Pathwise Gradient Variance Reduction with Control Variates in Variational Inference

Kenyon NG[1][0000−0002−6315−9831] and Susan Wei[1][0000−0002−6842−2352]

The University of Melbourne
kenyon.ng@student.unimelb.edu.au,susan.wei@unimelb.edu.au

Abstract. Variational inference in Bayesian deep learning often involves computing the gradient of an expectation that lacks a closed-form solution. In these cases, pathwise and score-function gradient estimators are the most common approaches. The pathwise estimator is often favoured for its substantially lower variance compared to the score-function estimator, which typically requires variance reduction techniques. However, recent research suggests that even pathwise gradient estimators could benefit from variance reduction. In this work, we review existing control-variates-based variance reduction methods for pathwise gradient estimators to assess their effectiveness. Notably, these methods often rely on integrand approximations and are applicable only to simple variational families. To address this limitation, we propose applying zero-variance control variates to pathwise gradient estimators. This approach offers the advantage of requiring minimal assumptions about the variational distribution, other than being able to sample from it.

Keywords: Stochastic variational inference · Reparametrization trick · Pathwise gradient · Zero-variance control variates · Bayesian deep learning

1 Introduction

Given an observed dataset $\mathcal{D} = \{x_i\}_{i=1}^N$ governed by a data generating process that depends on latent variables $z \in \mathbb{R}^{\dim_z}$ and a prior $p(z)$ of these latent variables, we are often interested in the posterior distribution $p(z|\mathcal{D}) \propto p(\mathcal{D}|z)p(z)$. This posterior is often known only up to a normalising constant and requires approximation. Variational inference (VI) offers a way to approximate the posterior with a simpler, tractable distribution from the variational family $\mathcal{Q} = \{q(z;\lambda) : \lambda \in \mathbb{R}^{\dim_\lambda}\}$. This is typically done by minimising the Kullback-Leibler (KL) divergence from the variational distribution $q(z;\lambda)$ to $p(z|\mathcal{D})$, expressed as $\mathbb{E}_{q(z;\lambda)}[\log q(z;\lambda) - \log p(z|\mathcal{D})]$, or equivalently, maximising the evidence lower bound (ELBO)

$$\lambda^* = \arg\max_\lambda \mathbb{E}_{q(z;\lambda)}[\log p(z,\mathcal{D}) - \log q(z;\lambda)],$$

with respect to the variational parameter λ. This approach is often preferred to avoid computing the intractable normalising constant of $p(z|\mathcal{D})$.

The closed-form solution for λ^* is generally unavailable. Stochastic VI [11], which optimises with minibatch SGD, has revolutionised and broadened the applications of VI. It necessitates computing the gradient of the *minibatch ELBO*, denoted as $\text{mELBO}(\lambda) = \mathbb{E}_{q(z;\lambda)}\left[r(z;\lambda)\right]$, where

$$r(z;\lambda) = \frac{N}{B}\sum_{i\in\text{batch}}\left[\log p(x_i|z)\right] + \log\frac{p(z)}{q(z;\lambda)}. \tag{1}$$

Here N and B are the data and batch size respectively. One challenge in stochastic VI is computing the gradient of mELBO, $\nabla_\lambda \mathbb{E}_{q(z;\lambda)}[r(z;\lambda)]$. As the gradient is taken with respect to the parameter of q, we cannot simply push the gradient operator through the expectation, making the computation non-trivial. In the VI literature, there are two main types of gradient estimators for computing $\nabla_\lambda \mathbb{E}_{q(z;\lambda)}[r(z;\lambda)]$: 1) the **pathwise gradient estimator**, also known as the reparametrization trick; and 2) the **score-function estimator**, or REINFORCE. The latter has broader applicability but often comes with higher variance. Indeed, the score-function estimator is almost always used in conjunction with control variates to reduce its variance [18, 12]. While variance reduction for the pathwise gradient estimator is less common, recent work suggests it may be beneficial [14, 7]. In this work, we are primarily interested in reducing variance of the pathwise gradient estimator.

The pathwise gradient estimator is readily applicable only to *reparametrizable* distributions $q(z;\lambda)$. These are distributions where we can generate z equivalently from a transformation $z = T(\epsilon;\lambda)$, where $\epsilon \in \mathbb{R}^{\dim z} \sim q_0(\epsilon)$ and q_0 is referred to as the *base distribution*, independent of λ. For example, a Gaussian distribution $z \sim \mathcal{N}(\mu, \sigma^2 I)$ and its corresponding transformation is $T(\epsilon;\lambda) = \mu + \sigma\epsilon$, where ϵ follows a standard Gaussian distribution and $\lambda = (\mu, \sigma)$. When q is reparametrizable, we can push the gradient operator inside the expectation, giving us the gradient of mELBO as

$$g(\lambda) \coloneqq \nabla_\lambda \,\text{mELBO}(\lambda) = \mathbb{E}_{q_0(\epsilon)}\varphi(\epsilon;\lambda), \tag{2}$$

where we define $\varphi(\epsilon;\lambda) = \nabla_\lambda\left[r(T(\epsilon;\lambda);\lambda)\right]$. The pathwise gradient estimator is then a Monte Carlo estimator of (2) using samples $\{\epsilon_{[l]}\}_{l=1}^L$ from q_0

$$\hat{g}(\epsilon_{[1]},\ldots,\epsilon_{[L]};\lambda) \coloneqq \frac{1}{L}\sum_{l=1}^{L}\varphi(\epsilon_{[l]};\lambda). \tag{3}$$

We will refer to L as the number of gradient samples.

The variance of the gradient estimator $\mathbb{V}[\hat{g}] = \mathbb{E}\|\hat{g}\|^2 - \|\mathbb{E}g\|^2 = \frac{1}{L}\mathbb{V}[\varphi]$ is thought to play a significant role in the convergence properties of the mELBO optimizer. Here, the expectations and variances are taken with respect to q_0 — from this point on, any expectations or variances without a subscript refer to q_0. To reduce the variance of (3), we can add a *control variate* (CV), $c(\cdot) \in \mathbb{R}^{\dim\lambda}$, to the pathwise gradient estimator

$$\frac{1}{L}\sum_{l=1}^{L}\left[\varphi(\epsilon_{[l]};\lambda) + c(\epsilon_{[l]})\right], \tag{4}$$

where the CV is a random variable with zero expectation, that is, $\mathbb{E}[\varphi + c] = \mathbb{E}\varphi$. Let $\mathrm{Tr}(\mathbb{C}[\varphi, c])$ denote the trace of the covariance matrix $\mathbb{C}[\varphi, c] = \mathbb{E}[(\varphi - \mathbb{E}\varphi)(c - \mathbb{E}c)^\top]$. A good CV should exhibit a strong, negative correlation with φ, since

$$\mathbb{V}[L^{-1} \sum_{l=1}^{L} \varphi(\epsilon_{[l]}; \lambda) + c(\epsilon_{[l]})] = \mathbb{V}[\hat{g}] + L^{-1}(\mathbb{V}[c] + 2\,\mathrm{Tr}(\mathbb{C}[\varphi, c])) \tag{5}$$

Therefore, as long as $\mathrm{Tr}(\mathbb{C}[\varphi, c]) < 0$ and $|\,\mathrm{Tr}(\mathbb{C}[\varphi, c])| < \frac{1}{2}\mathbb{V}[c]$, the CV-adjusted gradient estimator (4) will exhibit a smaller variance than (3). Finally, we can form a CV as a linear combination of multiple CVs. Let $C : \mathbb{R}^{\dim_z} \to \mathbb{R}^{\dim_\lambda \times J}$ be a matrix with J CVs as its columns. This leads to the CV-adjusted gradient estimator

$$\hat{g}(\epsilon_{[1]}, \ldots, \epsilon_{[L]}; \lambda) \coloneqq \frac{1}{L} \sum_{l=1}^{L} \left[\varphi(\epsilon_{[l]}; \lambda) + C(\epsilon_{[l]})\beta\right], \tag{6}$$

which remains a valid CV due to the linearity of expectation operators. Here $\beta \in \mathbb{R}^J$ is a vector of coefficients corresponding to each CV. This construction allows us to combine several weak CVs into a stronger one by adjusting β.

For obvious reasons, applying CV is only worthwhile if its computation is cheaper than increasing the number of samples in (3). From (5), we see that the estimator variance can be halved either by doubling L or halving $\mathbb{V}[\varphi + c]$. This poses a unique challenge when applying CV in the low L regime (common in VI where L is often very low), as the cost of CV may outweigh the cost of increasing L for the same variance reduction. CVs developed for Markov Chain Monte Carlo (MCMC) do not easily apply here because 1) they require large L, but L can be as small as one in stochastic VI, and 2) MCMC variance reduction is typically done at the very end, whereas in stochastic VI, it's needed for each gradient update.

Contribution This work reviews existing CV-adjusted pathwise gradient estimators in the context of VI, primarily examining whether employing CV leads to faster convergence in terms of wall-clock time. We are also motivated by the gap in VI literature on gradient variance reduction when q is reparametrizable, but its mean and covariance are not available in closed form. A good example is normalizing flow, where z is the result of pushing a base distribution q_0 through an invertible transformation $T(\cdot; \lambda)$ parameterised by λ, i.e. $z = T(\epsilon; \lambda)$ where $\epsilon \sim q_0$. This transformation can be arbitrarily complex and often involves neural networks. To address this, we introduce a CV-adjusted gradient estimator based on zero-variance control variates (ZVCV) [1, 15, 16], which doesn't have this limitation.

This paper is structured as follows: Section 2 reviews the latest advancements in variance reduction techniques for pathwise gradient estimators in VI. Sections 3 and 4 discuss methods for selecting β and C respectively. The novel ZVCV-based method is introduced in Section 4.2. Finally, experimental results are presented in Section 5.

2 Related Work

Variance reduction for the pathwise gradient estimator in VI has been explored in [14] and [7]. These works focused on designing a single CV (i.e. C has only one column) with the form $C = \mathbb{E}\tilde{\varphi} - \tilde{\varphi}$, where $\tilde{\varphi}(\epsilon; \lambda)$ approximates $\varphi(\epsilon; \lambda)$. The expectation $\mathbb{E}\tilde{\varphi}$ is intended to be theoretically tractable, but this usually places restrictions on T (and therefore, q).

For instance, [14] proposed a $\tilde{\varphi}$ based on the first-order Taylor expansion of $\nabla_z \log p(z, \mathcal{D})$. However, this necessitates the expensive computation of the Hessian $\nabla_z^2 \log p(z, \mathcal{D})$. [7] improved upon this by using a quadratic function to approximate $\log p(z, \mathcal{D})$. Their method only requires the first-order gradient $\nabla_z \log p(z, \mathcal{D})$, and their $\mathbb{E}\tilde{\varphi}$ has a closed-form solution as a function of the mean and covariance of q. This method can be further extended to accommodate q without a closed-form mean and covariance by estimating $\mathbb{E}\tilde{\varphi}$ empirically; see Section 4.1 for details. In both of these work, they focused on Gaussian q.

Our proposed estimator based on ZVCV shares similarities with another work from the same group in [7]. Like [6], we propose combining weak CVs into a stronger one. However, our work differs in how we construct individual CVs and the optimisation criterion for β.

3 Selecting β for CV-adjusted pathwise gradient estimators

The utility of CV depends on the choice of β and C in (6). In this section, we will discuss various strategies to pick an appropriate β given a family of C.

3.1 A unique set of β for each dimension of λ

The formulation in (6) suggests that the same set of β is used across the dimensions of φ. This can be too restrictive for C that are weakly-correlated to φ. In such instance, having a unique set of β coefficients for each dimension of φ can be beneficial, as it allows the coefficients to be selected on a per-dimension basis. In fact, this can be easily done by turning C into a $\dim_\lambda \times (J\dim_\lambda)$-dimensional, block diagonal matrix $\mathrm{diag}(C_{1,:}, \ldots, C_{\dim_\lambda,:})$, where $C_{i,:}$ is the i^{th} row of the original C. In other words, we expand the number of CV to $J\dim_\lambda$, and each CV will only reduce the variance of a single dimension of φ.

3.2 Optimisation criteria for β

The β is usually chosen to minimise the variance of $\hat{h}$. In practice, this variance is evaluated empirically due to the lack of its closed-form expression. There are three approaches in the literature, the first of which is a direct approximation of the variance with samples $\{\epsilon_{[l]}\}_{l=1}^L$,

$$\mathbb{V}[\varphi + C\beta] \approx \frac{1}{L(L-1)} \sum_{l>l'} \|\varphi(\epsilon_{[l]}) + C(\epsilon_{[l]})\beta - \varphi(\epsilon_{[l']}) - C(\epsilon_{[l']})\beta\|^2,$$

as seen in [2]. The second approach is based on the definition of variance

$$\mathbb{V}[\varphi + C\beta] = \mathbb{E}\|\varphi - \mathbb{E}[\varphi + C\beta] + C\beta\|^2 \approx \min_{\alpha \in \mathbb{R}^{\dim_\lambda}} \frac{1}{L}\sum_{l=1}^{L} \|\varphi(\epsilon_{[l]}) + \alpha + C(\epsilon_{[l]})\beta\|^2, \tag{7}$$

where $\alpha \in \mathbb{R}^{\dim_\lambda}$ is an intercept term in place of the unknown $\mathbb{E}[\varphi + C\beta]$. This is cheaper to compute than the former as it only requires $O(L)$ operations rather than $O(L^2)$ [19]. Finally, the third approach relies on the assumption that $\mathbb{E}[C\beta] = 0$ and is based on the observation that $\mathbb{V}[\varphi + C\beta] = \mathbb{E}\|\varphi + C\beta\|^2 - \|\mathbb{E}\varphi\|^2$. This suggests that $\mathbb{V}[\varphi + C\beta]$ can be equivalently minimised by solving $\beta^* = \arg\min_\beta \mathbb{E}\|\varphi + C\beta\|^2$, the solution of which is given

$$\beta^* = -\mathbb{E}[C^\top C]^{-1}\mathbb{E}[C^\top \varphi]. \tag{8}$$

See [6] for the derivation. This approach, however, often requires estimating the expectations empirically and performing a costly inversion of size J matrix.

In the development of our CV, we focus on the second approach (7) as this is generally the cheapest among the three. This approach is also equivalent to solving a linear least squares problem

$$\begin{bmatrix}\alpha^* \\ \beta^*\end{bmatrix} = \arg\min_{\alpha,\beta} \sum_{l=1}^{L} \left\| \varphi(\epsilon_{[l]}) + \begin{bmatrix}\mathbb{I}_{\dim_\lambda} & C(\epsilon_{[l]})\end{bmatrix} \begin{bmatrix}\alpha \\ \beta\end{bmatrix} \right\|^2, \tag{9}$$

and has a unique and closed-form solution when the corresponding design matrix is full-column rank. Even when such condition is not satisfied, a penalty term can be added to the objective function of (9) and we end up with a penalised least squares problem [21]. Alternatively, we can solve (9) with an iterative optimisation algorithm to obtain a (non-unique) solution [19].

4 Control variates

Having reviewed methods to select β given a family of C, we now turn our attention to constructing C. We will first propose a simple modification of [7] that will enable it to work for variational distributions q with unknown mean and covariance. Subsequently, we will introduce ZVCV, which can be constructed without the knowledge of q or T.

4.1 Quadratic approximation control variates

In this section, we review the quadratic-approximation CV proposed in [7]. An important distinction at the outset is their assumption that the entropy term in mELBO, $-\mathbb{E}_{q(z)} \log q(z;\lambda)$, is *known*. As such the focus of [7] is to reduce the variance of $\mathbb{E}\nabla_\lambda f(T(\epsilon;\lambda))$, where

$$f(z) = \frac{N}{B}\sum_{i \in \text{batch}} [\log p(x_i|z)] + \log p(z). \tag{10}$$

The CV is based on the quadratic approximation of (10), $\tilde{f}(z;v) = b_v^\top(z-z_0) + \frac{1}{2}(z-z_0)^\top B_v(z-z_0)$, and has the form of

$$C(\epsilon) = \mathbb{E}[\nabla_\lambda \tilde{f}(T(\epsilon;\lambda))] - \nabla_\lambda \tilde{f}(T(\epsilon;\lambda)). \tag{11}$$

Here, $v = \{B_v, b_v\}$ are the parameters of the quadratic equation that are chosen to minimise the L^2 difference between $\nabla f(z)$ and $\nabla \tilde{f}(z)$. We will drop v in $\tilde{f}$ for the sake of brevity. The location parameter z_0 is set to $\mathbb{E}T(\epsilon;\lambda)$. This quadratic approximation of f can also be viewed as a linear approximation on ∇f.

The first term in (11) has a closed-form expression that depends on the mean and covariance of $q(z;\lambda)$, making the expectation cheap to evaluate when they are readily available. However, this is not the case when $T(\epsilon;\lambda)$ is arbitrarily complex, e.g. normalizing flow. A direct workaround of this limitation is to replace $\mathbb{E}\nabla_\lambda \tilde{f}(T(\epsilon;\lambda))$ with its empirical estimate based on samples of ϵ. Note that $\tilde{f}$ requires $z_0 = \mathbb{E}T(\epsilon;\lambda)$, which we estimate using another independent set of ϵ. See Algorithm 1 for a summary of the procedure.

As (10) is a part of the Monte Carlo estimator (6), it could be tempting to estimate $\mathbb{E}\nabla_\lambda \tilde{f}(T(\epsilon;\lambda))$ with an average of the $\nabla_\lambda \tilde{f}(T(\epsilon;\lambda))$ evaluations that have been computed in (6). This is to be avoided as it will result in the two terms in (11) cancelling each other out.

As (11) is designed to be strongly correlated with φ when $\tilde{f}$ is reasonably close to f, the choice of β becomes less significant. [7] opted to minimise the expected squared norm with a scalar β (note that C is a column vector in this case), the solution of which is given in (8). In their work, the expectations $\mathbb{E}[C^\top\varphi]$ and $\mathbb{E}[C^TC]$ are replaced with their empirical estimates computed from C and φ in (6), instead of fresh evaluations. However, the resulting gradient estimate is biased due to the dependency of between β and C, as $\mathbb{E}[C(\epsilon)\beta(\epsilon)] \neq 0$ in general.

While this bias is not mentioned explicitly in [7], we conjecture that they overcame the issue by estimating the expectations with C and φ computed from previous iterations, as specified in Algorithm 1. Therefore, their β is independent from C in the current iteration. This will avoid introducing bias to the gradient estimate at the cost of having sub-optimal β. They also claimed that their estimates of $\mathbb{E}[C^\top\varphi]$ and $\mathbb{E}[C^TC]$ (and by extension, β) do not differ much across iterations. Moreover, their β is largely acting as an auxiliary 'switch' of the CV when $\tilde{f}$ is a poor approximation of f, rather than the primary mechanism to reduce the estimator variance, since the β will be almost 0 when $\tilde{f}$ is not approximating well (i.e. $\mathbb{C}[\varphi, C] \approx 0$). Their C only kicks in when it is sufficiently correlated to φ.

Lastly, we return to the discussion of the entropy term at the beginning of this section. Our setup is more general than [7] as we does not assume the entropy term $-\mathbb{E}_{q(z)} \log q(z;\lambda)$ to necessarily have a closed-form expression, i.e. our $r(z,\lambda)$ includes $-\log q(z;\lambda)$. Although it was claimed in [7] that their quadratic approximation CV can also be similarly designed for $r(z,\lambda)$ in (1) rather than $f(z,\lambda)$ in (10), we found the implementation difficult because the updating step of v requires the gradient $\nabla_z \log q(z;\lambda)$, and in turn $\frac{\partial\lambda}{\partial z}$, which is challenging to compute.

4.2 Zero-variance control variates

The CV in [7] require one to know the mean and covariance of $q(z;\lambda)$. To avoid this requirement, we propose the use of gradient-based CV [1, 15, 16]. These CV are generated by applying a Stein operator $\mathcal{L}$ to a class of user-specified functions $P(z)$. Typically the Stein operator uses $\nabla_z \log q(z)$, the gradients of the log probability density function for the distribution over which the expectation is taken, but it does not require any other information about φ or T.

We will focus on the form of gradient-based CV known as ZVCV [1, 15]. ZVCV uses the second order Langevin Stein operator and a polynomial $P(z) = \sum_{j=1}^{J} \beta_j P_j(z)$, where $P_j(z)$ is the jth monomial in the polynomial and J is the number of monomials. The CV are

$$\{\mathcal{L}P_j(z)\}_{j=1}^{J} = \{\Delta_z P_j(z) + \nabla_z P_j(z) \cdot \nabla_z \log q(z)\}_{j=1}^{J},$$

where Δ_z is the Laplacian operator and $q(z)$ is the probability density function for the distribution over which the expectation is taken. A sufficient condition for these CV to have zero expectation is that the tails of q decay faster than a polynomial rate [17, Appendix B], which is satisfied by Gaussian q for example.

In this paper, we only consider first-order polynomials, so there are $J = \dim_z$ CV of the form $\left\{\frac{\partial}{\partial z_j} \log q(z)\right\}_{j=1}^{\dim_z}$. Here, z_j refers to the j^{th} dimension of z. We do not find second-order polynomials to have any advantage over first-order polynomials; see Appendix G for a discussion. For pathwise gradient estimators using a standard Gaussian as the base distribution, these CV simplify further to $\{-\epsilon_j\}_{j=1}^{\dim_z}$. We are also using the same set of CV across different dimensions of φ, but assigning each dimension with a unique set of β. That is, the matrix C is a block-diagonal matrix $C(\epsilon) = \text{diag}(-\epsilon^\top, \ldots, -\epsilon^\top)$ of size $\dim_\lambda \times \dim_\lambda \dim_z$. This is in contrast to [7] where the values in C is different across dimensions, but their β is shared. The simplicity of ZVCV comes with the drawback that it is often not as correlated as the integrand. This makes the choice of β crucial.

Estimating $\boldsymbol{\beta}$ with the least squares approach As discussed in Section 3, the unique, closed-form solution of (9) requires the corresponding design matrix to have a full column rank. In our application where the models often has $\dim_z$ much greater than L, this is not the case as the corresponding design matrix is very wide. While this problem can be solved by adding a penalty term in (9) to shrink β towards 0 [6, 21], solving penalised least squares remains prohibitively expensive as it still involves inverting a matrix of size $\dim_z$. Instead, we propose mimicking penalised least squares by minimising (9) with respect to α and β with gradient descent. This is done by

1. Initialise α at $-\frac{1}{L}\sum_{l=1}^{L} \varphi(\epsilon_{[l]})$ and β at the zero vector. Set $\gamma^{(\alpha,\beta)}$ to a low value;
2. Take a descent step $(\alpha_{m+1}, \beta_{m+1}) \leftarrow (\alpha_m, \beta_m) - \gamma^{(\alpha,\beta)} \frac{1}{L} \sum_{l=1}^{L} \nabla_{\alpha,\beta} \|\varphi(\epsilon_{[l]}) + \alpha_m + C(\epsilon_{[l]})\beta_m\|^2$;
3. Repeat Step 2 for a few times.

See Algorithm 2 for a complete description. The combination of learning rate and number of iterations is analogous to the penalty in penalised least squares: a lower number of iterations and learning rate $\gamma^{(\alpha,\beta)}$ will result in a near-zero β that results from a stronger penalty (more shrinkage of β towards 0). This procedure is also similar in spirit to [19].

5 Experiments

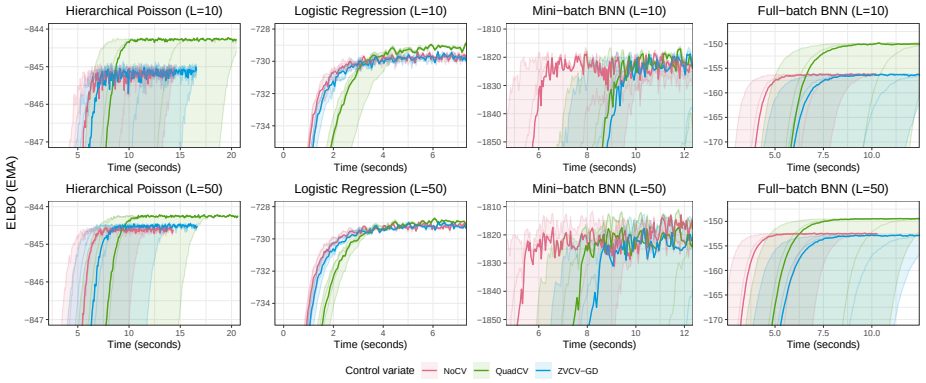
In these experiments, we assess the efficacy of various gradient estimators by performing VI on the following model-dataset pairs: logistic regression on the *a1a* dataset, a hierarchical Poisson model on the *frisk* dataset, and a Bayesian neural network (BNN) on a subset of the *redwine* dataset. For the BNN model, we consider both full-batch and mini-batch (size 32) gradient estimators. We utilise diagonal and low-rank Gaussian distributions, as well as Real NVP [4] as our variational family q. Additionally, we vary the number of gradient samples, setting $L = 10$ and 50, and compare three types of estimators: the vanilla estimator without any CV (NoCV), ZVCV-GD as described in Section 4.2, and QuadCV proposed by [7]. We report the ELBO, wall-clock time, and variance of the gradient estimators. Comprehensive setup details are provided in Appendix B.

ELBO against wall-clock time To assess whether the computational expense of calculating CV or additional gradient samples justifies the potential improvement in ELBO, we measure ELBO against wall-clock time, as illustrated in Figure 1. Our experiments reveal that NoCV generally converges to a respectable ELBO more swiftly. Furthermore, the performance gap between the estimators is even narrower when $L = 50$. An unexpected observation is that increasing L from 10 to 50 incurs negligible computational cost but produce meaningfully faster convergence, as evident when comparing the top and bottom rows of Figure 1a and 1b. It is important to note that the computational cost of extra gradient samples may vary depending on the construction of φ, and increasing L might not always be a worthwhile strategy for achieving faster convergence (see, for example, the BNNs experiments in Figure 4b of Appendix D).

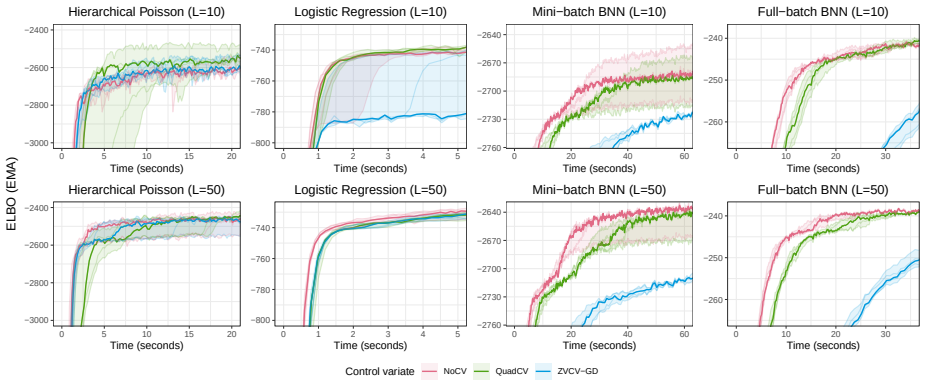
QuadCV does succeed in increasing the maximum achievable ELBO in certain scenarios, albeit at the expense of longer convergence times. For instance, QuadCV can improve ELBO by approximately 0.7 nats and 6 nats in hierarchical Poisson and full-batch BNN when using a mean-field Gaussian q at $L = 10$. However, this comes at a cost of roughly 50% to 100% more runtime compared to NoCV. Given finite computational resources and the absence of a universal guarantee that a slight ELBO increase will substantially enhance downstream metrics [24, 23, 5, 13, 3], it is left to practitioners to determine whether implementing CV is a worthwhile endeavour.

Additional experiments In addition to our main results, we present further findings in the Appendix to explore the efficacy of CV under various settings.

Specifically, Appendix C examines ELBO and variance reduction against iteration count, Appendix D explores low-rank Gaussian as q, Appendix E presents mean curves of ELBO, Appendix F provides an individual analysis of each curve, and Appendix G investigates different variants of ZVCV. Overall, we find that while CV can reduce variance in the gradient estimator, the computational overhead does not justify its implementation compared to simply increasing the number of gradient samples.



(a) Mean-field Gaussian with 10 (top) and 50 (bottom) gradient samples.



(b) Real NVP with 10 (top) and 50 (bottom) gradient samples.

Fig. 1: ELBO is plotted against wall-clock time for different numbers of gradient samples L and two families of q. The bold lines represent the median of ELBO values recorded at the same iteration across five repetitions. The shaded area illustrates the range of ELBO values across five repetitions. The ELBO values are smoothed using an exponential moving average. A higher ELBO indicates better performance. See Figure 8 for plots where the bold lines represent the mean ELBO.

6 Conclusion

In our study of the pathwise gradient estimator in VI, we reviewed the state-of-the-art CV for reducing gradient variance, namely the QuadCV in [7]. We identified a gap in the literature regarding variance reduction of pathwise gradient estimators in stochastic VI when the variational distribution has intractable mean and covariance, making QuadCV not directly applicable. To address this, we proposed using ZVCV, which does not assume specific conditions on the variational distribution.

However, our empirical results showed that neither the ZVCV-adjusted nor the QuadCV-adjusted estimator provided substantial improvement in our evaluation metrics to justify their implementation. Instead, we found that simply increasing the number of gradient samples was highly effective for improving convergence time.

Stepping back, it is worth discussing the fundamental value of variance reduction for pathwise gradient estimators in stochastic VI. Interestingly, a dramatic reduction in gradient variance may not lead to any noticeable effect on the ELBO. This was observed in our experiments — even with a substantially lower variance, the CV-adjusted gradient estimator did not meaningfully improve the ELBO optimization objective compared to the vanilla gradient estimator. We can thus expect that downstream metrics, such as log predictive density, will also reveal the general ineffectiveness of equipping the gradient estimator with a CV. These findings seem to indicate a negative phenomenon for pathwise gradients in stochastic VI: reducing gradient variance alone is insufficient to improve downstream performance.

In future work, we hope to explore ZVCV-adjusted gradient estimators in generative models where they may excel. ZVCV is particularly powerful when the distribution of interest is difficult to sample from, such as in energy-based models [20]. Additionally, implicit VI methods require the variational distribution to be reparametrizable but not the pathwise score, $\nabla_z \log q(z;\lambda)$, to be known (e.g. as in normalizing flows). [22] showed that the pathwise score can be written as an expectation, $\nabla_z \log q(z;\lambda) = E_{q(\epsilon|z;\lambda)} \nabla_z \log q(z|\epsilon;\lambda)$, where $q(\epsilon|z;\lambda)$ is the reverse conditional. In [22], the expectation with respect to the reverse conditional is based on MCMC samples. We could potentially improve efficiency by employing ZVCV here.

7 Acknowledgment

The authors would like to thank Leah South for her valuable discussions on the project. KN was supported by the Australian Government Research Training Program Scholarship and the Fred Knight Scholarship. SW was supported by the ARC Discovery Early Career Researcher Fellowship (DE200101253).

Bibliography

[1] Assaraf, R., Caffarel, M.: Zero-Variance Principle for Monte Carlo Algorithms. Physical Review Letters **83**(23), 4682–4685 (Dec 1999). https://doi.org/10.1103/PhysRevLett.83.4682

[2] Belomestny, D.V., Iosipoi, L.S., Zhivotovskiy, N.K.: Variance Reduction in Monte Carlo Estimators via Empirical Variance Minimization. Doklady Mathematics **98**(2), 494–497 (Sep 2018). https://doi.org/10.1134/S1064562418060261

[3] Deshpande, S., Ghosh, S., Nguyen, T.D., Broderick, T.: Are you using test log-likelihood correctly? In: I Can't Believe It's Not Better Workshop: Understanding Deep Learning Through Empirical Falsification (Dec 2022)

[4] Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using Real NVP. In: International Conference on Learning Representations (2017)

[5] Foong, A., Burt, D., Li, Y., Turner, R.: On the Expressiveness of Approximate Inference in Bayesian Neural Networks. In: Advances in Neural Information Processing Systems. vol. 33, pp. 15897–15908. Curran Associates, Inc. (2020)

[6] Geffner, T., Domke, J.: Using Large Ensembles of Control Variates for Variational Inference. In: Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018)

[7] Geffner, T., Domke, J.: Approximation Based Variance Reduction for Reparameterization Gradients. In: Advances in Neural Information Processing Systems. vol. 33, pp. 2397–2407. Curran Associates, Inc. (2020)

[8] Gelman, A., Fagan, J., Kiss, A.: An Analysis of the New York City Police Department's "Stop-and-Frisk" Policy in the Context of Claims of Racial Bias. Journal of the American Statistical Association **102**(479), 813–823 (Sep 2007). https://doi.org/10.1198/016214506000001040

[9] Gelman, A., Stern, H.S., Carlin, J.B., Dunson, D.B., Vehtari, A., Rubin, D.B.: Bayesian Data Analysis. CRC Press, third edn. (2013)

[10] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. pp. 249–256. JMLR Workshop and Conference Proceedings (Mar 2010)

[11] Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.: Stochastic Variational Inference. Journal of Machine Learning Research **14**(40), 1303–1347 (2013)

[12] Ji, G., Sujono, D., Sudderth, E.B.: Marginalized Stochastic Natural Gradients for Black-Box Variational Inference. In: Proceedings of the 38th International Conference on Machine Learning. pp. 4870–4881. PMLR (Jul 2021)

[13] Masegosa, A.: Learning under Model Misspecification: Applications to Variational and Ensemble methods. In: Advances in Neural Information Processing Systems. vol. 33, pp. 5479–5491. Curran Associates, Inc. (2020)

[14] Miller, A., Foti, N., D' Amour, A., Adams, R.P.: Reducing Reparameterization Gradient Variance. In: Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
[15] Mira, A., Solgi, R., Imparato, D.: Zero variance Markov chain Monte Carlo for Bayesian estimators. Statistics and Computing **23**(5), 653–662 (Sep 2013). https://doi.org/10.1007/s11222-012-9344-6
[16] Oates, C.J., Girolami, M., Chopin, N.: Control functionals for Monte Carlo integration. Journal of the Royal Statistical Society. Series B (Statistical Methodology) **79**(3), 695–718 (2017)
[17] Oates, C.J., Papamarkou, T., Girolami, M.: The Controlled Thermodynamic Integral for Bayesian Model Evidence Evaluation. Journal of the American Statistical Association **111**(514), 634–645 (Apr 2016). https://doi.org/10.1080/01621459.2015.1021006
[18] Ranganath, R., Gerrish, S., Blei, D.M.: Black Box Variational Inference. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics. pp. 814–822. PMLR (Apr 2014)
[19] Si, S., Oates, Chris.J., Duncan, A.B., Carin, L., Briol, F.X.: Scalable Control Variates for Monte Carlo Methods Via Stochastic Optimization. In: Keller, A. (ed.) Monte Carlo and Quasi-Monte Carlo Methods. pp. 205–221. Springer Proceedings in Mathematics & Statistics, Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-030-98319-2_10
[20] Song, Y., Kingma, D.P.: How to Train Your Energy-Based Models (Feb 2021)
[21] South, L.F., Oates, C.J., Mira, A., Drovandi, C.: Regularized Zero-Variance Control Variates. Bayesian Analysis -**1**(-1), 1–24 (Jan 2022). https://doi.org/10.1214/22-BA1328
[22] Titsias, M.K., Ruiz, F.: Unbiased Implicit Variational Inference. In: Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics. pp. 167–176. PMLR (Apr 2019)
[23] Yao, J., Pan, W., Ghosh, S., Doshi-Velez, F.: Quality of Uncertainty Quantification for Bayesian Neural Network Inference (Jun 2019). https://doi.org/10.48550/arXiv.1906.09686
[24] Yao, Y., Vehtari, A., Simpson, D., Gelman, A.: Yes, but Did It Work?: Evaluating Variational Inference. In: Proceedings of the 35th International Conference on Machine Learning. pp. 5581–5590. PMLR (Jul 2018)

A Implementation details

We present the implementation details and pseudocode in this section.

A.1 Potential bias in the gradient estimator

The unbiasedness of the CV-adjusted Monte Carlo estimator (6) relies on the assumption that the β are independent of C, since $\mathbb{E}[C(\epsilon)\beta(\epsilon)] \neq 0$ in general. This necessitates that β and C should be estimated with independent sets of ϵ samples. However, in practice, the β is estimated with the same set of ϵ in C to save computational time at the cost of introducing bias in the gradient estimates.

Algorithm 1 Quadratic approximation control variates with empirical estimates of $\mathbb{E}\tilde{f}$

Require: Learning rates $\gamma^{(\lambda)}$, $\gamma^{(v)}$.
  Initialise λ, v and control variate weight $\beta = 0$.
  for $k = 0, 1, 2, \cdots$ **do**
    Sample $\epsilon_{[1]}, \ldots \epsilon_{[L]} \sim q_0$ to compute $\varphi(\epsilon_{[l]}; \lambda_k)$
    Generate an independent set of 100 ϵ samples to estimate $z_0 = \mathbb{E}T(\epsilon; \lambda)$
    Generate an independent set of 100 ϵ samples to estimate $\mathbb{E}\nabla_\lambda \tilde{f}(T(\epsilon; \lambda); v_k)$
    Compute $h = \frac{1}{L}\sum_{l=1}^{L}\left[\varphi(\epsilon_{[l]}; \lambda_k) + C(\epsilon_{[l]})\beta\right]$ See (11)
    Take an ascent step $\lambda_{k+1} \leftarrow \lambda_k + \gamma^{(\lambda)} h$
    Estimate $\mathbb{E}[C^\top C]$ and $\mathbb{E}[C^\top \varphi]$ with $\epsilon_{[1]}, \ldots \epsilon_{[L]}$, and update β with (8).
    Take a descent step $v_{k+1} \leftarrow v_k - \gamma^{(v)} \frac{1}{2L}\sum_{l=1}^{L} \nabla_v \|\nabla_z f(T(\epsilon_{[l]}; \lambda_k)) - \nabla_z \tilde{f}(T(\epsilon_{[l]}; \lambda_k); v_k)\|^2$
  end for

B Experiment Setup

B.1 Models and datasets

We perform VI on the following model-dataset pairs: logistic regression on the *a1a* dataset, a hierarchical Poisson model on the *frisk* dataset, and Bayesian neural network (BNN) on the *redwine* dataset. For the BNN model, we consider a full-batch gradient estimator trained on a subset of 100 data points of the *redwine* dataset following the experimental setup in [7] and [14]. We also consider a mini-batch estimator of size 32 but trained on the full *redwine* datasets. With the exception of mini-batch BNN, these models appeared in either [7] or [14].

Algorithm 2 ZVCV-GD

Require: Learning rates $\gamma^{(\lambda)}$, $\gamma^{(\alpha,\beta)}$.
  Initialise λ
  for $k = 0, 1, 2, \cdots$ **do**
    Sample $\epsilon_{[1]}, \ldots \epsilon_{[L]} \sim q_0$
    Compute $\varphi(\epsilon_{[l]}; \lambda_k), \forall l = 1, \ldots, L$ See (2)
    Initialise $\alpha_0 = -\frac{1}{L}\sum_{l=1}^{L} \varphi(\epsilon_{[l]}; \lambda_k)$ and β_0 at the zero vector
    for $m = 0, 1, 2, \cdots$ **do**
      Take a descent step $(\alpha_{m+1}, \beta_{m+1}) \leftarrow (\alpha_m, \beta_m) - \gamma^{(\alpha,\beta)} \frac{1}{L}\sum_{l=1}^{L} \nabla_{\alpha,\beta} \|\varphi(\epsilon_{[l]}; \lambda_k) + \alpha_m + C(\epsilon_{[l]})\beta_m\|^2$
    end for
    Set β^* as the final value of β from the previous inner loop
    Compute $h = \frac{1}{L}\sum_{l=1}^{L} \varphi(\epsilon_{[l]}; \lambda_k) + C(\epsilon_{[l]})\beta^*$
    Take an ascent step $\lambda_{k+1} \leftarrow \lambda_k + \gamma^{(\lambda)} h$.
  end for

Logistic regression with the a1a dataset We extracted the $a1a$ dataset from the repository hosting [7]. We used the full dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^{1605}$ and 90% of the dataset for training. The response y_i is binary and is modelled as

$$\begin{aligned} w_0, \boldsymbol{w} &\sim \mathcal{N}(0, 10^2) \\ p(y_i|\boldsymbol{x}_i, z) &= \text{Bernoulli}\left(\frac{1}{1+\exp(-w_0 - \boldsymbol{w}^T\boldsymbol{x}_i)}\right), \end{aligned}$$

where $z = \{w_0, \boldsymbol{w}\}$ and $\dim_z = 120$. The size of training and test sets are 1440 and 165 respectively.

Hierarchical Poisson regression with the frisk dataset This example is coming from [8]. We only used a subset of data (weapon-related crime, precincts with 10%-40% of black proportion), as in [14] and [7]. The response y_{ep} denotes the number of frisk events due to weapons crimes within an ethnicity group e in precinct p over a 15-months period in New York City:

$$\begin{aligned} \mu &\sim \mathcal{N}(0, 10^2) \\ \log\sigma_\alpha, \log\sigma_\beta &\sim \mathcal{N}(0, 10^2) \\ \alpha_e &\sim \mathcal{N}(0, \sigma_\alpha^2) \\ \beta_p &\sim \mathcal{N}(0, \sigma_\beta^2) \\ \log\lambda_{ep} &= \mu + \alpha_e + \beta_p + \log N_{ep} \\ p(y_{ep}|z) &= \text{Poisson}(\lambda_{ep}), \end{aligned}$$

where $z = \{\alpha_1, \alpha_2, \beta_1, \ldots, \beta_{32}, \mu, \log\sigma_\alpha, \log\sigma_\beta\}$ and $\dim_z = 37$. N_{ep} is the (scaled) total number of arrests of ethnicity group e in precinct p over the same period of time. We do not split out a test set due to its small size (total data size is 96).

Bayesian neural network with the redwine dataset We push a vector input $\boldsymbol{x}_i$ through a 50-unit hidden layer and ReLU activation's to predict wine quality. The response y_i is an integer from 1 to 10 (inclusive) measuring the score of red wine. We place an uniform improper prior on the log-variance of the weights and error [9, Section 5.7]:

$$\begin{aligned}
&p(\log \alpha^2) \propto 1, \quad \text{equivalent to } p(\alpha) \propto \alpha^{-1} \\
&p(\log \tau^2) \propto 1, \quad \text{equivalent to } p(\tau) \propto \tau^{-1} \\
&w_i \sim \mathcal{N}(0, \alpha^2), \quad i = 1, \ldots, 651 \\
&y_i | \boldsymbol{x}_i, \boldsymbol{w}, \tau \sim \mathcal{N}(\phi(\boldsymbol{x}, \boldsymbol{w}), \tau^2)
\end{aligned}$$

where ϕ is a multi-layer perception. Here, $z = \{\log \alpha^2, \log \tau^2, \boldsymbol{w}\}$ and $\dim_z = 653$. For full-batch gradient descent, we use two mutually exclusive subsets of 100 data point as train and test sets, as in [14] and [7]. For mini-batch gradient descent, we use 90% of the full dataset for training and the rest for testing (size of train and test sets are 1431 and 168 respectively).

B.2 Variational families

Three classes of variational families are considered:

- **Mean-field Gaussian** The covariance of the Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ is parameterised by log-scale parameters, i.e. $\Sigma = \text{diag}\left(\exp(2 \log \sigma_1, \ldots, 2 \log \sigma_{\dim_z})\right)$.
- **Rank-5 Gaussian** The covariance of the Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ is parameterised by a factor $F \in \mathbb{R}^{\dim_z \times 5}$ and diagonal components, i.e. $\Sigma = FF^\top + \text{diag}\left(\exp(2 \log \sigma_1, \ldots, 2 \log \sigma_{\dim_z})\right)$.
- **Real NVP** We use a real NVP normalizing flow [4] with two coupling layers and compose the layers in alternate pattern. The flow has a standard multivariate Gaussian as its base distribution. The scale and translation networks have the same architecture of $8 \times 16 \times 16$ hidden units with ReLU activations, followed by a fully connected layer. There is an additional tanh activation at the tail of the scale network to prevent the exponential term from blowing up.

We only present the results for mean-field Gaussian and real NVP in the main section. The results for rank-5 Gaussian are included in Appendix D, as they are largely similar to those obtained for mean-field Gaussian.

B.3 Optimiser and learning rate

We use an Adam optimiser and set its learning rate $\gamma^{(\lambda)} = 0.01$, except for the BNN models with real NVP where we set $\gamma^{(\lambda)} = 0.001$. These learning rates have been selected as the most best options, in terms of convergence time to a respectable ELBO, from the set of $\{0.1, 0.01, 0.001, 0.0001\}$.

B.4 Initialisations

We repeated the experiment five times, each time using different initialisations of λ to assess the convergence performance of each method under varying initial conditions. For the mean-field Gaussian, the λ values were randomly sampled from a zero-mean Gaussian distribution with a scale parameter of 0.5. In contrast, for real NVP, we initialised the λ values using a Glorot normal initialiser [10]. These choices of initialisers were made deliberately to ensure a wide range of initial values, covering both favourable and unfavourable starting points. Consequently, we expect to observe a diverse range of ELBO trajectories.

B.5 Control variates

The gradient estimator is equipped with the following control variate strategies:

- **NoCV** The vanilla gradient estimator without any control variates.
- **ZVCV-GD** A ZVCV with β minimising least squares with an inner gradient descent, as described in Algorithm 2 and Section 4.2. We set the learning rate $\gamma^{(\alpha,\beta)} = 0.001$ and iterated the inner gradient descent 4 times for each outer Adam step. These hyperparameter choices may not always yield the maximum variance reduction in every situation, but they represent a good compromise with computation time. Additionally, we have discovered that prolonging the inner gradient descent iterations does not necessarily lead to better variance reduction. For a more comprehensive discussion, please refer to Appendix G.
- **QuadCV** This is the original algorithm presented in [7] when q is Gaussian (i.e. the mean and covariance of q are readily available). When q is real NVP, we use Algorithm 1 and 100 samples to estimate $\mathbb{E}T(\epsilon;\lambda)$ and $\mathbb{E}\nabla_\lambda \tilde{f}(T(\epsilon;\lambda))$. The learning rate $\gamma^{(v)}$ is set to $\gamma^{(\lambda)}$, following the original work.

Note that above we only compare our method in detail with [7] as it is a direct improvement of [14].

B.6 Evaluation settings

ELBO The ELBO for evaluation purpose is always computed with the full dataset (even when using mini-batched ELBO for optimisation) and 500 samples from q.

Wall-clock time We timed our VI implementation in JAX and ran on an Nvidia A100 80GB GPU. It is worth noting that recorded times may vary among computing platforms and implementations, given that our code was compiled with XLA (resulting in platform-dependent binaries) and ran without memory constraints.

Variance ratio We also present the variance ratio, $\mathbb{V}[\hat{h}]/\mathbb{V}[\hat{g}]$ where $\hat{g}$ and $\hat{h}$ as defined in (3) and (6) respectively, in every 50 iterations; a ratio less than 1 indicates a reduction in variance relative to the corresponding NoCV with the same number of L. The variance of the gradient estimators is computed by repeatedly sampling 100 gradients (say, $\hat{g}_{[1]}, \ldots, \hat{g}_{[100]}$) from the estimator and computed with $\mathbb{V}[\hat{g}] \approx \frac{1}{100}\sum_j \|\hat{g}_{[j]} - (\frac{1}{100}\sum_i \hat{g}_{[i]})\|^2$.

Computation of variance ratio The variance ratio $\mathbb{V}[\hat{h}]/\mathbb{V}[\hat{g}]$ was computed with the following step:

1. Collect 100 samples of $\hat{g}$ resulting in $\{\hat{g}_{[j]}\}_{j=1}^{100}$;
2. For each $\hat{g}_{[j]}$, compute its corresponding control-variate-adjusted gradient estimate $\hat{h}$ (6) to collect $\{\hat{h}_{[j]}\}_{j=1}^{100}$;
3. Estimate $\mathbb{V}[\hat{g}] \approx \frac{1}{100}\sum_j \|\hat{g}_{[j]} - (\frac{1}{100}\sum_i \hat{g}_{[i]})\|^2$. Repeat the same step for $\mathbb{V}[\hat{h}]$;
4. Calculate the ratio $\mathbb{V}[\hat{h}]/\mathbb{V}[\hat{g}]$.

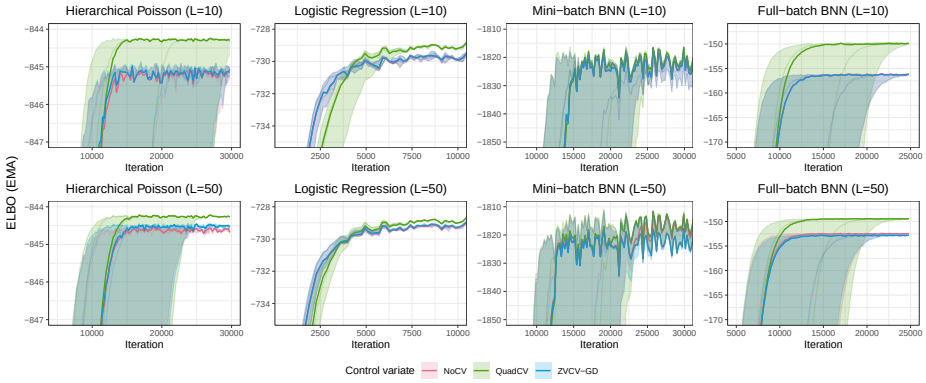
This ratio is designed to evaluate the effectiveness of control variates in reducing variance relative to a corresponding gradient estimator without control variates. Therefore, in our work, the ratio is always computed with a pair of $\hat{g}$ and $\hat{h}$ with the same number of samples.

C Median ELBO against iteration counts

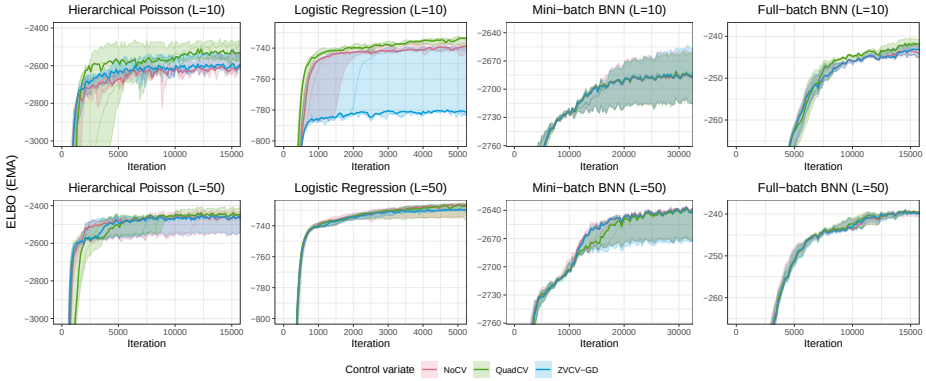
The results in Figure 2 demonstrate that QuadCV generally outperform NoCV, while ZVCV-GD provides only marginal improvement and can even converge to a suboptimal maximum in some cases (e.g. logistic regression, real NVP 2, and $L = 10$). The performance gap between the estimators also decreases as the number of gradient samples L increases, as seen in the bottom rows of Figure 2a and 2b. It should be noted that QuadCVs may perform poorly in the early stages of gradient descent (e.g. logistic regression on mean-field Gaussian and hierarchical Poisson on real NVP) as it takes time to learn the quadratic function $\tilde{f}$. In general, there is also a high degree of variability in ELBO across different runs. This is especially noticeable in Figure 2a due to the substantial impact of λ initialisation on optimisation convergence. For a more detailed examination of the individual trajectories with various initialisations, please refer to Appendix F.

The variance ratio of the gradient estimators can help explain the performance gap observed in Figure 2. As shown in Figure 3, QuadCV generally achieves a lower variance than ZVCV-GD, particularly for Gaussian q when $\mathbb{E}\tilde{f}$ can be computed exactly. The estimator with ZVCV-GD and larger L tends to perform better in models with fewer CV (i.e. low $\dim_z$), as the β is less susceptible to overfitting when solving the least squares with the gradient descent algorithm discussed in Section 4.2. On the contrary, in models with large $\dim_z$, such as BNNs, ZVCV-GD fails to reduce variance.

A noteworthy characteristic of QuadCV is that variance reduction only becomes prominent after $\tilde{f}$ in (11) has been adequately trained. This typically



(a) Mean-field Gaussian with 10 (top) and 50 (bottom) gradient samples.



(b) Real NVP with 10 (top) and 50 (bottom) gradient samples.

Fig. 2: ELBO is plotted against the number of gradient descent steps for different numbers of gradient samples L and two families of q. The bold lines represent the median of ELBO values recorded at the same iteration across five repetitions. The shaded area illustrates the range of ELBO values across five repetitions. The ELBO values are smoothed using an exponential moving average. The trajectories of ZVCV-GD and NoCV are nearly identical in both full-batch and mini-batch BNN when $L = 10$. A higher ELBO indicates better performance. See Figure 6 for plots where the bold lines represent the mean ELBO.

occurs as the optimisation process nears convergence. With a QuadCV-adjusted gradient estimator, it is possible to push the ELBO at convergence a few nats further, although significant time has to be spent to reach convergence at all. However, this raises an interesting question about the worthiness of such an effort, as a relatively minor improvement in ELBO may not necessarily translate into substantially improved downstream metrics; see Appendix F for a more in-depth discussion.

The comparison between $L = 10$ and $L = 50$ in Figure 2 suggests that variance reduction in the early stages can facilitate quicker convergence in terms of iteration counts (notice the leftward shift in the trajectories for $L = 50$. This observation implies that employing a larger number of gradient samples is an effective strategy to improve the convergence performance of stochastic VI, as long as the computation of additional gradient samples remains cost-effective in the overall optimisation process. It is important to note that increasing L from 10 to 50 immediately reduces the gradient estimator's variance by five-fold (equivalent to a variance ratio of 0.2) from the very first iteration of the optimisation, in contrast to QuadCV. These results suggest that variance reduction is more beneficial during the initial stages of optimisation when the goal is to expedite convergence towards a satisfactory ELBO, rather than aiming to attain the maximum achievable ELBO.

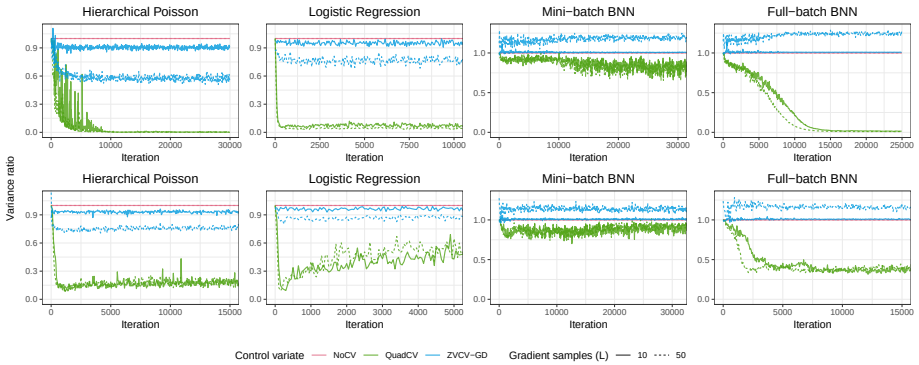


Fig. 3: We present the variance ratio $\mathbb{V}[\hat{h}]/\mathbb{V}[\hat{g}]$, where $\hat{g}$ is NoCV and $\hat{h}$ is either ZVCV-GD or QuadCV, at each iteration. We show only the median variance ratios recorded at the same iteration across five repetitions, omitting the individual variance ratios from each repetition to prevent clutter in the plots. The ratios from mean-field Gaussian and real NVP are shown in top and bottom rows respectively. Note that NoCV (in red) is always 1 by definition. We see that ZVCV-GD (in blue) struggles to reduce variance in the BNN models. There is also a significant overlap in QuadCV between $L = 10$ (solid green) and $L = 50$ (dotted green). A lower ratio indicates better performance. See Figure 7 for plots where the bold lines represent the mean variance ratios.

D Results from rank-5 Gaussian

The insight derived from Figure 4 and 5 below are similar to those obtained from Figure 2, 1 and 3. In most cases, the cost for evaluating control variates outweighs the improvement in ELBO achieved through variance reduction in the gradient estimator. We observe marginal gain in ELBO despite the estimators with control variates taking longer time to converge.

E Mean ELBO trajectories and variance ratio

We have recreated the figures in Section 5 and Appendix D, with the exception that the bold lines now represent the means of ELBO or variance ratios, as opposed to their medians. Using means provides a more transparent depiction of the robustness of each method, although it can be substantially influenced by the repetition that starts farthest from the optimal λ. Ideally, individual trajectories should be plotted separately (as in Appendix F), but this is not feasible due to space limitations. Nonetheless, the findings of this study are substantiated by interpreting either the mean or median of the evaluation statistics.

F Individual runs of full-batch BNN with mean-field Gaussian

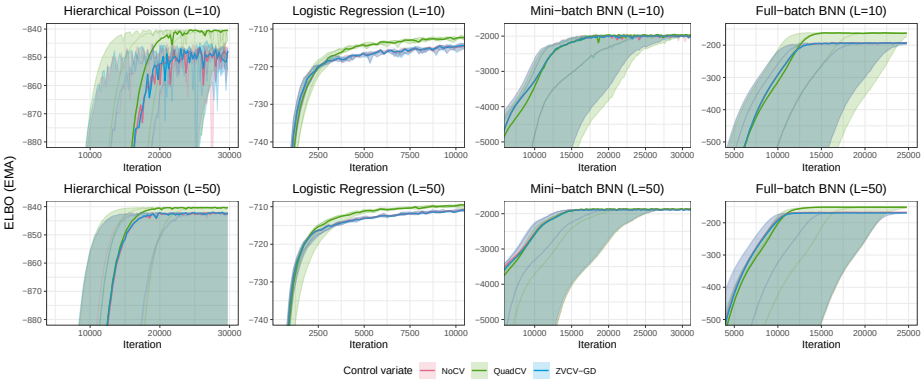
We zoom in on a particular model and variational family from the experiments in the main text. Our aim in this section is to look the trajectory according to each initialisation separately to help visualise the impact of initialisation on convergence. Due to space limitations, we have only included trajectories from full-batch BNN with mean-field Gaussian. In addition to the ELBO reported in the main text, we also report the downstream metric, log pointwise predictive density evaluated on a test set (test lppd), which is popular in the VI literature. Mathematically, the test lppd is defined as

$$\sum_{x \in \mathcal{D}_{\text{test}}} \log \left(|\mathcal{Z}|^{-1} \sum_{z \in \mathcal{Z}} p(x|z) \right).$$

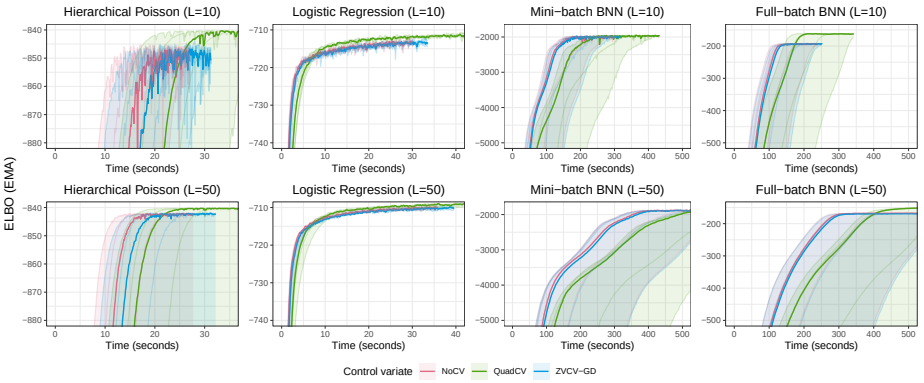
Here, $\mathcal{D}_{\text{test}}$ represents a test set, $\mathcal{Z}$ is a set of samples drawn from $q(z;\lambda)$, and $|\mathcal{Z}|$ indicates the cardinality of $\mathcal{Z}$. We have set $|\mathcal{Z}| = 1000$ in our experiments. The test lppd is also referred to as the test log-likelihood, test log-predictive, or predictive log-likelihood in the literature [23, 3].

Figures 11a clearly show that trajectories vary substantially with different initialisations. This is consistent with the high variability of ELBO trajectories in Figure 2.

In all cases, increasing L, the number of gradient samples, effectively reduces the variance of the gradient estimator from the outset of the optimisation process. This stands in contrast to QuadCV, which only becomes effective after the



(a) ELBO versus iteration counts.



(b) ELBO versus wall-clock time.

Fig. 4: ELBO is plotted against gradient descent steps and wall-clock time for varying numbers of gradient samples L using rank-5 Gaussian. The bold lines represent the median of ELBO values recorded at the same iteration across five repetitions. The ELBO values have been smoothed using an exponential moving average. A higher ELBO indicates better performance. See Figure 9 for plots where the bold lines represent the mean ELBO.

(a) Variance ratio

Fig. 5: We present the variance ratio $\mathbb{V}[\hat{h}]/\mathbb{V}[\hat{g}]$ of rank-5 Gaussian, where $\hat{g}$ is NoCV and $\hat{h}$ is either ZVCV-GD or QuadCV, at each iteration. We show only the median variance ratios recorded at the same iteration across five repetitions, omitting the individual variance ratios from each repetition to prevent clutter in the plots. Note that NoCV (in red) is always 1 by definition. We see that ZVCV-GD (in blue) struggles to reduce variance in the BNN models. There is also some overlap between $L = 10$ (solid green) and $L = 50$ (dotted green). A lower ratio indicates better performance. A lower ratio indicates better performance. See Figure 10 for plots where the bold lines represent the mean variance ratios.

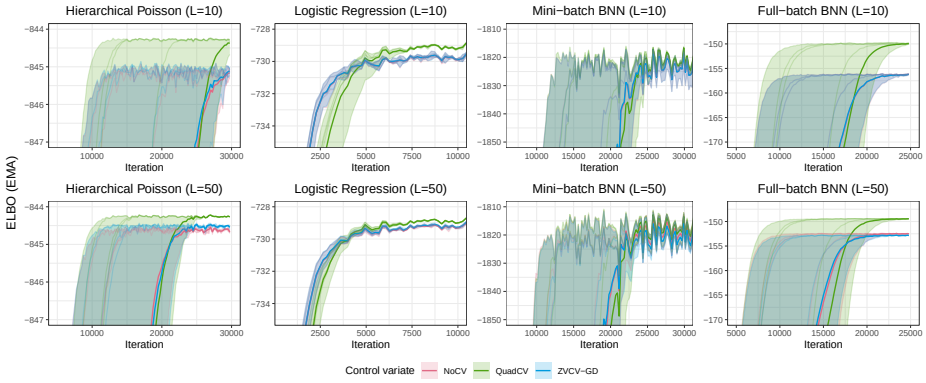
quadratic approximation $\tilde{f}$ in (11) has been adequately trained (Figure 11b). Consequently, QuadCV performs poorly in the early and middle stages of optimisation (as seen in Repetitions 2 and 3 in Figure 11a).

Prior research on variance reduction in pathwise gradient estimators [14, 6, 7] often aims to push the boundaries of attainable ELBO. Achieving this typically requires longer training periods. However, we are of the opinion that the additional ELBO gained through this effort does not warrant the extra computational cost incurred by implementing control variates. This is particularly relevant given that improvements in downstream metrics, such as test lppd, are marginal when compared to improvements achieved in the earlier stages of optimisation (note the y-axis scale in Figure 12a and 12b).

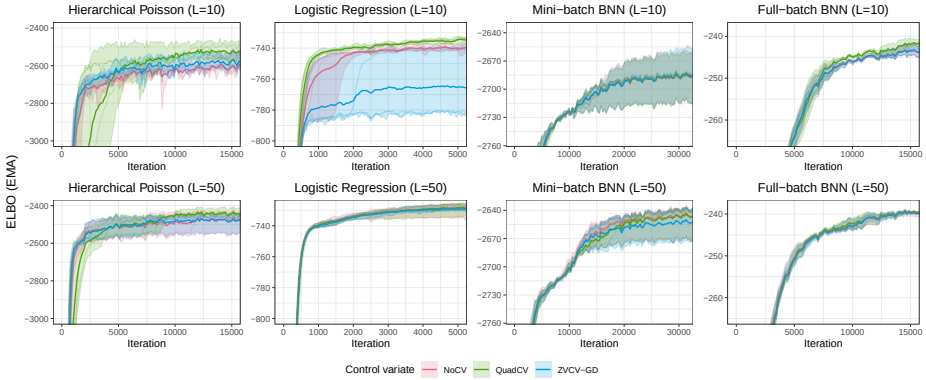
For instance, in Repetition 1, there is only a 3 nats improvement in test lppd (over a test set of size 100), while substantial improvements are observed in the earlier stages, often in the scale of hundreds. These 3 nats come at a cost of over 50% additional computation time compared to NoCV (as indicated in Figure 1a). Furthermore, it is worth noting that an improvement in ELBO does not invariably guarantee a substantial improvement in downstream statistics, as evidenced in previous works, such as [24, 23, 5, 13, 3].

G Comparison of ZVCV-GD with different hyperparameters

We conducted experiments with ZVCV-GD that explore various hyperparameter settings, running with both first- and second-order polynomials (Figure 13), and



(a) Mean-field Gaussian with 10 (top) and 50 (bottom) gradient samples.



(b) Real NVP with 10 (top) and 50 (bottom) gradient samples.

Fig. 6: ELBO is plotted against the number of gradient descent steps for different numbers of gradient samples L and two families of q. The bold lines represent the mean of ELBO values recorded at the same iteration across five repetitions. The shaded area illustrates the range of ELBO values across five repetitions. The ELBO values are smoothed using an exponential moving average. The trajectories of ZVCV-GD and NoCV are nearly identical in both full-batch and mini-batch BNN when $L = 10$. A higher ELBO indicates better performance. See Figure 2 for plots where the bold lines represent the median ELBO.

testing different number of steps in the inner gradient descent loop (Figure 14). We focus on the hierarchical Poisson model using a mean-field Gaussian and setting $L = 10$. We repeated the experiment five times, each time with different initialisations. The red trajectories in Figure 13 and 14 correspond to the default settings of ZVCV-GD as specified in Section 5.

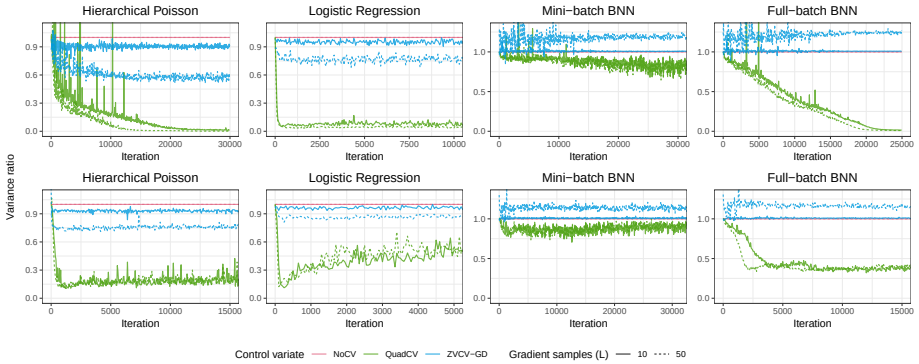
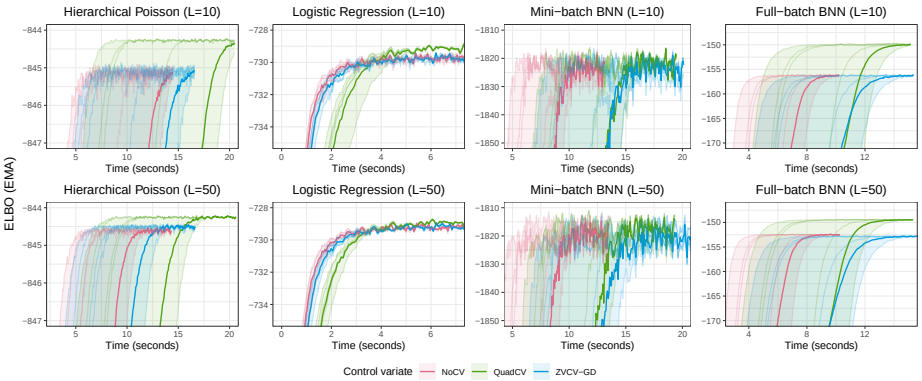
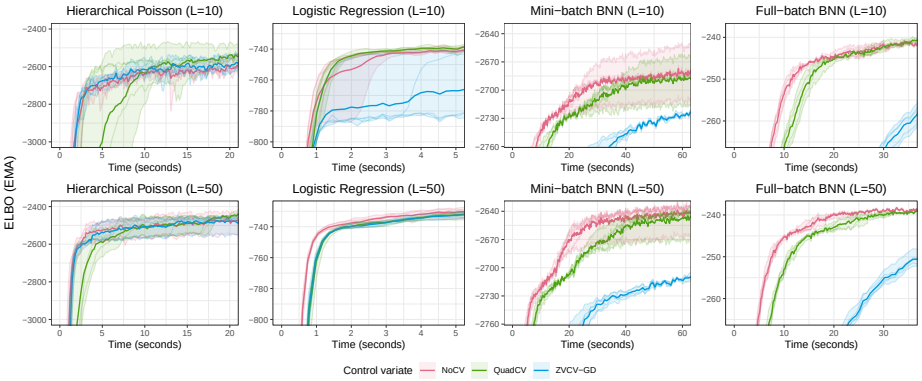


Fig. 7: We present the variance ratio $\mathbb{V}[\hat{h}]/\mathbb{V}[\hat{g}]$, where $\hat{g}$ is NoCV and $\hat{h}$ is either ZVCV-GD or QuadCV, at each iteration. We show only the mean variance ratios recorded at the same iteration across five repetitions, omitting the individual variance ratios from each repetition to prevent clutter in the plots. The ratios from mean-field Gaussian and real NVP are shown in top and bottom rows respectively. Note that NoCV (in red) is always 1 by definition. We see that ZVCV-GD (in blue) struggles to reduce variance in the BNN models. There is also a significant overlap in QuadCV between $L = 10$ (solid green) and $L = 50$ (dotted green). A lower ratio indicates better performance. See Figure 3 for plots where the bold lines represent the median variance ratios.

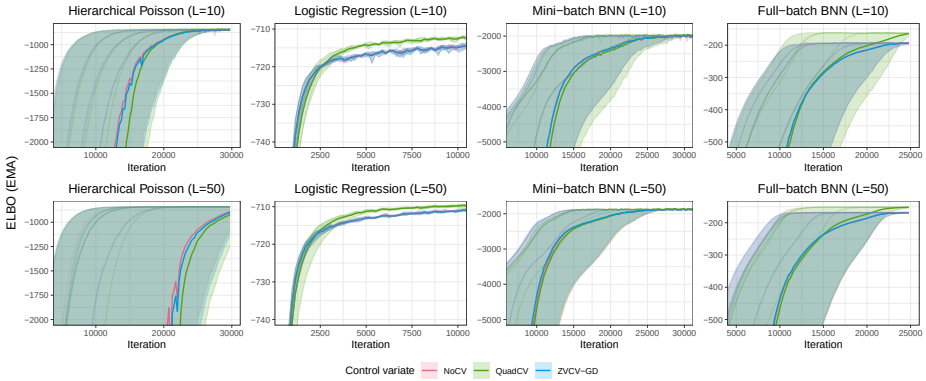


(a) Mean-field Gaussian with 10 (top) and 50 (bottom) gradient samples.

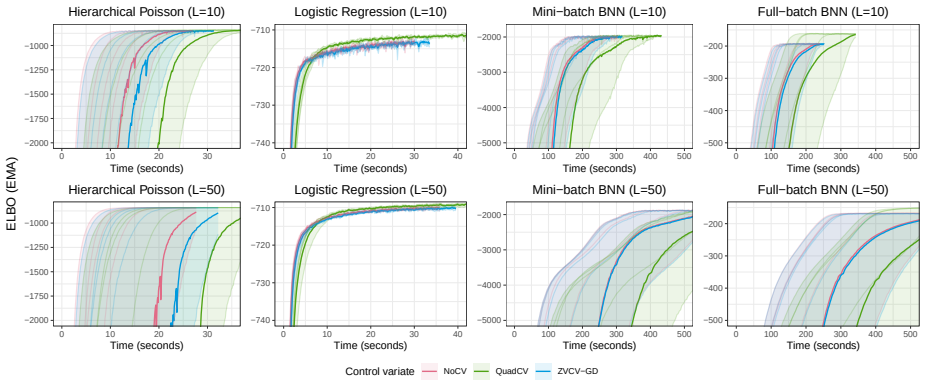


(b) Real NVP with 10 (top) and 50 (bottom) gradient samples.

Fig. 8: ELBO is plotted against wall-clock time for different numbers of gradient samples L and two families of q. The bold lines represent the mean of ELBO values recorded at the same iteration across five repetitions. The shaded area illustrates the range of ELBO values across five repetitions. The ELBO values are smoothed using an exponential moving average. A higher ELBO indicates better performance. See Figure 1 for plots where the bold lines represent the median ELBO.



(a) ELBO versus iteration counts.



(b) ELBO versus wall-clock time.

Fig. 9: ELBO is plotted against gradient descent steps and wall-clock time for varying numbers of gradient samples L using rank-5 Gaussian. The bold lines represent the mean of ELBO values recorded at the same iteration across five repetitions. The ELBO values have been smoothed using an exponential moving average. A higher ELBO indicates better performance. See Figure 4 for plots where the bold lines represent the median ELBO.

(a) Variance ratio

Fig. 10: We present the variance ratio $\mathbb{V}[\hat{h}]/\mathbb{V}[\hat{g}]$ of rank-5 Gaussian, where $\hat{g}$ is NoCV and $\hat{h}$ is either ZVCV-GD or QuadCV, at each iteration. We show only the mean variance ratios recorded at the same iteration across five repetitions, omitting the individual variance ratios from each repetition to prevent clutter in the plots. Note that NoCV (in red) is always 1 by definition. We see that ZVCV-GD (in blue) struggles to reduce variance in the BNN models. There is also some overlap between $L = 10$ (solid green) and $L = 50$ (dotted green). A lower ratio indicates better performance. See Figure 5 for plots where the bold lines represent the median variance ratios.

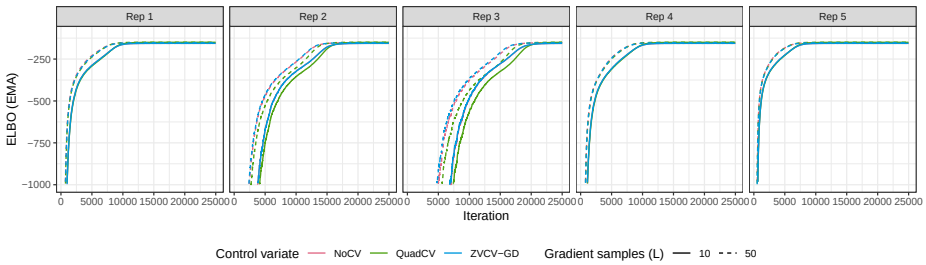
Figure 13b reveals that second-order ZVCV-GD did not effectively reduce variance in the gradient estimator; instead, it introduced additional noise into the estimator. This detrimental impact is also evident in the ELBO trajectories, as shown in Figure 13a. In light of these findings, we concluded that the simpler first-order ZVCV-GD is preferable over the second-order variant.

In Figure 14, we present the ELBO trajectories and variance ratios obtained by running the inner gradient descent (GD) of ZVCV-GD with three different settings: 4 steps, 20 steps, and 'until convergence'. Here, 'convergence' is defined as the point at which the residual of the inner least squares problem in (9) no longer decreases substantially.

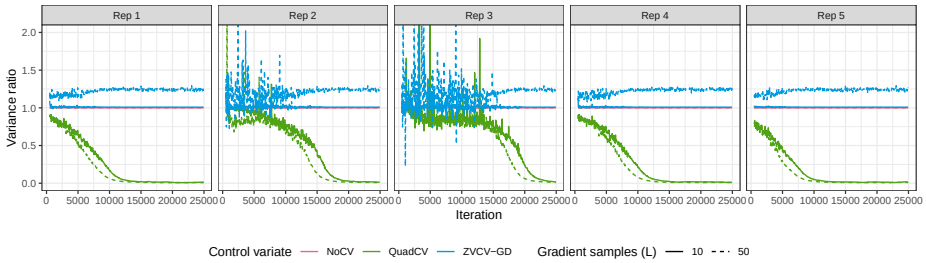
We observe that running the inner GD until convergence does not necessarily yield the greatest variance reduction, as illustrated in Figure 14b. This phenomenon can be attributed to overfitting the linear regression in (9), where the number of rows in C is considerably smaller than the number of columns. On the other hand, iterating the inner GD 20 times achieves a more substantial variance reduction compared to the default 4 steps.

However, it is worth highlighting that there is no discernible impact on the ELBO trajectories when varying the number of GD steps, as demonstrated in Figure 14a.

The optimal number of steps is not always evident without experimentation. Hence, we typically opt for 4 steps to balance computational efficiency and the risk of over-optimizing the inner GD process.

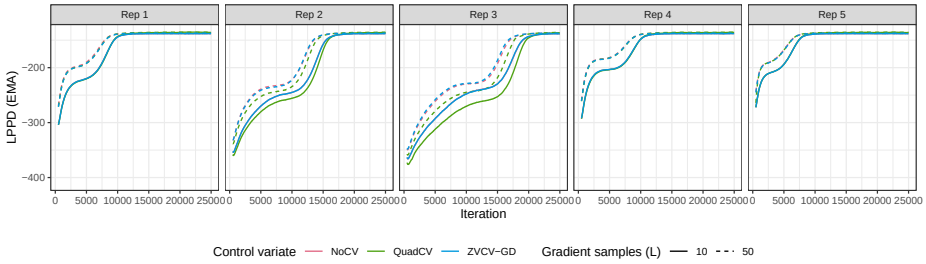


(a) ELBO trajectories. This is a zoomed-out version of the last column of Figure 2a. Higher values are preferred.

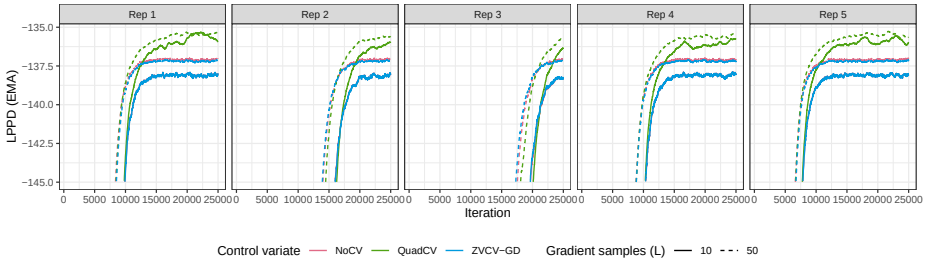


(b) Variance ratios. A reading of 1 indicates no variance reduction. Lower values are preferable.

Fig. 11: The trajectories of ELBO and variance ratio for full-batch BNN with mean-field Gaussian are depicted over the course of iterations, with each of the five repetitions presented individually. By definition, the variance ratio of NoCV (red) is 1. Notably, there is a substantial overlap between NoCV (in red) and ZVCV-GD (in blue). In some cases, the distinctions between all three methods are hardly discernible. However, there is a relatively noticeable difference between $L = 10$ and $L = 50$.

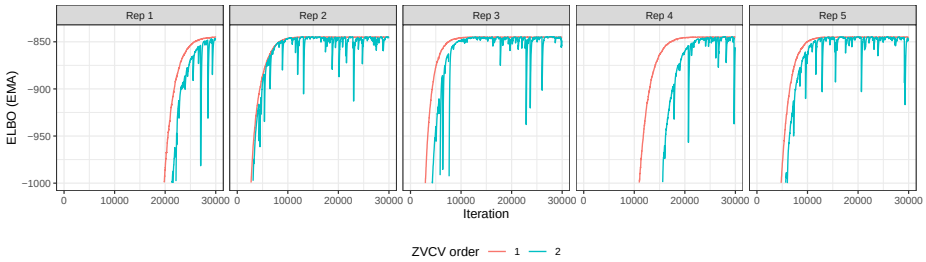


(a) Test lppd trajectories.

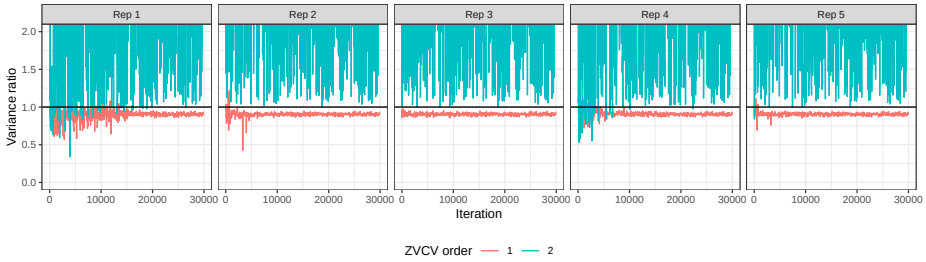


(b) Test lppd trajectories, zooming in between lppd $= (-145, -135)$.

Fig. 12: The trajectories of test lppd for full-batch BNN with mean-field Gaussian are depicted over the course of iterations, with each of the five repetitions presented individually. Notably, there is a substantial overlap between NoCV (in red) and ZVCV-GD (in blue). In some cases, the distinctions between all three methods are hardly discernible. However, there is a relatively noticeable difference between $L = 10$ and $L = 50$. Higher values are preferred.

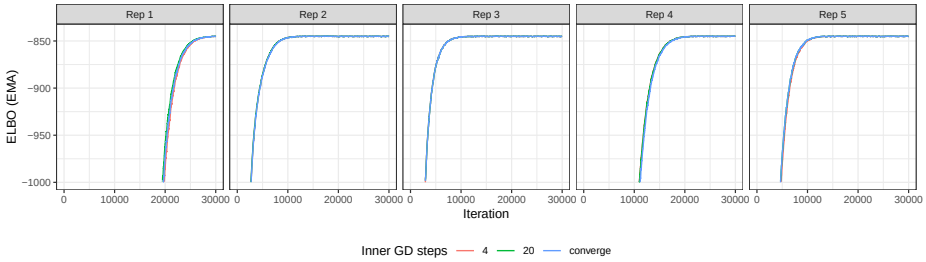


(a) ELBO trajectories. Higher values are preferable.

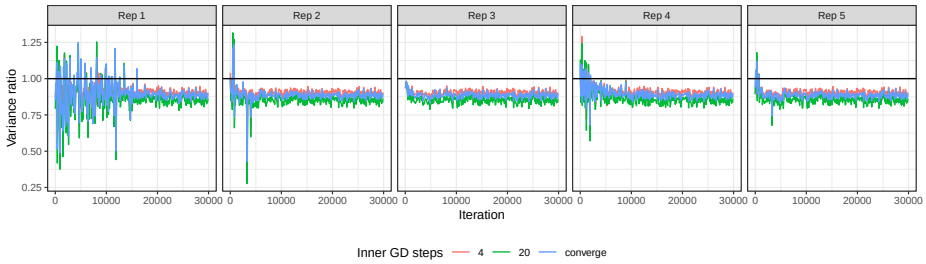


(b) Variance ratios. A reading of 1 indicates no variance reduction. Lower values are preferable.

Fig. 13: ELBO trajectories and variance ratios for hierarchical Poisson models using mean-field Gaussian, ZVCV-GD with $L = 10$, and first- and second-order ZVCV-GD both with 4 inner GD steps. The experiment was repeated five times, each time with different initialisations.



(a) ELBO trajectories. Higher values are preferable.



(b) Variance ratios. A reading of 1 indicates no variance reduction. Lower values are preferable.

Fig. 14: ELBO trajectories and variance ratios for hierarchical Poisson models using mean-field Gaussian, (first-order) ZVCV-GD with $L = 10$, running with different number of steps in the inner gradient descent. The experiment was repeated five times, each time with different initialisations. The ELBO trajectories for different GD steps are practically indistinguishable. The erratic variance ratio readings occur during the early optimisation stages in the low ELBO region, where gradient magnitudes are substantial.