your model
    deep_bayesian_model(x_normal)
```

Code 1: Python code for RQMC sampling from a Gaussian distribution.