5th Symposium on Advances in Approximate Bayesian Inference, 2023 1–20

Sample Average Approximation for Black-Box VI

Javier Burroni JBURRONI@CS.UMASS.EDU
Justin Domke DOMKE@CS.UMASS.EDU
Daniel Sheldon SHELDON@CS.UMASS.EDU
University of Massachusetts Amherst

Abstract

We present a novel approach for black-box VI that bypasses the difficulties of stochastic gradient ascent, including the task of selecting step-sizes. Our approach involves using a sequence of sample average approximation (SAA) problems. SAA approximates the solution of stochastic optimization problems by transforming them into deterministic ones. We use quasi-Newton methods and line search to solve each deterministic optimization problem and present a heuristic policy to automate hyperparameter selection. Our experiments show that our method simplifies the VI problem and achieves faster performance than existing methods.

1. Introduction

Variational inference (VI) is a powerful technique in machine learning that allows us to approximate the posterior distribution of a latent variable given some observed data. This is done by formulating the problem as an optimization problem, where the objective is to find a distribution from a family of distributions that is as close as possible to the true distribution. To achieve this, VI maximizes the evidence lower bound (ELBO), which is a lower bound on the log-likelihood of the observed data (Wainwright et al., 2008; Jaakkola and Jordan, 1997; Beal, 2003).

One popular approach to solving VI problems is black-box VI, which uses stochastic gradient descent (SGD) to optimize the ELBO (Wingate and Weber, 2013; Ranganath et al., 2014). In this method, an unbiased estimator of the gradient of the ELBO is used for stochastic optimization. However, selecting an appropriate step-size for SGD can be challenging and have a significant impact on the outcome. Recent work by Agrawal et al. (2020) recommends performing a comprehensive search over step-sizes to avoid the suboptimality of using a previously-selected step-size. In practice, users often turn to adaptive methods like Adam (Kingma and Ba, 2015) or AdaGrad (Duchi et al., 2011) to adjust the step-size on-the-fly, but these methods also require tuning of hyperparameters, which can be time-consuming and error-prone.

We propose a robust alternative stochastic-optimization approach using sample average approximation (SAA), focusing on statistical models without data-subsampling. Our contributions include: (i) using SAA to solve VI problems, employing nonlinear optimization tools (Healy and Schruben, 1991; Robinson, 1996; Shapiro and Wardi, 1996; Kleywegt et al., 2002; Kim et al., 2015); (ii) applying quasi-Newton methods with line-search for efficient optimization; (iii) addressing Monte Carlo error with a sequence of SAAs with increasing sample sizes (Chen and Schmeiser, 2001); (iv) presenting the SAA for VI algorithm with default scheduling and stopping criteria, competitive in accuracy and computational cost, simplifying the VI process.

2. Background

We are interested in approximating the posterior distribution of a latent variable given some observed data, i.e., $p(Z \mid x)$, where Z is the latent variable and x is the observed data. To achieve this, we will approximate the posterior with a distribution from an indexed family of approximations $\mathcal{Q} = \{q_\theta \mid \theta \in \mathbb{R}^d\}$, where θ is a vector of parameters that parameterize the approximation $q_\theta(Z)$, and d is the dimension of θ.
VI proposes to approximate the posterior distribution by finding a member from $\mathcal{Q}$ that is closest in Kullback-Leibler divergence to the true distribution. This is achieved by maximizing the evidence lower bound (ELBO), which is a function of the parameters: $\mathcal{L}\colon \theta \mapsto \mathbb{E}_{q_\theta}[\ln p(Z,x) - \ln q_\theta(Z)]$. The optimization problem can be formulated as:

$$\max_{\theta\in\Theta} \mathcal{L}(\theta) = \max_{\theta\in\Theta} \mathbb{E}[\ln p(Z,x) - \ln q_\theta(Z)], \qquad Z \sim q_\theta. \tag{1}$$

Under smoothness assumptions, black-box VI presents this problem as a smooth stochastic optimization problem (SOP) and suggests solving it using methods based on stochastic gradient descent (SGD). Specifically, it uses stochastic gradient ascent to maximize the ELBO by updating the parameters as follows:
At every iteration, samples $z_1, \ldots, z_n$ from q_{θ_t} are drawn and the sample mean of the function $g_{\theta_t}(Z)$ is being computed, where $g_{\theta_t}(Z)$ is a $\mathbb{R}^d$-valued random vector whose expectation equals the gradient. Then, this estimate is used to update the parameters according to:

$$\theta_{t+1} = \theta_t + \gamma_t \frac{1}{n}\sum_{i=1}^{n} g_{\theta_t}(z_i) \qquad \text{for } t \in \mathbb{N}, \text{ and } \gamma_t \in \mathbb{R}+. \tag{2}$$

The function g_{θ_t} can be obtained using various methods, including the score function estimator (Wingate and Weber, 2013; Ranganath et al., 2014) or, if the distribution is reparameterizable, the 'reparameterization trick' (Kingma and Welling, 2013; Fu, 2006; Kingma et al., 2019; Rezende et al., 2014), among others. A random variable Z comes from a reparameterizable distribution q_θ if there exist a C^1 function g_θ and a density q_{base} such that $Z = g_\theta(\epsilon)$ for $\epsilon \sim q_{\text{base}}$. We refer to these ϵ values as noise. In such case, the stochastic optimization problem becomes

$$\max_{\theta\in\Theta} \mathcal{L}(\theta) = \max_{\theta\in\Theta} \mathbb{E}[\ln p(g_\theta(\epsilon), x) - \ln q_\theta(g_\theta(\epsilon))], \qquad \epsilon \sim q_{\text{base}}. \tag{3}$$

The explanation above simplifies complexities in choosing hyperparameters, particularly step size γ_t. Users can choose a schedule meeting Robbins-Monro conditions (Robbins and Monro, 1951), but specific sequences affect convergence speed differently, and line-search methods are impractical due to random loss-function estimation. Moreover, the choice of the number of samples n per iteration can impact optimization, as larger n provides a more accurate gradient estimate but may increase computational cost.

3. Methods

3.1. Sample Average Approximation

The problem of ELBO maximization in the reparameterization setting of Eq. (3) is formulated as an SOP where the stochasticity comes from a fixed probability distribution, i.e., a probability distribution which does not depend on θ. Furthermore, the function inside the expectation is a smooth function of the parameters θ. Solutions to these problems can be approximated using the *sample average approximation* (SAA): a sample average over a *fixed sample* replaces the expectation, effectively transforming the SOP into a deterministic optimization problem.

We propose to use SAA for black-box VI. To use SAA, we take n i.i.d. samples $\boldsymbol{\epsilon} = \epsilon_1, \ldots, \epsilon_n$ from the distribution q_{base} and define the deterministic *training objective* function $\hat{\mathcal{L}}_{\boldsymbol{\epsilon}} \colon \theta \mapsto \frac{1}{n}\sum_{i=1}^{n}[\ln p(z_\theta(\epsilon_i), x) - \ln q_\theta(z_\theta(\epsilon_i))]$, which is a function of θ alone.

Then, the optimization problem in Eq. (3) can be transformed into a deterministic optimization problem

$$\max_{\theta\in\Theta} \hat{\mathcal{L}}_{\boldsymbol{\epsilon}}(\theta) = \max_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n}[\ln p(z_\theta(\epsilon_i), x) - \ln q_\theta(z_\theta(\epsilon_i))] = \max_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} v_\theta(\epsilon_i), \tag{4}$$

where we introduced the *log-weights* $v_\theta(\epsilon_i) = \ln p(z_\theta(\epsilon_i), x) - \ln q_\theta(z_\theta(\epsilon_i))$, also known as *log-importance ratios*. As the optimization is performed with the fixed set $\boldsymbol{\epsilon}$, we refer to it as the training noise.

We want to recover the optimal parameters $\theta^*_{\boldsymbol{\epsilon}}$ of $\hat{\mathcal{L}}_{\boldsymbol{\epsilon}}$. In an unconstrained smooth optimization setting, we need to specify how to compute a search direction and a step size. For the search direction, we will use L-BFGS (Broyden, 1970; Fletcher, 2013; Goldfarb, 1970; Shanno, 1970; Nocedal, 1980). For a detailed description of the L-BFGS algorithm, refer to Nocedal and Wright (1999).

In contrast to the SGD setting, deterministic optimization allows us to specify the step size using line search and ask for it to satisfy the *strong Wolfe conditions* (Nocedal and Wright, 1999). This allows us to use a step size that is guaranteed to increase the objective function. We will use L-BFGS with line search to compute an approximation to the optimal value of Eq. (4), and denote the process that does so by $\text{Opt}(\theta, n, \boldsymbol{\epsilon}, \tau)$. Here, τ is the maximum number of iterations for which L-BFGS will run, and θ is an initial value of the parameters. Besides the arguments of $\hat{\mathcal{L}}_{\boldsymbol{\epsilon}}(\theta)$, we also need to specify the value of τ.

Detection and mitigation of overfitting. It is important to understand that the training objective $\hat{\mathcal{L}}_{\boldsymbol{\epsilon}}(\theta)$ and the ELBO $\mathcal{L}(\theta)$, may differ for a fixed θ. The ELBO is an expectation over the distribution q_θ, while the training objective is computed based on an average over a fixed sample $\boldsymbol{\epsilon}$. In contrast, the optimal ELBO refers to the value of the ELBO achieved by the maximizer θ^* of Eq. (1), i.e., $\mathcal{L}(\theta^*)$.

During optimization with a fixed sample of training noise $\boldsymbol{\epsilon}_n = \epsilon_1, \ldots, \epsilon_n$, one might wonder how much the learned parameters $\theta^*_{\boldsymbol{\epsilon}_n}$ and the distribution $q_{\theta^*_{\boldsymbol{\epsilon}_n}}$ depend on these noise samples. In particular, how this dependency translates into a gap between the ELBO $\mathcal{L}(\theta^*_{\boldsymbol{\epsilon}_n})$ and the optimal ELBO $\mathcal{L}(\theta^*)$. Fortunately, there are two results by Mak et al. (1999) that are relevant to our discussion. Note that until the noise variables $\epsilon_1, \ldots, \epsilon_n$ are realized, the quantities $\theta^*_{\boldsymbol{\epsilon}_n}$ and $\hat{\mathcal{L}}_{\boldsymbol{\epsilon}_n}(\theta^*_{\boldsymbol{\epsilon}_n})$ are random. Let $\hat{\boldsymbol{\epsilon}}_{n+1} = \hat{\epsilon}_1, \ldots, \hat{\epsilon}_{n+1}$, be a sample of size $n+1$ taken i.i.d. from q_{base}. Assuming that the optimization process converges to a global optimum, it holds in expectation that: (i) the ELBO and training objective sandwich the optimal ELBO, i.e., $\mathbb{E}\,\mathcal{L}(\theta^*_{\boldsymbol{\epsilon}_n}) \leq \mathcal{L}(\theta^*) \leq \mathbb{E}\,\hat{\mathcal{L}}_{\boldsymbol{\epsilon}_n}(\theta^*_{\boldsymbol{\epsilon}_n})$; and (ii) the training objective converges monotonically to the optimal ELBO from above, i.e., $\mathbb{E}\,\hat{\mathcal{L}}_{\hat{\boldsymbol{\epsilon}}_{n+1}}(\theta^*_{\hat{\boldsymbol{\epsilon}}_{n+1}}) \leq \mathbb{E}\,\hat{\mathcal{L}}_{\boldsymbol{\epsilon}_n}(\theta^*_{\boldsymbol{\epsilon}_n})$. These results mean that we can use statistical techniques to quantify the discrepancy between

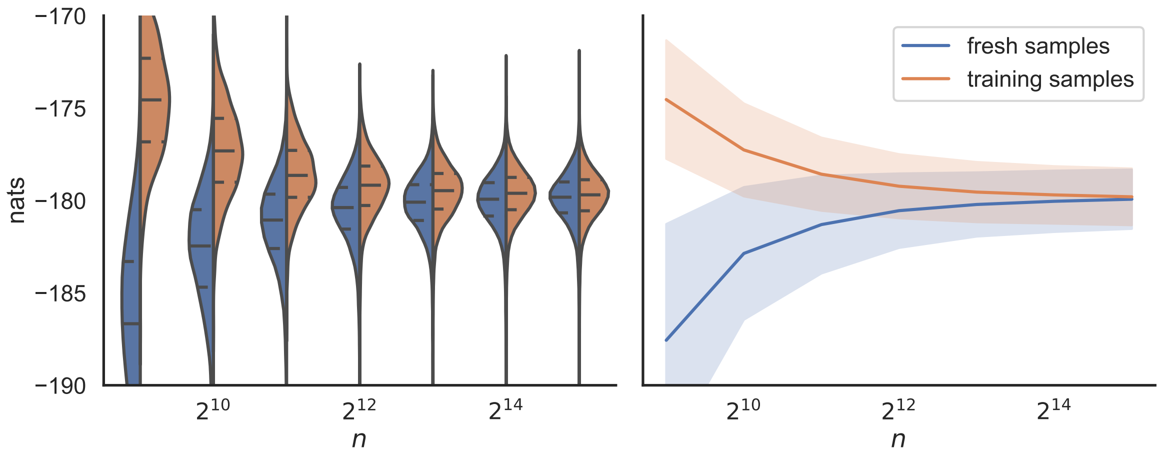

Figure 1: Distribution of log-weights for a fresh sample of noise and the training noise, as a function of the number n of samples used for training on the `mushrooms` dataset. (Left) Violin plot showing the distribution of log-weights. (Right) Line plot depicting the mean $(\bar{v} \pm \sigma)$ of the log-weights. The means of the log-weights correspond to an estimation of the ELBO and the training objective. The overfitting to the training noise is reduced by training using a larger sample size.

the ELBO and the training objective by comparing the distribution of the log-weights for a fresh sample of noise (testing noise) and the training noise. Figure 1 displays the distribution of log-weights for a growing sample size. As the number of samples increases, the training objective value decreases and approaches that of the ELBO estimation, which in turn increases, indicating progress toward ELBO maximization. We mitigate overfitting by solving a sequence of SAA approximations for an increasing sequence of sample sizes, which creates a sequence of solutions. This approach is based on epi-convergence, implying that under certain conditions, the sequence of solutions will converge to a solution of the original problem with probability one. However, it is unknown how well these theoretical results can be applied to our setting, and this remains an area for future research.

3.2. Algorithm

In this section, we present an algorithm that uses SAA to approximate the solution to the optimization problem of maximizing the ELBO. Our objective is to find a good approximation to the solution with a reasonable computational cost and avoid the overfitting phenomenon described earlier. To this end, we build our stopping criteria based on overfitting. In Algorithm 1 in Appendix A we describe the algorithm, consisting of two procedures: the optimizer Opt and the convergence checker. We previously described the optimizer, in which we used a quasi-Newton method. The convergence checker determines whether we need to continue the optimization process, and we will describe it below.

The algorithm starts with an initial guess θ_0, an initial sample size n_0, and an initial maximum number of iterations for the optimizer τ_0. Default values for these parameters can be found in Table 2, also located in Appendix A. At each iteration t of the algorithm, we double the sample size to reduce the overfitting. First, we draw the training noise

$\boldsymbol{\epsilon}_{n_t} = \epsilon_1, \ldots, \epsilon_{n_t}$ from the base distribution q_{base}. Then, we use the optimizer to find the parameters θ_t^* that maximize the deterministic objective computed with the fixed training noise $\boldsymbol{\epsilon}_{n_t}$. If we find that the optimizer has reached the maximum number of iterations τ_t, we double the maximum number of iterations for the optimizer; otherwise, we keep the same maximum number of iterations. We repeat this process until the convergence checker determines that we have reached a good approximation to the overall solution of the stochastic optimization problem.

Stopping Algorithm 2, in the appendix, defines the stopping criteria for our optimization process, which involves computing log-weights. Specifically, given the training noise $\boldsymbol{\epsilon}_{n_t}$ and the parameters θ_t, we compute the log-weights $v_{\theta_t}(\epsilon_1), \ldots, v_{\theta_t}(\epsilon_{n_t})$, which we denote as $v_{\theta_t}(\boldsymbol{\epsilon}_{n_t})$. We also compute a new set of log-weights using a fresh sample of testing noise with size 10k, denoted by $v_{\theta_t}(\hat{\boldsymbol{\epsilon}}_{10\text{k}})$.
To decide when to stop optimizing, we use a two-sided t-test to compare the distribution of log-weights computed using the training noise $v_{\theta_t}(\boldsymbol{\epsilon}_{n_t})$ with the distribution of log-weights computed using a the testing noise $v_{\theta_t}(\hat{\boldsymbol{\epsilon}}_{10\text{k}})$. The null hypothesis is that the means of the two distributions are the same. This test was inspired by the test used in Mak et al. (1999). Our optimization process terminates when the hypothesis cannot be rejected with a significance level of 1%. We also introduce two additional stopping conditions: the maximum number of iterations max_t and the threshold δ for the difference between the training objective $\hat{\mathcal{L}}_{\boldsymbol{\epsilon}}(\theta_t)$ and the ELBO $\mathcal{L}(\theta_t)$. In our experiments, we set max_t to ensure that the maximum sample size was $n_{\max} = 2^{18}$, and δ to 0.01.

4. Related work

In the literature, several methods incorporate second-order information into stochastic optimization. Byrd et al. (2016) introduced batched-L-BFGS, which was later applied by Liu and Owen (2021) to the variational inference problem, with optional quasi-Monte Carlo (QMC) sampling. The methods involve a two-step algorithm and use a fixed-size noise sample for estimating the ELBO gradient. Our method differs from Liu and Owen (2021)'s approach by optimizing with a fixed set of noise and integrating the sample size consideration into the algorithm.
An alternative approach by Zhang et al. (2022) employs L-BFGS to identify modes or poles of the posterior distribution and uses the generated data to estimate the posterior covariance around the mode. This method resembles the Laplace approximation more closely.
Welandawe et al. (2022), inspired by Agrawal et al. (2020), employ SGD with a heuristic step-size schedule for ELBO optimization and use tools for stationarity detection. They observed that parameter averaging in the stationary regime enhances approximation quality.
In the machine learning literature, sample average approximation has been less common. Early works like Ng and Jordan (2000)'s PEGASUS applied the technique in the context of partially observable Markov decision processes, while Sheldon et al. (2010) used it in a network design setting. More recently, Balandat et al. (2020) adopted the technique for optimizing the acquisition function in Bayesian optimization.

5. Experiments

We now present the most relevant findings from our experiments, which are detailed in Appendix B. To evaluate the effectiveness of our method, we compared it with the Adam optimizer and the batched quasi-Newton method proposed by Liu and Owen (2021). We tested 12 models from the Stan repository (Stan Development Team, 2021; Carpenter et al., 2017) and Bayesian logistic regression with 5 UCI datasets (Dua and Graff, 2017).

Our results show that for simpler models, all methods performed similarly when a Gaussian distribution with diagonal covariance was used as an approximating distribution. However, for more complex models, our method outperformed the other methods when a Gaussian distribution with full covariance was used. Interestingly, our method usually found a better optimizer than Adam (with the best of three swept step sizes), sometimes achieving an ELBO that was 200 nats higher and at a faster speed.

Comparing our method with batched quasi-Newton on the Stan models, as shown in Table 5, revealed the difficulties of not incorporating a method to determine the right sample size and the benefits of using a fixed set of noise samples for optimization. With most of the models, batched quasi-Newton was unable to find a good optimizer, even when using 128 samples.

	ELBO Diff.	Time ratio
	Adam − SAA for VI	Adam/SAA for VI
Stan models		
congress	0.03	51.12
election88	-239.57	1.01
election88Exp	—	2.30
electric	-74.20	5.73
electric-one	-0.00	60.18
hepatitis	-68.31	1.11
irt	-51.59	0.52
mesquite	0.04	127.40
radon	-6.97	6.59
sonar	-0.28	3.90
wells	0.03	114.24

Table 1: Comparison of ELBO differences and time ratio between Adam and SAA for VI on Stan models using full-rank covariance. ELBO Diff. represents the difference between the median ELBO achieved by Adam and SAA for VI, with negative values indicating that SAA for VI achieved a higher ELBO. Time ratio values greater than 1 indicate that Adam is slower than SAA for VI. See Tables 3 and 4 in Appendix B for more details.

6. Conclusion

In this paper, we introduced the SAA for VI algorithm, which provides an effective and accurate solution to variational inference problems, significantly reducing the reliance on manual hyperparameter tuning. This promising method enhances both efficiency and precision in addressing these challenges.

References

Abhinav Agrawal, Daniel R Sheldon, and Justin Domke. Advances in black-box vi: Normalizing flows, importance weighting, and optimization. *Advances in Neural Information Processing Systems*, 33:17358–17369, 2020.

Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. Botorch: a framework for efficient monte-carlo bayesian optimization. *Advances in neural information processing systems*, 33:21524–21538, 2020.

Matthew J Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, UCL (University College London), 2003.

Charles G Broyden. The convergence of a class of double-rank minimization algorithms: 2. the new algorithm. *IMA journal of applied mathematics*, 6(3):222–231, 1970.

Javier Burroni, Kenta Takatsu, Justin Domke, and Daniel Sheldon. U-statistics for importance-weighted variational inference. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=oXmwAPlbVw`.

Richard H Byrd, Samantha L Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *SIAM Journal on Optimization*, 26 (2):1008–1031, 2016.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

Huifen Chen and Bruce W Schmeiser. Stochastic root finding via retrospective approximation. *IIE Transactions*, 33(3):259–275, 2001.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

Michael C Fu. Gradient estimation. *Handbooks in operations research and management science*, 13:575–616, 2006.

Donald Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24(109):23–26, 1970.

K. Healy and L.W. Schruben. Retrospective simulation response optimization. In *1991 Winter Simulation Conference Proceedings.*, pages 901–906, 1991. doi: 10.1109/WSC.1991.185703.

Tommi S Jaakkola and Michael I Jordan. A variational approach to bayesian logistic regression models and their extensions. In *Sixth International Workshop on Artificial Intelligence and Statistics*, pages 283–294. PMLR, 1997.

Sujin Kim, Raghu Pasupathy, and Shane G Henderson. A guide to sample average approximation. *Handbook of simulation optimization*, pages 207–243, 2015.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

Anton J. Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002. doi: 10.1137/S1052623499363220. URL `https://doi.org/10.1137/S1052623499363220`.

Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. Automatic differentiation variational inference. *Journal of machine learning research*, 2017.

Sifan Liu and Art B Owen. Quasi-monte carlo quasi-newton in variational bayes. *The Journal of Machine Learning Research*, 22(1):11043–11065, 2021.

Wai-Kei Mak, David P Morton, and R Kevin Wood. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations research letters*, 24(1-2): 47–56, 1999.

Andrew Y Ng and Michael Jordan. Pegasus: a policy search method for large mdps and pomdps. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 406–415, 2000.

Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.

Jorge Nocedal and Stephen J Wright. *Numerical optimization.* Springer, 1999.

Raghu Pasupathy. On choosing parameters in retrospective-approximation algorithms for stochastic root finding and simulation optimization. *Operations Research*, 58(4-part-1): 889–901, 2010.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR, 2014.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Stephen M Robinson. Analysis of sample-path optimization. *Mathematics of Operations Research*, 21(3):513–528, 1996.

David F Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24(111):647–656, 1970.

Alexander Shapiro and Yorai Wardi. Convergence analysis of stochastic algorithms. *Mathematics of operations research*, 21(3):615–628, 1996.

Daniel Sheldon, Bistra Dilkina, Adam N Elmachtoub, Ryan Finseth, Ashish Sabharwal, Jon Conrad, Carla Gomes, David Shmoys, William Allen, Ole Amundsen, et al. Maximizing the spread of cascades using network design. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pages 517–526, 2010.

Stan Development Team. Stan Example models, 2021. URL `https://github.com/stan-dev/example-models`.

Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

Manushi Welandawe, Michael Riis Andersen, Aki Vehtari, and Jonathan H Huggins. Robust, automated, and accurate black-box variational inference. *arXiv preprint arXiv:2203.15945*, 2022.

David Wingate and Theophane Weber. Automated variational inference in probabilistic programming. *arXiv preprint arXiv:1301.1299*, 2013.

Lu Zhang, Bob Carpenter, Andrew Gelman, and Aki Vehtari. Pathfinder: Parallel quasi-newton variational inference. *Journal of Machine Learning Research*, 23(306):1–49, 2022.

Appendix A. Algorithm Pseudocode

Algorithm 1: SAA for VI

```
Input: $\theta_0$, $n_0$, $\tau_0$
Output: $\theta^*$
$t \leftarrow 0$
while $t = 0$ or not converged($\theta_t, \boldsymbol{\epsilon}_{n_t}, t$)? do
    $t \leftarrow t + 1$, $n_t \leftarrow 2n_{t-1}$
    $\boldsymbol{\epsilon}_{n_t} \leftarrow \epsilon_1, \ldots, \epsilon_{n_t}$, where $\epsilon_i \sim q_{\text{base}}$
    $\theta_t \leftarrow \text{Opt}(\theta_{t-1}, n_t, \boldsymbol{\epsilon}_{n_t}, \tau_t)$
    if optimizer reached max iters $\tau_t$
        $\tau_{t+1} \leftarrow 2\tau_t$
    else
        $\tau_{t+1} \leftarrow \tau_t$
return $\theta^* \leftarrow \theta_t$
```

Algorithm 2: converged?

```
Input: $\theta_t$, $\boldsymbol{\epsilon}_{n_t}$, $t$
Data: max_t, $\delta$
Output: converged, a boolean
$\hat{\boldsymbol{\epsilon}}_{10\text{k}} \leftarrow \hat{\epsilon}_1, \ldots, \hat{\epsilon}_{10\text{k}}$, where $\hat{\epsilon}_i \sim q_{\text{base}}$
obj $\leftarrow$ mean($v_{\theta_t}(\boldsymbol{\epsilon}_{n_t})$)
elbo $\leftarrow$ mean($v_{\theta_t}(\hat{\boldsymbol{\epsilon}}_{10\text{k}})$)
/* Perform t-test:              */
test $\leftarrow$ t_test($v_{\theta_t}(\boldsymbol{\epsilon}_{n_t}), v_{\theta_t}(\hat{\boldsymbol{\epsilon}}_{10\text{k}})$)
if (test.pvalue $< 0.01$)
    converged $\leftarrow |\text{elbo} - \text{obj}| < \delta$
else
    converged $\leftarrow$ True
return converged;
```

Input	Default value
θ_0	random from $\mathcal{N}(0, 1)$
n_0	32
τ_0	300

Table 2: Default values for the input parameters of the algorithm.

Appendix B. Experiments

In this section, we present the detailed experimental evidence for our proposed method. We adopt the experimental setup of Burroni et al. (2023) and consider two types of models: 12 models from the Stan examples repository (Stan Development Team, 2021; Carpenter et al., 2017) and Bayesian logistic regression with 5 UCI datasets (Dua and Graff, 2017). For each model $p(\mathbf{Z}, x)$, where $\mathbf{Z}$ is a d-dimensional random vector, the approximating distribution q_θ can either be a diagonal Gaussian or a d-dimensional multivariate Gaussian distribution. The former is a product of d independent Gaussians, where the parameters μ_i and $\sigma_i^2 > 0$ are specific to each Z_i. The latter has parameters μ_i and LL^{T}, where $L \in \mathbb{R}^{d \times d}$ is a lower-triangular matrix with diagonal elements that are positive, enforced by applying the `softplus` transformation. We use the constraints framework from PyTorch (Paszke et al., 2019) to transform the model p into one with unconstrained real-valued latent variables, as done by Kucukelbir et al. (2017).

We run two sets of experiments. First, we conduct performance comparisons where we assess our proposed method against two other methods: Adam with a fixed step-size, which is commonly used for black-box VI optimization, and batched quasi-Newton, a newer method that introduces second-order information in the optimization process. Second, we conduct an ablation study to explore how our decisions affect the algorithm's performance. We present the results of these experiments in the following subsections.

B.1. Performance comparison

B.1.1. Adam

In order to solve the black-box VI problem, it is standard practice to use Adam (Kingma and Ba, 2015) as the default optimizer. This is evident from examples in Pyro[1] and the TensorFlow-Probability VI tutorial.[2] Despite the fact that the influence of the step-size in the optimization process is less relevant with Adam than with SGD, it is still a factor to consider. In our study, we compared Adam to our proposed method, SAA for VI. For Adam, we optimized each model and approximating distribution combination with three different step-sizes: 0.1, 0.01, and 0.001, and 20 repetitions of each combination. At each iteration with Adam, we estimated the gradient of the ELBO by taking 16 samples from q_θ. For each model and approximating distribution, we selected the step-size that provided the highest median ELBO across the 20 repetitions. Please see Appendix C for more details on the Adam experiments. For SAA for VI, we used the algorithm described in Section 3.2, using the default parameter values of Table 2.
We conducted two comparisons for our study. First, we compared the median ELBO, obtained across 20 repetitions, at the end of the optimization process using Adam and SAA for VI. Initially, we ran the Adam experiments for $40,000$ iterations, but we found that for some models, there was a persistent large gap between the maximum median ELBO achieved with Adam and that of SAA for VI. We increased the maximum number of iterations to reduce the gap for models such as `election88`, `electric`, `irt`, `madelon`, and `radon`. (See Table 9 in the appendix). Table 3 presents the comparisons of median ELBOs. Although the Adam optimizer achieves a slightly higher median ELBO for some models—due to the stopping criterion of SAA for VI—, SAA for VI achieves a noticeably higher median ELBO for complex models. We also observed that Adam diverged for models such as `election88Exp`. Additionally, Adam diverged for the `hepatitis` model when optimized for more than $40,000$ iterations, which partially explained the large gap between the median ELBOs of Adam and SAA for VI.
Second, we compared the time taken to achieve a given ELBO. For each combination of model and approximating distribution, we computed the minimum between the median ELBO achieved by Adam and the median ELBO achieved by SAA for VI. This allows us to determine a value of the ELBO that was achieved for at least 50% of the runs, regardless of the optimization. We then computed, for each run, the least time required for the ELBO to equal the previously computed value. Table 4 presents the time (in seconds) required to achieve the ELBO when using Adam and our proposed method and the ratio between them. For example, running optimization for the `wells` model takes almost a minute when

1. See, for instance, the examples in Pyro-SVI.
2. Adam is also used in the TensorFlow-Probability VI Tutorial.

	Diagonal Covariance			Full Rank Covariance		
	Adam (i)	SAA for VI (ii)	Diff. (i) − (ii)	Adam (iv)	SAA for VI (v)	Diff. (iv) − (v)
Bayes log. regr.						
a1a	-654.80	-655.62	0.82	-637.22	-636.40	-0.82
australian	-268.36	-269.40	1.04	-256.82	-256.73	-0.09
ionosphere	-138.30	-139.28	0.98	-124.44	-124.35	-0.09
madelon	-2,466.27	-2,466.18	-0.09	-2,600.39	-2,399.65	-200.73
mushrooms	-210.01	-211.31	1.29	-180.60	-179.88	-0.72
Stan models						
congress	421.91	421.78	0.13	423.58	423.55	0.03
election88	-1,419.04	-1,419.75	0.71	-1,636.65	-1,397.08	-239.57
election88Exp	-1,376.05	-1,381.72	5.68	—	-1,391.98	—
electric	-788.84	-788.88	0.04	-861.11	-786.91	-74.20
electric-one-pred	-818.33	-818.37	0.04	-818.00	-818.00	-0.00
hepatitis	-560.43	-560.44	0.01	-625.68	-557.37	-68.31
hiv-chr	-608.41	-608.80	0.39	—	-582.77	—
irt	-15,888.03	-15,887.92	-0.10	-15,936.25	-15,884.66	-51.59
mesquite	-30.08	-30.18	0.10	-29.78	-29.82	0.04
radon	-1,210.64	-1,210.69	0.04	-1,216.43	-1,209.46	-6.97
sonar	-149.59	-152.18	2.59	-110.32	-110.04	-0.28
wells	-2,042.37	-2,042.43	0.06	-2,041.90	-2,041.93	0.03

Table 3: Comparison of Adam and SAA for VI: Median of the highest ELBO achieved across multiple optimization runs with different seeds for each model and approximating distribution. Adam was optimized using step-sizes of 0.1, 0.01, and 0.001, and the configuration with the highest median ELBO is reported. We additionally included the difference between the median ELBO achieved by Adam and SAA for VI: negative values indicate that SAA for VI achieved a higher ELBO than Adam. For further details, see Section B.1. The full results are provided in Tables 7 and 8 in Appendix C.

using Adam, compared to less than 200 milliseconds when using SAA for VI, i.e., the time required to achieve the same ELBO is approximately 300 times slower when using Adam. It is worth noting that SAA for VI was at a disadvantage in the comparison, because the actual compute time required by Adam was three time larger than the reported one due to the selection of the step-size.

B.1.2. Batched quasi-Newton

As noted earlier in Section 4, our method exhibits certain differences compared to the batched quasi-Newton technique developed by Liu and Owen (2021), which also integrates second-order information into VI. In this section, we aim to empirically highlight the signifi-

	Diagonal Covariance			Full Rank Covariance		
	Adam (i)	SAA for VI (ii)	Ratio (i)/(ii)	Adam (iv)	SAA for VI (v)	Ratio (iv)/(v)
Bayes log. regr.						
a1a	29.01	1.23	23.64	79.37	64.00	1.24
australian	13.22	0.15	88.70	51.96	14.40	3.61
ionosphere	9.10	0.15	61.08	28.45	9.98	2.85
madelon	111.49	17.67	6.31	1,532.24	486.04	3.15
mushrooms	125.88	5.15	24.43	219.44	293.83	0.75
Stan models						
congress	26.39	0.12	222.89	37.96	0.74	51.12
election88	278.21	13.91	20.00	473.31	467.31	1.01
election88Exp	265.38	9.69	27.40	264.24	114.79	2.30
electric	52.41	4.34	12.08	177.32	30.92	5.73
electric-one-pred	34.01	0.54	63.17	48.12	0.80	60.18
hepatitis	173.97	12.26	14.19	203.42	183.29	1.11
hiv-chr	46.55	2.79	16.67	136.51	34.55	3.95
irt	572.47	310.32	1.84	1,650.03	3,184.38	0.52
mesquite	16.34	0.09	183.35	30.91	0.24	127.40
radon	89.78	13.97	6.43	185.43	28.16	6.59
sonar	7.55	0.23	32.96	47.74	12.23	3.90
wells	56.17	0.17	323.48	68.20	0.60	114.24

Table 4: Comparison of running time, in seconds, for Adam and SAA in VI across different datasets and distribution approximations, and Adam to SAA time ratio. Values of ratio greater than 1 indicate that Adam is slower than SAA for VI. See Section B.1 for more information.

cance of these differences, specifically the use of a sequence of sample average approximations with an increasing number of samples.

To carry out this comparison, we implemented the batched quasi-Newton method in PyTorch without employing quasi-Monte Carlo sampling and compared it to our method. We ran the experiments for 40,000 iterations, with 20 independent runs for each. Initially, we used a sample size of 16 and increased it by a factor of 2 for models where the method encountered difficulties, up to a maximum of 128 samples. We consistently used $B = 20$ as recommended in the original paper.

When employing a simpler approximating distribution, such as a Gaussian distribution with a diagonal covariance matrix, the batched quasi-Newton method demonstrates performance on par with SAA for VI (refer to Table 10 in the appendix). However, the method encounters difficulties when using a more complex Gaussian distribution with a full-rank covariance matrix as the approximating distribution.

Table 5 displays the median final ELBO across runs for various models. The batched quasi-Newton method reaches optimal performance for most Bayesian logistic regression models but faces difficulties with models from the Stan example library. Even when increasing the sample size to 128, a significantly larger sample size than commonly employed with SGD, the method still falls short of reaching the optimal value. Additionally, we show in the appendix that the wall-clock time taken by the batched quasi-Newton method is often similar to or slower than the time taken by SAA for VI.

B.2. Ablation study

Impact of warm start. The optimization process requires a decision on whether to use warm start or draw fresh parameters for each iteration. Suppose that the inner optimization process Opt has already converged to parameters θ_t^*. Despite the convergence, it may still be necessary to run the inner optimization process more times, as described in Section 3.2, to reduce overfitting. The question then arises whether it is computationally advantageous to use θ_t^* as the initial parameters or to draw a new set of parameters from a suitable distribution.

Pasupathy (2010) provides an intuition of why using a warm start is helpful: in principle, the optimization process for larger sample sizes begin from a place that probably is close to a solution. However, we wanted to empirically verify this intuition. To determine the most efficient approach, we conducted an experiment to compare the performance of warm start and drawing fresh parameters across different models and approximating distributions. For each combination of models and distribution, we ran the sequence of SAA problems until convergence, using either warm start or by sampling new parameters at the beginning of each inner optimization. Specifically, for the sequence of sample sizes $(n_t)_{t\in\mathbb{N}}$ described above, we ran the inner optimization process Opt until it converged. At each iteration t, we initialized the process either with the previously computed optimal parameters θ_{t-1}^* (warm start) or by drawing a new random set of parameters (fresh start). We continue this process until the algorithm converges. We again used 20 repetitions for each configuration and report the median results. Our results, presented in Table 6, show that although the difference in nats between the median run is small, using warm start results in a significant reduction in the total time taken to converge. For example, on the `election88` dataset, using fresh samples

		Full Rank Covariance				
		Batched Quasi-Newton—Sample Size				SAA for VI
		16	32	64	128	
Bayes log. regr.						
a1a		-636.49				-636.40
australian		-256.80				-256.73
ionosphere		-124.44				-124.35
madelon	✗	-2,418.90	-2,412.41	-2,407.51	-2,406.02	-2,399.65
mushrooms		-179.96				-179.88
Stan models						
congress		423.59				423.55
election88	✗	-6.06×10^{11}	-7.74×10^{11}	-8.64×10^{11}	-5.42×10^{11}	-1,397.08
election88Exp	✗	-6.60×10^{18}	-1.91×10^{18}	-2.90×10^{16}	-8.34×10^{15}	-1,391.98
electric	✗	-2.14×10^{10}	-8.02×10^{9}	-2.81×10^{9}	-1.67×10^{9}	-786.91
electric-one-pred		-1,021.43	-901.83	-818.00		-818.00
hepatitis	✗	-2.33×10^{10}	-1.20×10^{10}	-1.51×10^{10}	-9.30×10^{9}	-557.37
hiv-chr	✗	-2.49×10^{15}	-5.32×10^{15}	-1.39×10^{15}	-1.85×10^{14}	-582.77
irt	✗	-1.08×10^{5}	-17,758.76	-16,619.78	-16,121.43	-15,884.66
mesquite		-29.78				-29.82
radon	✗	-4.09×10^{6}	-3.13×10^{5}	-53,638.39	-1.71×10^{5}	-1,209.46
sonar		-110.10				-110.04
wells		-2,041.90				-2,041.93

Table 5: ELBO achieved by the batched quasi-Newton method for VI with a full-rank Gaussian, as proposed by Liu and Owen (2021). The results for SAA for VI are included as a benchmark (refer to column (v) of Table 3). It is observed that the batched quasi-Newton method frequently converges to suboptimal solutions, indicated by ✗, especially in models from the Stan examples repository. In certain cases, such as the `election88` dataset, the SAA for VI method demonstrates a significant performance advantage over the batched quasi-Newton method. The initial sample size for the batched quasi-Newton method was set to 16 and increased when necessary to enhance the method's ELBO.

	Fresh start to Warm start ELBO difference (nats)		Fresh start to Warm start time ratio	
	Diagonal	Full Rank	Diagonal	Full Rank
Bayes log. regr.				
a1a	0.11	0.00	1.45	2.10
australian	-0.13	0.00	1.09	1.92
ionosphere	-0.14	-0.00	1.13	2.21
madelon	-0.03	-0.01	2.44	8.23
mushrooms	0.04	-0.00	1.73	4.42
Stan models				
congress	0.00	0.04	1.10	8.22
election88	-1.90	0.82	6.60	53.81
election88Exp	-3.11	7.15	6.07	17.77
electric	-0.01	-0.00	3.74	7.66
electric-one-pred	0.00	-0.03	1.46	0.36
hepatitis	0.00	0.01	3.27	3.25
hiv-chr	0.08	-0.09	4.45	1.54
irt	0.00	—	8.28	—
mesquite	-0.00	-0.01	1.10	1.50
radon	0.00	0.01	3.30	5.60
sonar	0.29	-0.00	1.40	1.18
wells	0.00	-0.00	1.05	1.13

Table 6: Median ELBO variation in nats resulting from switching between two approaches: fresh start, where parameters are refreshed at each iteration to warm start, where previously learned parameters are used as the starting point. (Negative values indicate that the warm start approach is better.) We also provide the ratio of median time taken by the fresh start approach compared to the warm start approach. (Values larger than 1 indicate that the warm start approach is faster.) Our results indicate that warm start approaches can significantly reduce the optimization time required.

takes 53× more time than using a warm start due to the inner optimization process Opt taking more iterations to find a good solution at each step.

Appendix C. Detailed comparison with Adam

We now provide additional details about the experimental setup presented in Section B.1. We used the Adam optimizer with the default parameters from the `torch.optim` package in PyTorch (Paszke et al., 2019), except for the step-size, which we varied across 0.1, 0.01, and 0.001. To approximate the distributions, we used a Gaussian with a Diagonal covariance matrix and a more expressive Gaussian with a Full-Rank covariance matrix. Tables 7 and 8 present the experiment results disaggregated by step-size. In all cases we ran 20 repetitions of the experiments, and we estimated the objective function using 16 samples from the variational approximation q_{θ_t}. Every 100 iterations we estimate the ELBO using $10,000$ fresh samples from q_{θ_t}. Initially, we ran the experiments for $40,000$ iterations, but we found that the Full-Rank approximation produced unsatisfactory results for some models. We, therefore, increased the number of iterations for those models, but observed only slight changes in the maximum achieved ELBO, as shown in Table 9 and Table 8. It is also worth noting that the `hepatitis` model diverged when we ran it for more than $40,000$ iterations using the Full-Rank approximation.

	Adam—Step Size			SAA for VI
	0.1	0.01	0.001	
Bayes log. regr.				
a1a	-656.28	-655.00	-654.80	-655.62
australian	-268.81	-268.42	-268.36	-269.40
ionosphere	-138.88	-138.38	-138.30	-139.28
madelon	-2,495.47	-2,470.17	-2,466.27	-2,466.18
mushrooms	-210.98	-210.21	-210.01	-211.31
Stan models				
congress	421.86	421.90	421.91	421.78
election88	-1,436.03	-1,420.13	-1,419.04	-1,419.75
election88Exp	-1,376.40	-1,376.05	-1,381.56	-1,381.72
electric	-790.67	-789.06	-788.84	-788.88
electric-one-pred	-818.34	-818.33	-1,062.59	-818.37
hepatitis	-564.10	-560.84	-560.43	-560.44
hiv-chr	-611.65	-608.81	-608.41	-608.80
irt	-15,895.91	-15,889.38	-15,888.03	-15,887.92
mesquite	-30.09	-30.08	-30.08	-30.18
radon	-1,211.59	-1,210.79	-1,210.64	-1,210.69
sonar	-151.07	-149.81	-149.59	-152.18
wells	-2,042.38	-2,042.37	-2,042.37	-2,042.43

Table 7: Maximum ELBO Achieved by Adam and SAA for VI with Gaussian Distribution and **Diagonal** Covariance Matrix as Approximating Distribution: Median Across Seeds. The table shows the median of the maximum ELBO achieved by Adam and SAA for each model when using a Gaussian Distribution with Diagonal covariance matrix as approximating distribution. For each step-size used with Adam, we ran the algorithm 20 times and reported the median of the maximum ELBO achieved.

	Adam—Step Sizes			SAA for VI
	0.1	0.01	0.001	
Bayes log. regr.				
a1a	-1,366.86	-646.02	-637.22	-636.40
australian	-269.17	-257.54	-256.82	-256.73
ionosphere	-147.75	-125.17	-124.44	-124.35
madelon	-66,611.92	-7,596.15	-2,600.39	-2,399.65
mushrooms	-243.46	-182.63	-180.60	-179.88
Stan models				
congress	423.32	423.53	423.58	423.55
election88	—	-1,636.65	—	-1,397.08
election88Exp	—	—	—	-1,391.98
electric	—	-861.11	—	-786.91
electric-one-pred	-818.01	-818.00	-1,085.83	-818.00
hepatitis	—	-625.68	—	-557.37
hiv-chr	—	—	—	-582.77
irt	-127,607.15	-18,846.63	-15,936.25	-15,884.66
mesquite	-29.80	-29.79	-29.78	-29.82
radon	—	-1,216.43	-106,178.55	-1,209.46
sonar	-383.33	-114.35	-110.32	-110.04
wells	-2,041.91	-2,041.90	-2,041.90	-2,041.93

Table 8: Maximum ELBO Achieved by Adam and SAA for VI with Gaussian Distribution and **Full-Rank** Covariance Matrix as Approximating Distribution: Median Across Seeds. The table shows the median of the maximum ELBO achieved by Adam and SAA for each model when using a Gaussian Distribution with Full-Rank covariance matrix as approximating distribution. For each step-size used with Adam, we ran the algorithm 20 times and reported the median of the maximum ELBO achieved.

	Adam	
	Diagonal Covariance	Full Rank Covariance
Bayes log. regr.		
a1a	40,000	40,000
australian	40,000	40,000
ionosphere	40,000	40,000
madelon	40,000	400,000
mushrooms	40,000	40,000
Stan models		
congress	40,000	40,000
election88	40,000	400,000
election88Exp	40,000	40,000
electric	40,000	400,000
electric-one-pred	40,000	40,000
hepatitis	40,000	40,000
hiv-chr	40,000	40,000
irt	40,000	200,000
sonar	40,000	40,000
mesquite	40,000	40,000
radon	40,000	400,000
wells	40,000	40,000

Table 9: Maximum number of iterations for Adam optimization using Gaussian distribution with **Diagonal** or **Full-Rank** Covariance Matrix. Some models (`election88`, `electric`, `irt`, `madelon`, and `radon`) were run for up to 10 times more iterations to achieve a comparable ELBO to SAA for VI.

Appendix D. Detailed comparison with Batched-BFGS

In this section, we provide further details about the experiments conducted using the Batched-BFGS method of Liu and Owen (2021). Table 10 compares the performance of Batched-BFGS with our method when the approximating distribution is a Gaussian distribution with a diagonal covariance matrix. This table complements Table 5. As mentioned earlier, the results in this setting are quite similar to ours.

Additionally, we report the wall-clock time for each experiment in Table 11. We executed each experiment for 40,000 iterations and performed 20 independent runs for each one. Our method incorporates a stopping criterion based on convergence. To ensure a fair comparison with Batched-BFGS, we need to detect when the algorithm converges. The closest approximation to this is calculating the total time taken until the algorithm reaches within 1 nat of the maximum ELBO achieved in that run. These results are presented in Table 11.

Similar to the experiments with Adam, this calculation does not account for the time spent on sample sizes that were not useful.

	Diagonal Gaussian	
	Batched Quasi-Newton 16	SAA for VI
Bayes log. regr.		
a1a	-654.95	-655.62
australian	-268.48	-269.40
ionosphere	-138.48	-139.28
madelon	-2,466.61	-2,466.18
mushrooms	-210.31	-211.31
Stan models		
congress	421.91	421.78
election88	-1,425.87	-1,419.75
election88Exp	-1,382.65	-1,381.72
electric	-788.88	-788.88
electric-one-pred	-851.43	-818.37
hepatitis	-560.58	-560.44
hiv-chr	-608.58	-608.80
irt	-15,888.14	-15,887.92
mesquite	-30.08	-30.18
radon	-1,210.73	-1,210.69
sonar	-150.18	-152.18
wells	-2,042.38	-2,042.43

Table 10: Comparison of the ELBOs obtained by Batched-BFGS and SAA for VI when using a diagonal Gaussian distribution as the approximating distribution. The Batched-BFGS method of Liu and Owen (2021) is executed using a sample size of 16. Median results are reported from 20 independent runs for each model. The corresponding results for SAA for VI can also be found in column (ii) of Table 3.

		Batched Quasi-Newton				
		Diagonal–Sample Size	Full Rank–Sample Size			
		16	16	32	64	128
Bayes log. regr.						
a1a		2.25	61.08	38.36	25.75	30.79
australian		0.83	4.13	3.17	3.26	3.68
ionosphere		0.71	3.10	2.05	2.21	1.97
madelon	✗	9.35	3519.20	6823.51	4420.06	1665.36
mushrooms		7.16	98.28	98.99	64.74	141.95
Stan models						
congress		1.56	5.24	5.17	4.96	6.13
election88	✗	1405.99	6.09	3.46	4.50	10.41
election88Exp	✗	576.61	2.52	3.29	4.79	7.05
electric	✗	9.64	1.11	2.60	4.20	1.84
electric-one-pred	✗	1.25	0.72	2.19	3.18	3.31
hepatitis	✗	12.99	4.11	4.95	5.25	6.79
hiv-chr		14.93	1.31	1.45	1.80	2.11
irt	✗	231.20	5250.35	22133.59	21165.13	30352.88
mesquite		0.48	0.55	0.65	0.64	0.71
radon	✗	5.53	1.43	3.11	23.61	515.68
sonar		0.76	9.45	4.96	4.10	3.76
wells		0.97	1.05	1.99	3.35	7.15

Table 11: Median wall-clock time in seconds for the Batched Quasi-Newton optimization process to reach within 1 nat of the maximum ELBO. The ✗ denotes non-convergence for models using Batched-BFGS with a Full Rank Covariance Matrix approximation. The results show comparable performance to SAA for VI [refer to Table 4, columns (ii) and (v)].