(R1: The distribution learned is not better for downstream tasks. Is ELBO the right metric? Are 3 nats relevant? ...)

* **Why ELBO?** The ELBO is widely accepted and used in the VI community, and a lot of work compares methods in terms of speed and convergence wrt ELBO (Miller et al. NIPS 2017, Buchholz et al. 2018, among others), showing ELBO improvements and speedups similar to ours. Our work sits in this program.

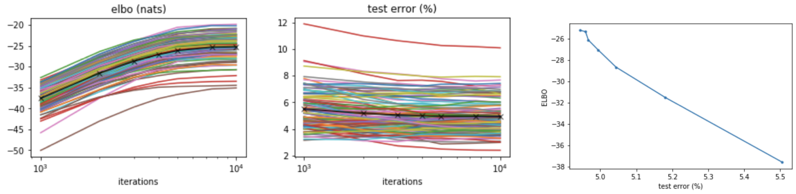
* **Why not test error?** The ELBO is the quantity that VI optimizes, it is more stable than test error (it does not depend on validation data – see results below), and it does not depend on model miss-specification (test error does).

* **Should use normalized ELBO.** We don't see why we should normalize the ELBO wrt the size of the dataset (and no references were provided to support the use of normalized ELBOs). The ELBO measures improvements in KL divergence; a measure between two distributions over the random variable z, not over the data.

* **Are 3 nats significant?** There are cases where a divergence of 3 nats is certainly significant. Take some posterior $p(z|x)$, and choose a set A such that $p(z \in A|x) = 0.05$. Choose $q(z)$ to be $q(z) = 20p(z|x)$ for $z \in A$ and $q(z) = 0$ if $z \notin A$. In this case we get $\mathrm{KL}(q(z)||p(z|x)) = \log 20 \approx 3$, despite the distributions being quite different.

* **With a dataset of size 100 and a difference of 3-4 nats it is extremely likely that if we were to evaluate on downstream metrics [...] differences between the learned approximate posteriors would be essentially non-existent.** We think this claim is unfounded, and no references were provided. We ran simulations to support our position. We consider a 10 dimensional logistic regression model with a dataset of size 100. We obtain approximating distributions that achieve different ELBO values by running VI for different numbers of iterations. For each approximating distribution we use a test set to measure classification test error. We repeat this 100 times. Results are shown next.

We see that a difference of 3 nats leads to improvements of around 0.2% in test error. This contradicts the claim that that the a 3 nats improvement leads to non-existent improvements in performance on downstream metrics.



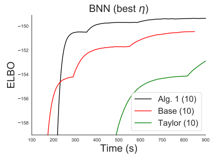
(R1, R2: Convergence wrt wall clock time.) The paper shows the time cost of each method in Table 1. While results wrt to wall clock time may be obtained from the table, we'd be happy to add them to the paper. For instance, the two leftmost figures below show some examples (two models, diagonal plus low rank q, best step-size chosen retrospectively). Our method achieves speedups of at least 3×. Results for other cases are similar, with speedups ranging from 3× to 7×.

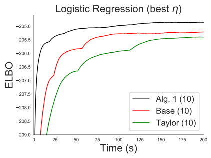
(R2, R4: Comparison to other methods.) We focused on the development of a control variate (CV) that addressed the issues of the one proposed by Miller et al. (2017). Our CV may be used jointly with other variance reduction methods, and thus should not be seen as a replacement but as an addition to them. For instance, the sticking-the-landing (STL) estimator (Roeder et al. 2017) can be used with our CV in two ways: a) setting the base gradient estimator to the STL; b) creating the "STL control variate" and using in concert with our CV (Geffner and Domke, 2018). Our CV can also be used with Randomized QMC sampling (Buchholz et al 2018). Finally, while we focus on reparameterization, our CV could be used with other estimators as well, such as the score function or generalized reparameterization (Ruiz et al. 2016), as long as the covariance of the variational distribution is known. This is done by obtaining the second term from eq. 5 using the corresponding estimator (instead of reparameterization). We'll add a discussion about this in the paper.

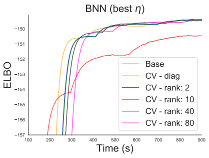
(R3: Low novelty.) We respectfully disagree. Miller et al. used a quadratic function for the mean parameters, but a linear function for the scale (problematic with non-factorized distributions). We provide an extensive new analysis of this. Also, double-descent was previously used, but for the score function estimator with importance sampling (Ruiz et al. 2016) or with relaxations for discrete variables (Tucker et al. 2016, Grathwohl et al. 2018). We use these ideas jointly in a different way to address the main issue with the CV by Miller et al. (uncovered with our new analysis).

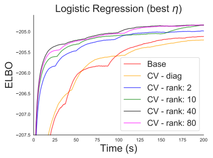
(R3: Rank use for the control variate.) We chose ranks 10 and 20 to show that even with ranks considerably smaller than the dimension of the problem the control variate lead to improved results. Results for more ranks could be insightful, and we'll happily add them to the paper. For instance, the two rightmost figures below show results for different ranks.

(R3: Description of method by Miller et al.) We gave a lot of thought to how to present this. We resolved the ambiguities in their paper by looking at the code, which reflects the exact method used for their experiments. (We ran simulations with the exact same models and random seeds to verify that the method we present and the one used in their code are equivalent.) We believe they mention the method by Bekas because they use matrix-vector products to estimate the trace of a matrix (the scaled Hessian). However, with the analysis we present, we show that the way they combine this with baselines leads to the undesirable cancellations described in our paper.


BNN (best η)
ELBO
Time (s)
Alg. 1 (10)
Base (10)
Taylor (10)


Logistic Regression (best η)
ELBO
Time (s)
Alg. 1 (10)
Base (10)
Taylor (10)


BNN (best η)
ELBO
Time (s)
Base
CV - diag
CV - rank: 2
CV - rank: 10
CV - rank: 40
CV - rank: 80


Logistic Regression (best η)
ELBO
Time (s)
Base
CV - diag
CV - rank: 2
CV - rank: 10
CV - rank: 40
CV - rank: 80