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 $O(N^{-1})$, since $\mathrm{Var}[\hat{I}_N] = \mathrm{Var}[\psi(U)]/N$.

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):

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

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 $\mathbf{u}_1, \cdots, \mathbf{u}_N$ 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 $D^*(\mathbf{u}_{1:N}) = O((\log N)^{2d-2}/N^2)$. 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.

Ramdomized 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, 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 $\hat{\mathbf{u}}_n := \mathbf{u}_n + v \mod 1$ preserves the properties of the QMC sequence with probability 1 and is marginally uniformly distributed.

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.

Scrambled nets (Owen, 1997) represent a more sophisticated approach. Assuming smoothness of the derivatives of the function, Gerber (2015) showed recently, that rates of $O(N^{-2})$ 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 $\psi : [0,1]^d \to \mathbb{R}$ 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,*

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

where $N = \alpha b^m$.

In words, $\forall \tau > 0$ the RQMC error rate is $O(N^{-3+\tau})$ 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

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

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

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

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 $U_{N,1}, \cdots, U_{N,t-1}$ 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 $\operatorname{tr}\operatorname{Var}[\hat{g}_N(\lambda)] \leq M_V \times r(N)$ 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 $\mathbf{u}_1, \cdots, \mathbf{u}_N$ 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 $r(N) = O(N^{-2})$, that goes to 0 faster than the Monte Carlo rate $1/N$.

By the Lipschitz assumption we have that $F(\lambda) \leq F(\overline{\lambda}) + \nabla F(\overline{\lambda})^T(\lambda - \overline{\lambda}) + \frac{1}{2}L\|\lambda - \overline{\lambda}\|_2^2, \forall \lambda, \overline{\lambda}$, 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 $\mathbb{E}_{U_{N,t}}\|\hat{g}_N(\lambda_t)\|_2^2 = \operatorname{tr}\operatorname{Var}_{U_{N,t}}[\hat{g}_N(\lambda_t)] + \|\mathbb{E}_{U_{N,t}}\hat{g}_N(\lambda_t)\|_2^2$ and after exploiting the fact that $\mathbb{E}_{U_{N,t}}\hat{g}_N(\lambda_t) = \nabla F(\lambda_t)$ we obtain

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

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

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

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

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

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

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

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

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. $2c(F(\lambda) - F(\lambda^\star)) \leq \|\nabla F(\lambda)\|_2^2, \forall \lambda$. The following proof uses this result. Based on the previous proof we get

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

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

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

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

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

to both sides of the equation. This yields

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

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:

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

After simplification we get

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

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

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 $\mathbf{u} \mapsto g_{\Gamma(\mathbf{u})}(\lambda)$ has continuous mixed partial derivatives of order d for all λ, $\mathrm{tr}\,\mathrm{Var}\left[\hat{g}_{N_t}(\lambda)\right] = O(1/N_t^2)$ and consequently $\mathrm{tr}\,\mathrm{Var}\left[\hat{g}_{N_t}(\lambda)\right] \leq \hat{M}_V \times (1/N_t^2)$, where $\hat{M}_V$ is an universal upper bound on the variance. We recall that $N_t = \underline{N} + \lceil \tau^t \rceil$. Consequently

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

Now we take an intermediate result from the previous proof:

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

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

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

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

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

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

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

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

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
$\sigma_\beta \sim \mathrm{LogNormal}(0.5)$	intercept hyper prior
$\epsilon \sim \mathrm{LogNormal}(0.5)$	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

$\mu \sim \mathcal{N}(0, 10^2)$	mean offset
$\log \sigma_\alpha^2, \log \sigma_\beta^2 \sim \mathcal{N}(0, 10^2)$	group variances
$\alpha_e \sim \mathcal{N}(0, \sigma_\alpha^2)$	ethnicity effect
$\beta_p \sim \mathcal{N}(0, \sigma_\beta^2)$	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
$w_i \sim \mathcal{N}(0, 1/\alpha)$	weights
$y \sim \mathcal{N}(\phi(\mathbf{x}, \mathbf{w}), 1/\tau)$	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 transfrom_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 = 100
    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 = transfrom_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.