entropy

MDPI

Article

Sampling the Variational Posterior with Local Refinement

Marton Havasi [1,*,†], Jasper Snoek [2], Dustin Tran [2], Jonathan Gordon [1] and José Miguel Hernández-Lobato [1]

[1] Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK; jongordon06@gmail.com (J.G.); jmh233@cam.ac.uk (J.M.H.-L.)
[2] Brain Team, Google Research, Mountain View, CA 94043, USA; jsnoek@google.com (J.S.); trandustin@google.com (D.T.)
* Correspondence: mh740@cam.ac.uk
† Work partially completed as a Google intern.

Abstract: Variational inference is an optimization-based method for approximating the posterior distribution of the parameters in Bayesian probabilistic models. A key challenge of variational inference is to approximate the posterior with a distribution that is computationally tractable yet sufficiently expressive. We propose a novel method for generating samples from a highly flexible variational approximation. The method starts with a coarse initial approximation and generates samples by refining it in selected, local regions. This allows the samples to capture dependencies and multi-modality in the posterior, even when these are absent from the initial approximation. We demonstrate theoretically that our method always improves the quality of the approximation (as measured by the evidence lower bound). In experiments, our method consistently outperforms recent variational inference methods in terms of log-likelihood and ELBO across three example tasks: the Eight-Schools example (an inference task in a hierarchical model), training a ResNet-20 (Bayesian inference in a large neural network), and the Mushroom task (posterior sampling in a contextual bandit problem).

Keywords: bayesian inference; variational inference; deep neural networks; contextual bandits

Citation: Havasi, M.; Snoek, J.; Tran, D.; Gordon, J.; Hernández-Lobato, J.M. Sampling the Variational Posterior with Local Refinement. *Entropy* **2021**, *23*, 1475. https://doi.org/10.3390/e23111475

Academic Editor: Wray Buntine

Received: 30 September 2021
Accepted: 3 November 2021
Published: 8 November 2021

1. Introduction

Uncertainty plays a crucial role in a multitude of machine learning applications, ranging from weather prediction to drug discovery. Poor predictive uncertainty risks potentially poor outcomes, especially in domains such as medical diagnosis or autonomous vehicles, where high confidence errors may be especially costly [1]. Thus, it is tremendously important that the underlying model provides high quality uncertainty estimates along with its predictions. By marginalizing over a posterior distribution over the parameters given the training data, Bayesian inference provides a principled approach to capturing uncertainty. Unfortunately, exact Bayesian inference is not generally tractable. Variational inference (VI) instead approximates the true posterior with a simpler distribution. VI is appealing since it reduces the problem of inference to an optimization problem, where the goal is to minimize the discrepancy between the true posterior and the variational posterior. The key challenge, however, is the task of training expressive posterior approximations that can capture the true posterior without significantly increasing computational and memory costs. The most widely used one is the mean-field approximation, where the posterior is represented using an independent Gaussian distribution over all the model parameters. The mean-field approximation is easy to train, but it fails to capture dependencies and multi-modality in the true posterior.

This paper describes a novel method for generating samples from a highly flexible posterior approximation. The idea is to start with a coarse, mean-field approximation and make a series of inexpensive, local refinements to it. At the end, we draw a sample from the refined region. We show that through this process, we can generate samples that capture both *dependencies* and *multi-modality* in the true posterior.

The refinements take place at gradually decreasing scales starting with large scale changes, moving towards small scale adjustments. The regions of these adjustments are determined by sampling the values of additive auxiliary variables. Formally, we express the model parameters $\mathbf{w}$ using a number of additive auxiliary variables $\mathbf{w} = \mathbf{a}_1 + \ldots + \mathbf{a}_K$ (Figure 1 left) that leave the marginal distribution unchanged. The refinement process takes place over K optimization steps. In each step, we sample the value of an auxiliary variable according to the current variational approximation $\mathbf{a}_k \sim q(\mathbf{a}_k)$ and optimize the approximation by conditioning on the newly sampled value $q(\mathbf{w}) \approx p(\mathbf{w}|\boldsymbol{x}, \boldsymbol{y}, \mathbf{a}_{1:k})$ $(k = 1 \ldots K)$. At the end, we obtain a sample $\mathbf{w} = \mathbf{a}_1 + \ldots + \mathbf{a}_K$ from the refined posterior $q_{\text{ref}}(\mathbf{w})$. To obtain further samples, we must go back to our initial, coarse approximation and repeat the K-step process again. We refer to the refinements as local, because after sampling each auxiliary variable, the process moves towards smaller scale adjustments until it reaches $\mathbf{w}$.

The refined posterior is a highly flexible approximation to the true posterior. It is able to capture dependencies and multi-modality even when these are absent from the initial variational approximation. We demonstrate the multi-modality of the refined posterior on a synthetic example, and we show how the refined posterior is able to capture dependencies in a hierarchical inference problem.

We theoretically show that the refined posterior improves the ELBO over the initial variational approximation. We also demonstrate this empirically by applying the method to Bayesian neural networks on common regression and image classification benchmarks.

Generating each sample requires a series of optimization steps that come with associated computational costs. We found that in a deep neural network, the computational overhead of generating a small set of samples for prediction amounts to ∼30% of the cost of training the initial variational approximation; thus, the refinement process is able to generate a set of high-quality posterior samples at the cost of a small computational overhead (compared to training a standard mean-field approximation).

An ideal application of our method is using it to generate posterior samples for Thompson sampling, which is a popular approach to tackle contextual bandit tasks. It works by sampling a random hypothesis from the posterior to decide on each action. In this scenario, the computational cost is not a key consideration, we can spend further computation on generating high quality posterior samples. We show that the high quality samples generated by refining the posterior improve the performance of Thompson sampling in contextual bandit task as measured by the cumulative regret.

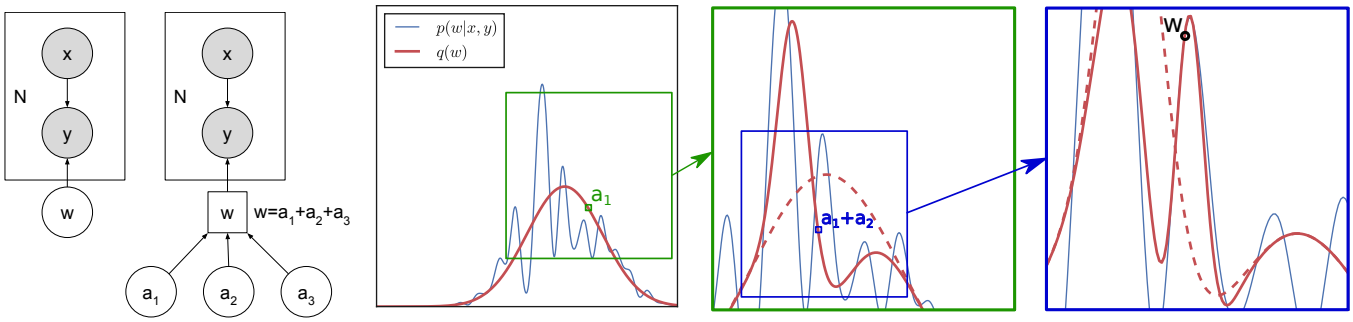

Figure 1. (**Left**) The supervised learning model and augmented model, respectively, where w is expressed as a sum of independent auxiliary variables. (**Right**) High level illustration of the refining algorithm. In each iteration, the value of an auxiliary variable is fixed, and the posterior is locally adjusted. In the final iteration, a sample is drawn from $q(w)$. Through the iterations, the variational distribution is able to approximate well the true posterior in a local region.

Organization of the Paper

In Section 2, we start by introducing the notation and giving an overview of variational inference. Then, we present our proposed algorithm for generating samples from a

refined variational distribution. Through two examples, we show that refined posterior can capture both dependencies and multi-modality. In Section 3, we provide theoretical guarantees that the refinement step always improves the quality of the variational distribution (measured by the ELBO) under mild conditions. In Section 4, we evaluate the effectiveness of the method on Bayesian neural networks on a set of UCI regression and image classification benchmarks. We observe that our method consistently improves the quality of the approximation, as evidenced by a higher ELBO and likelihood of the samples. We also demonstrate that the high-quality posterior samples can be used in Thompson sampling to reduce the cumulative regret in a contextual bandit task. In Section 5, we discuss a related works and place our method in context.

2. Materials and Methods

In this section, we first describe standard variational inference (VI), followed by a detailed description of our proposed sample generation method that refines the variational posterior. The inputs and labels are denoted by $\boldsymbol{x} \subseteq \mathcal{X}$ and $\boldsymbol{y} \subseteq \mathcal{Y}$, respectively, and $\mathbf{w}$ denotes the model parameters.

2.1. Variational Inference

Exact Bayesian inference is often intractable and is NP-hard in the worst case. Variational inference attempts to approximate the true posterior $p(\mathbf{w}|\boldsymbol{x}, \boldsymbol{y})$ with an approximate posterior $q_\phi(\mathbf{w})$, typically from a simple family of distributions, for example independent Gaussians over the weights, i.e., the mean-field approximation. To ensure that the approximate posterior is close to the true posterior, the parameters of $q_\phi(\mathbf{w})$, ϕ are optimized to minimize their Kullback–Leibler divergence: $\mathrm{KL}\big[\, q_\phi(\mathbf{w}) \;||\; p(\mathbf{w}|\boldsymbol{x}, \boldsymbol{y}) \,\big]$. At the limit of $\mathrm{KL}\big[\, q_\phi(\mathbf{w}) \;||\; p(\mathbf{w}|\boldsymbol{x}, \boldsymbol{y}) \,\big] = 0$, the approximate posterior exactly captures the true posterior, although this might not be achievable if $p(\mathbf{w}|\boldsymbol{x}, \boldsymbol{y})$ is outside of the distribution family of $q_\phi(\mathbf{w})$.

In order to minimize the KL-divergence, variational inference optimizes the evidence lower bound (ELBO) w.r.t. ϕ (denoted as $\mathcal{L}(\phi)$), which is a lower bound to the log marginal likelihood $\log p(\boldsymbol{y}|\boldsymbol{x})$. Since the marginal log-likelihood can be expressed as the sum of the KL-divergence and the ELBO, maximizing the ELBO is equivalent to minimizing the KL divergence:

$$\begin{aligned} \log p(\boldsymbol{y}|\boldsymbol{x}) &= \underbrace{\mathrm{KL}\big[\, q_\phi(\mathbf{w}) \;||\; p(\mathbf{w}|\boldsymbol{x}, \boldsymbol{y}) \,\big]}_{\geq 0} + \mathcal{L}(\phi) \\ &\geq \mathcal{L}(\phi) \\ &= \mathbb{E}_{q_\phi}\big[\, \log p(\boldsymbol{y}|\boldsymbol{x}, \mathbf{w}) \big] - \mathrm{KL}\big[\, q_\phi(\mathbf{w}) \;||\; p(\mathbf{w}) \,\big] \end{aligned} \tag{1}$$

due to non-negativity of the KL-divergence.

Following the optimization of ϕ, the model can be used to make predictions on unseen data. For an input $\boldsymbol{x}'$, the predictive distribution $p(\mathbf{y}'|\boldsymbol{x}', \mathbf{y}, \mathbf{x})$ can be approximated by stochastically drawing a small number of sample model parameters $\mathbf{w}_{1:M} \sim q_\phi(\mathbf{w})$ and averaging their prediction in an ensemble model $p(\mathbf{y}'|\boldsymbol{x}', \mathbf{y}, \mathbf{x}) \approx \frac{1}{M} \sum_{i=1}^{M} p(\mathbf{y}'|\boldsymbol{x}', \mathbf{w}_i)$.

2.2. Refining the Variational Posterior

The main issue with variational inference is the inflexibility of the posterior approximation. The most widely used variant of variational inference, mean-field variational inference, approximates the posterior with independent Gaussians across all dimensions. This approximation is too simplistic to capture the complexities of the posterior for complicated models. With our proposed method, it is feasible to generate samples from a detailed posterior by starting with a mean-field approximation and refining it in selected, local regions. Note that the method does not yield an analytic form to the detailed posterior, it generates a set of samples $\mathbf{w}_{1:M}$ from it.

The graphical model is augmented with a finite number of auxiliary variables $\mathbf{a}_{1:K}$ as shown in Figure 1. The constraints are that $(\mathbf{x}, \mathbf{y})$ must be conditionally independent of the auxiliary variables given $\mathbf{w}$ and that they must not affect the prior distribution $p(\mathbf{w})$. These constraints ensure that the marginal likelihood $\log p(\mathbf{y}|\boldsymbol{x})$ is unchanged, enabling us to train the augmented model with the same ELBO as the unaugmented model; thus, *the model is unaffected by the presence of the auxiliary variables*. Their purpose is solely to aid the inference procedure. Given a Gaussian prior $\mathcal{N}(\mathbf{w}|\mathbf{0}, \sigma_w^2 I)$ over $\mathbf{w}$, we express $\mathbf{w}$ as a sum of independent auxiliary variables (Although we are focusing on one specific definition of the auxiliary variables, additive auxiliary variables, note that all of our results straightforwardly generalize to arbitrary joint distributions $p(\mathbf{w}, \mathbf{a}_{1:K})$ that meet the constraints).

$$\mathbf{w} = \sum_{k=1}^{K} \mathbf{a}_k, \text{ with } p(\mathbf{a}_k) = \mathcal{N}(\mathbf{a}_k|\mathbf{0}, \sigma_{\mathbf{a}_k}^2 I) \quad \text{for } k = 1 \dots K,$$

while ensuring that $\sum_{k=1}^{K} \sigma_{a_k}^2 = \sigma_w^2$, so that the prior $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \sigma_w^2 I)$ remains unchanged.

We refine the approximate posterior to generate each sample $\mathbf{w}_{1:M}$. Specifically, this refers to iteratively sampling the values of the auxiliary variables $\mathbf{a}_{1:K}$ and then approximating the posterior of $\mathbf{w}$, conditional on the sampled values, i.e., $q_{\phi_k}(\mathbf{w})$ approximates $p(\mathbf{w}|\boldsymbol{x}, \boldsymbol{y}, \mathbf{a}_{1:k})$ for iterations $k = 1 \dots K$ (ϕ_k is dependent on $\mathbf{a}_{1:k}$) as shown in Algorithm 1.

Algorithm 1: Refine and Sample (ϕ_0)

Data: q_{ϕ_0}
Result: $\mathbf{w}_{1:M}$
for $m = 1, \dots, M$ **do**
  for $k = 1, \dots, K$ **do**
    Sample $\mathbf{a}_k \sim q_{\phi_{k-1}}(\mathbf{a}_k)$;
    Initialize $q_{\phi_k}(\mathbf{w}) \leftarrow q_{\phi_{k-1}}(\mathbf{w}|\mathbf{a}_k)$;
    Optimize $\phi_k \leftarrow \arg\max_{\phi_k} \mathcal{L}_{|\mathbf{a}_{1:k}}(\phi_k)$;
  end
  Sample $\mathbf{w}_m \sim q_{\phi_K}(\mathbf{w})$;
end
return $\mathbf{w}_{1:M}$;

That is, starting from the initial mean-field approximation $q_{\phi_0}(\mathbf{w})$, for $k = 1, \dots, K$,

1. Sample the value of $\mathbf{a}_k$ using the current variational approximation and fix its value.

$$\mathbf{a}_k \sim q_{\phi_{k-1}}(\mathbf{a}_k) = \int p(\mathbf{a}_k|\mathbf{a}_{1:k-1}, \mathbf{w}) q_{\phi_{k-1}}(\mathbf{w}) \, \mathrm{d}\mathbf{w} \tag{2}$$

   A sample can be obtained by first sampling $\mathbf{w} \sim q_{\phi_{k-1}}(\mathbf{w})$ followed by $\mathbf{a}_k \sim p(\mathbf{a}_k|\mathbf{a}_{1:k-1}, \mathbf{w})$. This is straightforward for exponential families and factorized distributions. The closed form for $q_{\phi_{k-1}}(\mathbf{a}_k)$ is provided in the Appendix A.

2. Optimize the variational approximation conditional on the sampled $\mathbf{a}_k$: $q_{\phi_k}(\mathbf{w}) \approx p(\mathbf{w}|\boldsymbol{x}, \boldsymbol{y}, \mathbf{a}_{1:k})$.

$$\phi_k \leftarrow \arg\min \mathrm{KL}\left[q_{\phi_k}(\mathbf{w}) \;||\; p(\mathbf{w}|\boldsymbol{x}, \boldsymbol{y}, \mathbf{a}_{1:k}) \right] \tag{3}$$

   This optimization is very fast in practice if ϕ_k is initialized using the solution from the previous iteration: $q_{\phi_k}(\mathbf{w}) \overset{\text{init}}{\leftarrow} q_{\phi_{k-1}}(\mathbf{w}|\mathbf{a}_k)$. The closed form of $q_{\phi_{k-1}}(\mathbf{w}|\mathbf{a}_k)$ provided in the Appendix A.

We then obtain $\mathbf{w} = \sum_{k=1}^{K} \mathbf{a}_k$. Analogous to VI, the KL-divergence in step 2 is minimized by maximizing the conditional ELBO

$$\mathcal{L}_{|\mathbf{a}_{1:k}}(\phi_k) = \mathbb{E}_{q_{\phi_k}}\left[\log p(\boldsymbol{y}|\boldsymbol{x}, \mathbf{w})\right] - \mathrm{KL}\left[\, q_{\phi_k}(\mathbf{w}) \;||\; p(\mathbf{w}|\mathbf{a}_{1:k}) \,\right], \tag{4}$$

where $p(\mathbf{w}|\mathbf{a}_{1:k}) = \mathcal{N}(\mathbf{w}|\sum_{i=1}^{k} \mathbf{a}_i, I(\sigma_{\mathbf{w}}^2 - \sum_{i=1}^{k} \sigma_i^2))$. Note that, when $k = K$, the numerical minimization of $\mathrm{KL}\left[\, q_{\phi_k}(\mathbf{w}) \;||\; p(\mathbf{w}|\boldsymbol{x}, \boldsymbol{y}, \mathbf{a}_{1:k}) \,\right]$ is unnecessary since in this case, the optimal $q_{\phi_K}(\mathbf{w})$ is a delta function located at the sum of the sampled $\mathbf{a}_{1:K}$.

In order to generate M independent samples $\mathbf{w}_{1:M}$ from the refined posterior, the previous process has to be repeated M times, sampling new values for $\mathbf{a}_{1:K}$ each time.

2.3. Multi-Modal Toy Example

We use a synthetic toy example to demonstrate the procedure and to show that through the refinement steps, the approach is able to capture multiple posterior modes. In this example, we have a single weight $\mathbf{w}$ with prior $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|0, 1)$ and a complex posterior with four modes. Figure 2b shows that a Gaussian approximation fails to capture the multi-modal nature of the true posterior.

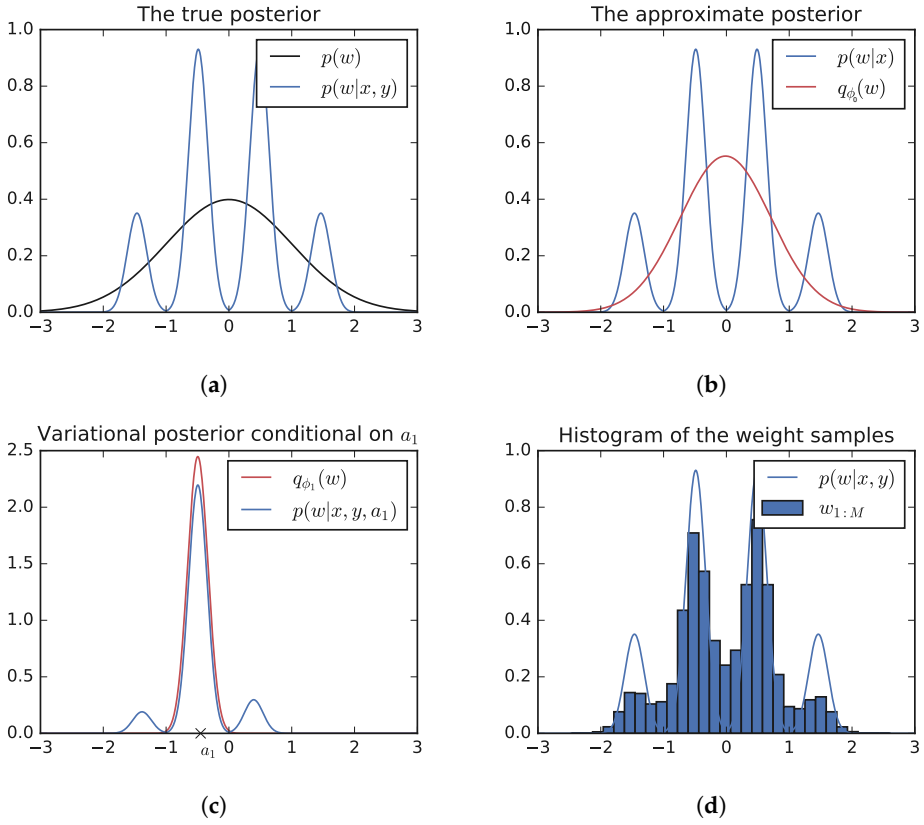

Figure 2. Our method can capture a multi-modal posterior starting with a Gaussian posterior approximation. (**a**) The true posterior, which is too complex to be well approximated by a Gaussian distribution. (**b**) The Gaussian approximate posterior after optimizing the ELBO (ELBO $= -1.79$). (**c**) We sample $\mathbf{a}_1$, optimize the resulting conditional ELBO to obtain $q_{\phi_1}(\mathbf{w})$ and then sample $\mathbf{w}_m \sim q_{\phi_1}(\mathbf{w})$. This whole process repeats $m = 1, \ldots, M$ times to obtain $\mathbf{w}_{1:M}$. (**d**) Histogram of the samples $\mathbf{w}_{1:M}$ obtained from the refined posterior approximation. ELBO ≥ -1.45.

We express $\mathbf{w}$ as the sum of $K = 2$ auxiliary variables: $\mathbf{w} = \mathbf{a}_1 + \mathbf{a}_2$ with $p(\mathbf{a}_1) = \mathcal{N}(\mathbf{a}_1|0, 0.8)$ and $p(\mathbf{a}_2) = \mathcal{N}(\mathbf{a}_2|0, 0.2)$, which recovers $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|0, 1)$ as per the constraint. The first step of the refinement process is sampling $\mathbf{a}_1 \sim q_{\phi_0}(\mathbf{a}_1) = \int p(\mathbf{a}_1|\mathbf{w}) q_{\phi_0}(\mathbf{w})\, \mathrm{d}\mathbf{w}$, where $q_{\phi_0}(\mathbf{w})$ is an initial mean field approximation to the poste-

rior. Then, the variational posterior is optimized conditional on the sampled $\mathbf{a}_1$; that is, $\phi_1 = \arg\min \mathrm{KL}\left[\, q_{\phi_1}(\mathbf{w}) \;||\; p(\mathbf{w}|\boldsymbol{x}, \boldsymbol{y}, \mathbf{a}_1) \,\right]$. Figure 2c shows that the conditional variational posterior is able to fit one of the posterior modes. Over many runs, the different values of $\mathbf{a}_1$ force the conditional posterior to fit different posterior modes, thus allowing the refined posterior to capture the multi-modal nature of the true posterior as shown in Figure 2d. Clearly, the refined posterior is a much better approximation to the true posterior than the Gaussian approximation though we note that the true posterior is not recovered exactly.

2.4. Capturing Dependencies: A Hierarchical Example

In this section, we use the eight-schools example from STAN [2,3] to show how the refined posterior can capture dependencies among the hidden variables and to discuss the effect of the number of auxiliary variables on the quality of the posterior approximation.

The eight-schools example studies the coaching effect of 8 schools. Each school reports the mean y_i and standard error σ_i of its coaching effect where $i = 1, \ldots, 8$. There is no prior reason to believe that any school was more effective than another so the model is stated in a hierarchical manner:

$$\mu \sim \mathcal{N}(0, 25), \quad \tau \sim \mathit{HalfCauchy}(0, 5), \quad \theta_i \sim \mathcal{N}(\mu, \tau^2), \quad y_i \sim \mathcal{N}(\theta_i, \sigma_i^2) \text{ for } i = 1 \ldots 8,$$

where the HalfCauchy distribution refers to a Cauchy distribution supported only on positive values (i.e., a symmetric half of the Cauchy distribution).

Factorized approximations perform poorly on this problem due to the dependency of θ on τ (for an excellent analysis of this problem, see [4]). In fact, the MAP solution is at $\tau = 0$, which is distant from the mean-field approximation that STAN uses for variational inference (ADVI, [5]) (Figure 3 left).

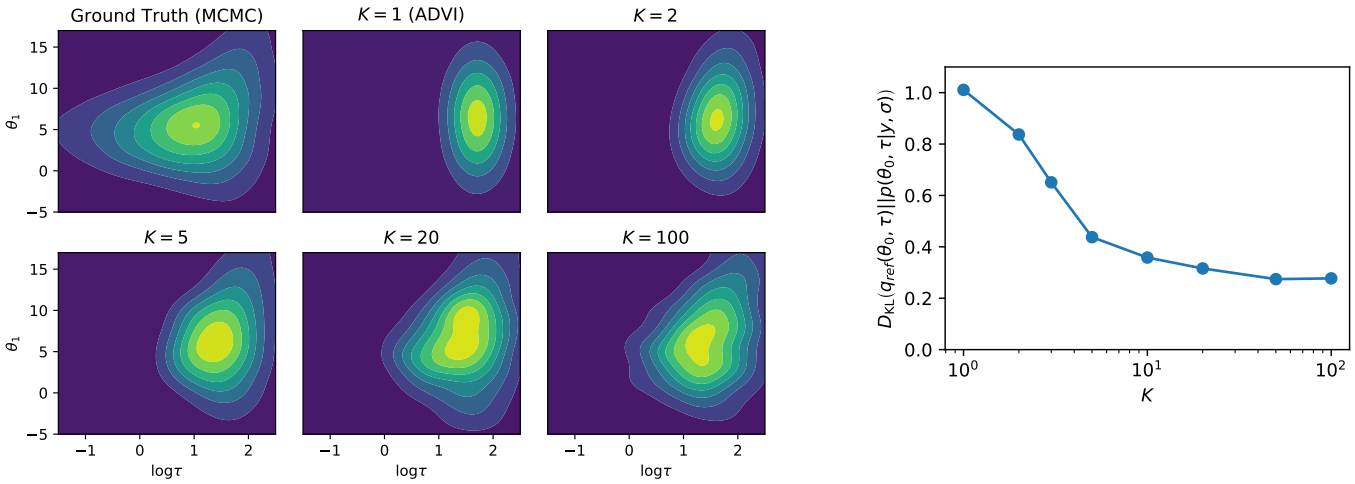

Figure 3. (**Left**) The refined posterior for increasing numbers of auxiliary variables. As K increases, the refined posterior is able to capture the dependency between θ_1 and τ. (**Right**) The KL divergence between the refined posterior and approximate posterior decreases as K grows. (Calculated using kernel density estimation.)

We show that our method can capture the dependencies between θ and τ. We introduce the following additive auxiliary variables:

$$\mu = \sum_{k=1}^{K} a_{\mu_k} \quad a_{\mu_k} \sim \mathcal{N}\left(0, \frac{25}{K}\right), \quad \tau = \left| \sum_{k=1}^{K} a_{\tau_k} \right|$$

$$a_{\tau_k} \sim \mathit{Cauchy}\left(0, \frac{5}{K}\right), \quad \theta = \mu + \tau \sum_{k=1}^{K} a_{\theta_k} \quad a_{\theta_k} \sim \mathcal{N}\left(0, \frac{1}{K}\right),$$

for $k = 1 \ldots K$. As required by the constraints, the auxiliary variables leave the model unchanged.

Figure 3 left shows the approximate posterior for various K values. At $K = 1$, the model is equivalent to ADVI, and as K increases, we can see that the refined posterior is able to capture the dependencies between τ and θ_1 and results in a non-Gaussian form. The ground truth samples were obtained using the NUTS sampler in PyMC3 [6,7]. The density plots were generated using kernel-density-estimation.

2.5. Limit as $K \to \infty$

A natural question to ask is what happens as the number of auxiliary variables grows to infinity. We can estimate the KL-divergence of the refined posterior and the true posterior in the eight-schools example using kernel density estimation based on the samples generated from the refined posterior. We see that it monotonically decreases (Figure 3 middle). Indeed, we show theoretically that each auxiliary variable increases the ELBO and hence decreases the KL-divergence to the true posterior. However, the precise condition for convergence to the true posterior remains an open question.

3. Theoretical Results

We claim that the refinement process must improve the variational approximation over the initial mean-field approximation as measured by the ELBO.

This claim is formalized in the following proposition.

Proposition 1. *Let*

$$\mathrm{ELBO}_{ref} = \mathop{\mathbb{E}}_{q_{ref}} \left[\log p(\boldsymbol{y}|\boldsymbol{x}, \mathbf{w}) \right] - \mathrm{KL}\left[q_{ref}(\mathbf{w}) \;||\; p(\mathbf{w}) \right]$$

be the ELBO of the refined posterior (where q_{ref} is the distribution that our process generates samples from), let

$$\mathrm{ELBO}_{aux} = \mathop{\mathbb{E}}_{q_{ref}} \left[\log p(\boldsymbol{y}|\boldsymbol{x}, \mathbf{w}) \right] - \mathrm{KL}\left[q_{ref}(\mathbf{a}_{1:K}) \;||\; p(\mathbf{a}_{1:K}) \right]$$

be the ELBO accounting for the auxiliary variables, and let

$$\mathrm{ELBO}_{init} = \mathop{\mathbb{E}}_{q_{\phi_0}} \left[\log p(\boldsymbol{y}|\boldsymbol{x}, \mathbf{w}) \right] - \mathrm{KL}\left[q_{\phi_0}(\mathbf{w}) \;||\; p(\mathbf{w}) \right]$$

be the ELBO of the initial variational approximation. Then, the following inequalities hold:

$$\mathrm{ELBO}_{ref} \geq \mathrm{ELBO}_{aux} \geq \mathrm{ELBO}_{init}.$$

Thus, $\mathrm{ELBO}_{\mathrm{ref}}$, the ELBO of the distribution that we are generating samples from is greater than, or equal to $\mathrm{ELBO}_{\mathrm{init}}$, the ELBO of the initial mean-field approximation.

3.1. Proof of $\mathrm{ELBO}_{ref} \geq \mathrm{ELBO}_{aux}$

This is a consequence of the fact that $\mathbf{a}_{1:K}$ fully determines $\mathbf{w}$.

Proof.

$$\begin{aligned}
\text{ELBO}_{\text{ref}} - \text{ELBO}_{\text{aux}} &= \text{KL}[\, q_{\text{ref}}(\mathbf{a}_{1:K}) \;||\; p(\mathbf{a}_{1:K}) \,] - \text{KL}[\, q_{\text{ref}}(\mathbf{w}) \;||\; p(\mathbf{w}) \,] \\
&= \mathop{\mathbb{E}}_{q_{\text{ref}}(\mathbf{a}_{1:K})} \Big[\log \frac{q_{\text{ref}}(\mathbf{a}_{1:K})}{p(\mathbf{a}_{1:K})} - \log \frac{q_{\text{ref}}(\mathbf{w})}{p(\mathbf{w})} \Big] \\
&= \mathop{\mathbb{E}}_{q_{\text{ref}}(\mathbf{w})} \Big[\mathop{\mathbb{E}}_{q_{\text{ref}}(\mathbf{a}_{1:K}|\mathbf{w})} \Big[\log \frac{q_{\text{ref}}(\mathbf{a}_{1:K})}{p(\mathbf{a}_{1:K})} - \log \frac{q_{\text{ref}}(\mathbf{w})}{p(\mathbf{w})} \Big] \Big] \\
&= \mathop{\mathbb{E}}_{q_{\text{ref}}(\mathbf{w})} \Big[\mathop{\mathbb{E}}_{q_{\text{ref}}(\mathbf{a}_{1:K}|\mathbf{w})} \Big[\log \frac{q_{\text{ref}}(\mathbf{a}_{1:K}|\mathbf{w})}{p(\mathbf{a}_{1:K}|\mathbf{w})} \Big] \Big] \\
&= \mathop{\mathbb{E}}_{q_{\text{ref}}(\mathbf{w})} \Big[\underbrace{\text{KL}[\, q_{\text{ref}}(\mathbf{a}_{1:K}|\mathbf{w}) \;||\; p(\mathbf{a}_{1:K}|\mathbf{w}) \,]}_{\geq 0} \Big] \geq 0 \,,
\end{aligned}$$

where line 4 follows using Bayes' theorem: $q_{\text{ref}}(\mathbf{a}_{1:K}|\mathbf{w}) = \frac{q_{\text{ref}}(\mathbf{w}|\mathbf{a}_{1:K}) q_{\text{ref}}(\mathbf{a}_{1:K})}{q_{\text{ref}}(\mathbf{w})}$, $p(\mathbf{a}_{1:K}|\mathbf{w}) = \frac{p(\mathbf{w}|\mathbf{a}_{1:K}) p(\mathbf{a}_{1:K})}{p(\mathbf{w})}$ and that $q_{\text{ref}}(\mathbf{w}|\mathbf{a}_{1:K}) = p(\mathbf{w}|\mathbf{a}_{1:K}) = \delta_{Dirac}(\mathbf{w} - \sum_{k=1}^{K} \mathbf{a}_k)$. The proof is concluded using the non-negativity of the KL-divergence. □

Note that ELBO_{ref} is a valid ELBO—it is a lower bound to the marginal likelihood $\log p(y|x) \geq \text{ELBO}_{\text{ref}}$. Therefore, optimizing ELBO_{ref} through our sampling procedure decreases the KL divergence between q_{ref} and the true posterior.

3.2. *Proof of* $\text{ELBO}_{aux} \geq \text{ELBO}_{init}$

We prove this by demonstrating that improvement in the ELBO can be guaranteed in our method under the assumption that the conditional variational posterior $q_{\phi_{k-1}}(\mathbf{w}|\mathbf{a}_k)$ is within the variational family of q_{ϕ_k}, i.e., there exists ϕ_k^*, such that $q_{\phi_k^*}(\mathbf{w}) = q_{\phi_{k-1}}(\mathbf{w}|\mathbf{a}_k) \propto p(\mathbf{a}_k|\mathbf{w}, \mathbf{a}_{1:k-1}) q_{\phi_{k-1}}(\mathbf{w})$ for $k = 1 \ldots K$.

The central idea is to show that by initializing ϕ_k at ϕ_k^*, the variational distribution remains unchanged—therefore, $\text{ELBO}_{\text{aux}} = \text{ELBO}_{\text{init}}$. Then, as we optimize ϕ_k, we are optimizing the terms in ELBO_{aux} through $\mathcal{L}_{|a_{1:k}}(\phi_k)$. Therefore, $\text{ELBO}_{\text{aux}} \geq \text{ELBO}_{\text{init}}$.

Proof. We prove $\text{ELBO}_{\text{aux}} \geq \text{ELBO}_{\text{init}}$ by demonstrating that improvement in the ELBO can be guaranteed in our method under the assumption that the conditional variational posterior $q_{\phi_{k-1}}(\mathbf{w}|\mathbf{a}_k)$ is within the variational family of $q_{\phi_k}(\mathbf{w})$. i.e.,

$$\forall k \in \{1 \ldots K\} \exists \phi_k^* \text{s.t. } q_{\phi_k^*}(\mathbf{w}) = q_{\phi_{k-1}}(\mathbf{w}|\mathbf{a}_k) \propto p(\mathbf{a}_k|\mathbf{w}, \mathbf{a}_{1:k-1}) q_{\phi_{k-1}}(\mathbf{w}) \,. \tag{5}$$

This assumption holds for all exponential families of distributions.

The objective being optimized in each refinement step is

$$\mathcal{L}_{|\mathbf{a}_{1:k}}(\phi_k) = \mathbb{E}_{q_{\phi_k}(\mathbf{w})} \Big[p(\boldsymbol{y}|\boldsymbol{x}, \mathbf{w}) - \log \frac{q_{\phi_k}(\mathbf{w})}{p(\mathbf{w}|\mathbf{a}_{1:k})} \Big] \,. \tag{6}$$

From our assumption in Equation (5), it follows that

$$\mathcal{L}_{|\mathbf{a}_{1:k}}(\phi_k) \geq \mathcal{L}_{|\mathbf{a}_{1:k}}(\phi_k^*) \tag{7}$$

when we reach the global optima $\phi_k \leftarrow \arg\max_{\phi_k} \mathcal{L}_{|\mathbf{a}_{1:k}}(\phi_k)$. Even in the case when the optimizer is unable to find the global maximum, it is reasonable to assume that $\mathcal{L}_{|\mathbf{a}_{1:k}}(\phi_k) \geq \mathcal{L}_{|\mathbf{a}_{1:k}}(\phi_k^*)$, given that we initialize ϕ_k at ϕ_k^*.

The proof is based on mathematical induction on l. We show that for $l = 0 \ldots K$,

$$\mathop{\mathbb{E}}_{\substack{\mathbf{a}_k \sim q_{\phi_{k-1}}(\mathbf{a}_k) \\ k=1 \ldots l}} \Big[\mathcal{L}_{|\mathbf{a}_{1:l}}(\phi_l) - \sum_{k=1}^{l} \log \frac{q_{\phi_{k-1}}(\mathbf{a}_k)}{p(\mathbf{a}_k|\mathbf{a}_{1:k-1})} \Big] \geq \text{ELBO}_{\text{init}} \,, \tag{8}$$

which holds at $l = 0$, since $\mathcal{L}_{|}(\phi_0) = \text{ELBO}_{\text{init}}$.

Notice that for $k = 0 \dots K-1$,

$$\begin{aligned}
\mathbb{E}_{\mathbf{a}_{k+1}\sim q_{\phi_k}}\left[\mathcal{L}_{|\mathbf{a}_{1:k+1}}(\phi_{k+1})\right] &\geq \mathbb{E}_{\mathbf{a}_{k+1}\sim q_{\phi_k}}\left[\mathcal{L}_{|\mathbf{a}_{1:k+1}}(\phi^*_{k+1})\right] \\
&= \mathbb{E}_{\mathbf{a}_{k+1}\sim q_{\phi_k}}\left[\mathbb{E}_{q_{\phi_k}(\mathbf{w}|\mathbf{a}_{k+1})}\left[p(\boldsymbol{y}|\boldsymbol{x},\mathbf{w}) - \log\frac{q_{\phi_k}(\mathbf{w}|\mathbf{a}_{k+1})}{p(\mathbf{w}|\mathbf{a}_{1:k+1})}\right]\right] \\
&= \mathbb{E}_{\mathbf{a}_{k+1}\sim q_{\phi_k}}\left[\mathbb{E}_{q_{\phi_k}(\mathbf{w}|\mathbf{a}_{k+1})}\left[p(\boldsymbol{y}|\boldsymbol{x},\mathbf{w}) - \log\frac{q_{\phi_k}(\mathbf{w})}{p(\mathbf{w}|\mathbf{a}_{1:k})} + \log\frac{q_{\phi_k}(\mathbf{a}_{k+1})}{p(\mathbf{a}_{k+1}|\mathbf{a}_{1:k})}\right]\right] \\
&= \mathcal{L}_{|\mathbf{a}_{1:k}}(\phi_k) + \mathbb{E}_{\mathbf{a}_{k+1}\sim q_{\phi_k}}\left[\log\frac{q_{\phi_k}(\mathbf{a}_{k+1})}{p(\mathbf{a}_{k+1}|\mathbf{a}_{1:k})}\right],
\end{aligned} \tag{9}$$

where line 1 follows using Equation (7) and line 3 follows using Bayes' theorem: $q_{\phi_k}(\mathbf{w}|\mathbf{a}_{k+1}) = \frac{p(\mathbf{a}_{k+1}|\mathbf{w},\mathbf{a}_{1:k})q_{\phi_k}(\mathbf{w})}{q_{\phi_k}(\mathbf{a}_{k+1})}$ and $p(\mathbf{w}|\mathbf{a}_{1:k+1}) = \frac{p(\mathbf{a}_{k+1}|\mathbf{w},\mathbf{a}_{1:k})p(\mathbf{w}|\mathbf{a}_{1:k})}{p(\mathbf{a}_{k+1}|\mathbf{a}_{1:k})}$. After rearranging,

$$\mathcal{L}_{|\mathbf{a}_{1:k}}(\phi_k) \leq \mathbb{E}_{\mathbf{a}_{k+1}\sim q_{\phi_k}}\left[\mathcal{L}_{|\mathbf{a}_{1:k+1}}(\phi_{k+1}) - \log\frac{q_{\phi_k}(\mathbf{a}_{k+1})}{p(\mathbf{a}_{k+1}|\mathbf{a}_{1:k})}\right]. \tag{10}$$

Substituting this into the inductive hypothesis at $k = l$ proves the inductive step as shown next:

$$\begin{aligned}
&\text{ELBO}_{\text{init}} \\
&\leq \mathop{\mathbb{E}}_{\substack{\mathbf{a}_k\sim q_{\phi_{k-1}}\\ k=1\dots l}}\left[\mathcal{L}_{|\mathbf{a}_{1:l}}(\phi_l) - \sum_{k=1}^{l}\log\frac{q_{\phi_{k-1}}(\mathbf{a}_k)}{p(\mathbf{a}_k|\mathbf{a}_{1:k-1})}\right] \\
&\leq \mathop{\mathbb{E}}_{\substack{\mathbf{a}_k\sim q_{\phi_{k-1}}\\ k=1\dots l}}\left[\mathbb{E}_{\mathbf{a}_{l+1}\sim q_{\phi_l}}\left[\mathcal{L}_{|\mathbf{a}_{1:l+1}}(\phi_{l+1}) - \log\frac{q_{\phi_l}(\mathbf{a}_{l+1})}{p(\mathbf{a}_{l+1}|\mathbf{a}_{1:l})}\right] - \sum_{k=1}^{l}\log\frac{q_{\phi_{k-1}}(\mathbf{a}_k)}{p(\mathbf{a}_k|\mathbf{a}_{1:k-1})}\right] \\
&= \mathop{\mathbb{E}}_{\substack{\mathbf{a}_k\sim q_{\phi_{k-1}}\\ k=1\dots l+1}}\left[\mathcal{L}_{|\mathbf{a}_{1:l+1}}(\phi_{l+1}) - \sum_{k=1}^{l+1}\log\frac{q_{\phi_{k-1}}(\mathbf{a}_k)}{p(\mathbf{a}_k|\mathbf{a}_{1:k-1})}\right]
\end{aligned} \tag{11}$$

To finish the proof, examine the case $l = K$. Notice that

$$\mathcal{L}_{|\mathbf{a}_{1:K}}(\phi_K) = \mathbb{E}_{q_{\phi_K}(\mathbf{w})}\left[p(\boldsymbol{y}|\boldsymbol{x},\mathbf{w}) - \frac{q_{\phi_K}(\mathbf{w})}{p(\mathbf{w}|\mathbf{a}_{1:K})}\right] = p(\boldsymbol{y}|\boldsymbol{x},\mathbf{w}), \tag{12}$$

since $\mathbf{a}_{1:K}$ fully determines $\mathbf{w}$, i.e., $q_{\phi_K}(\mathbf{w}) = p(\mathbf{w}|\mathbf{a}_{1:K}) = \delta_{Dirac}(\mathbf{w} - \sum_{k=1}^{K}\mathbf{a}_k)$. Substituting Equation (12) in at $l = K$ yields the desired result:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\substack{\mathbf{a}_k\sim q_{\phi_{k-1}}\\ k=1\dots K}}\left[\mathcal{L}_{|\mathbf{a}_{1:K}}(\phi_K) - \sum_{k=1}^{K}\log\frac{q_{\phi_{k-1}}(\mathbf{a}_k)}{p(\mathbf{a}_k|\mathbf{a}_{1:k-1})}\right] \\
&= \mathop{\mathbb{E}}_{\substack{\mathbf{a}_k\sim q_{\phi_{k-1}}\\ k=1\dots K}}\left[p(\boldsymbol{y}|\mathbf{w},\boldsymbol{x}) - \sum_{k=1}^{K}\log\frac{q_{\phi_{k-1}}(\mathbf{a}_k)}{p(\mathbf{a}_k|\mathbf{a}_{1:k-1})}\right] \\
&= \mathop{\mathbb{E}}_{q_{\text{ref}}}\left[\log p(\boldsymbol{y}|\boldsymbol{x},\mathbf{w})\right] - \text{KL}[\,q_{\text{ref}}(\mathbf{a}_{1:K})\;||\;p(\mathbf{a}_{1:K})\,] \\
&= \text{ELBO}_{\text{aux}} \geq \text{ELBO}_{\text{init}},
\end{aligned} \tag{13}$$

concluding the proof. □

Note that this result implies that ELBO_{aux} must grow with each auxiliary variable. We demonstrate this empirically by estimating ELBO_{aux} as we sample the auxiliary variables in a neural network. The result is shown on Figure 4. We see that ELBO_{aux} grows after each iteration, exhibiting a stair pattern.

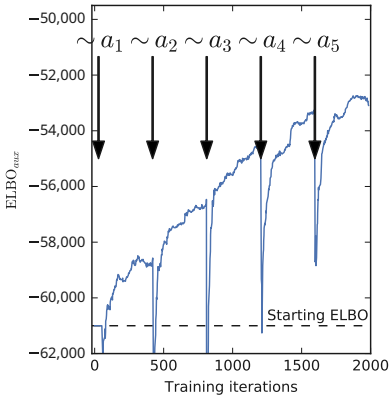

Figure 4. ELBO_{aux} is increasing as we sample the auxiliary variables. Calculated single sample Monte Carlo estimate of the expectation: $\text{ELBO}_{\text{aux}} = \mathbb{E}\left[\log p(\boldsymbol{y}|\boldsymbol{x}, \mathbf{w}) - \sum_{k=1}^{K} \log \frac{q_{\phi_{k-1}}(\mathbf{a}_k)}{p(\mathbf{a}_k|\mathbf{a}_{1:k-1})}\right]$ (Equation (13)). The sudden drops after sampling are optimizer artefacts because the momentum is reset after sampling. LeNet-5/CIFAR10.

4. Experimental Results

We showcase our method on two example tasks: inference in a Bayesian neural network and posterior sampling in a contextual bandit task.

4.1. Inference in Deep Neural Networks

The goal of this experiment is twofold. First, we empirically confirm the improvement in the ELBO, and second, we quantify the improvement in the uncertainty estimates due to the refinement. We conduct experiments on regression and classification benchmarks using Bayesian neural networks as the underlying model. We look at the marginal log-likelihood (MLL) of the predictions, as well as accuracy in classification tasks.

We used three baseline models for comparison: mean-field variational inference, multiplicative normalizing flows (MNF), and deep ensemble models. For all methods, we used a batch size of 256 and the Adam optimizer with the default learning rate of 0.001. The hyperparameters of each baseline were tuned using a Bayesian optimization package. We found batch size and learning rate to be consistent across methods.

First, Variational inference (VI, [8,9]). Naturally, we investigate the improvement of our method over variational inference with a mean-field Gaussian posterior approximation. We do inference over all weights and biases with a Gaussian prior centered at 0, the variance of the prior is tuned through empirical Bayes, and the model is trained for 30,000 iterations.

Second, Multiplicative normalizing flows (MNF, [10]). In this work, the posterior means are augmented with a multiplier from a flexible distribution parameterized by the masked RealNVP. This model is trained with the default flow parameters for 30,000 iterations.

Third, Deep ensemble models [11]. Deep ensemble models are shown to be surprisingly effective at quantifying uncertainty. For the regression datasets, we used adversarial training ($\epsilon = 0.01$), whereas in classification we did not (since adversarial training did not give an improvement in the classification benchmarks). For each dataset, we trained 10 ensemble members for 5000 iterations each.

Finally, our work, Refined VI. After training the initial mean-field approximation, we generate $M = 10$ refined samples $\mathbf{w}_{1:M}$, each with $K = 5$ auxiliary variables. The means on the prior distribution for the auxiliary variables are fixed at 0, and their prior variances form a geometric series (the intuition is that the auxiliary variables carry roughly equal information this way): $\sigma^2_{a_k} = 0.7\left(\sigma^2_w - \sum_{l=1}^{k-1} \sigma^2_{a_l}\right)$ for $k = 1 \dots K$. We experimented with different ratios between 0 and 1 for the geometric series and we found that 0.7 worked well. In each refinement iteration, we optimized the posterior with Adam [12] for 200 iterations. To keep the training stable, we kept the learning rate proportional to the standard deviation

of the conditional posterior: in iteration k, $\text{lr} = 0.001 \times 0.3^{\frac{k}{2}}$. Our code is available at https://github.com/google/edward2/experimental/auxiliary_sampling.

Following [13], we evaluate the methods on a set of UCI regression benchmarks on a feed forward neural network with a single hidden layer containing 50 units with a ReLU activation function (Table 1). The datasets used a random 80–20 split for training and testing, and we utilize the local reparametrization trick [14].

Table 1. Refining improves the ELBO across all regression benchmarks. Results on the UCI regression benchmarks with a single hidden layer containing 50 units. Metrics: marginal log-likelihood (MLL, higher is better), and the evidence lower bound (ELBO higher is better). The mean values and standard deviations are shown in the table. Bolded numbers indicate the highest ELBO (ELBO_{aux} is a lower bound to ELBO_{ref}, which is the true ELBO) and underlined numbers indicate the highest MLL.

	Deep Ensemble	**MNF**	**VI**		**Refined VI (This Work)**	
	MLL	**MLL**	**MLL**	**ELBO**	**MLL**	**ELBO_{aux}**
Boston	−9.136 ± 5.719	−2.920 ± 0.133	−2.874 ± 0.151	−668.2 ± 7.6	−2.851 ± 0.185	**−630.3** ± 7.7
Concrete	−4.062 ± 0.130	−3.202 ± 0.055	−3.138 ± 0.063	−3248.1 ± 68.5	−3.131 ± 0.062	**−3071.1** ± 64.0
Naval	3.995 ± 0.013	3.473 ± 0.007	5.969 ± 0.245	53,440.7 ± 2047.3	6.128 ± 0.171	**54,882.6** ± 1228.3
Energy	−0.666 ± 0.058	−0.756 ± 0.054	−0.749 ± 0.068	−1296.7 ± 66.3	−0.707 ± 0.094	**−1192.3** ± 62.0
Yacht	−0.984 ± 0.104	−1.339 ± 0.170	−1.749 ± 0.232	−928.7 ± 112.9	−1.626 ± 0.231	**−790.0** ± 84.7
Kin8nm	1.135 ± 0.012	1.125 ± 0.022	1.066 ± 0.019	6071.2 ± 61.7	1.069 ± 0.018	**6172.7** ± 67.6
Power	−3.935 ± 0.140	−2.835 ± 0.033	−2.826 ± 0.020	−22,496.5 ± 130.4	−2.820 ± 0.024	**−22,368.9** ± 85.3
Protein	−3.687 ± 0.013	−2.928 ± 0.0	−2.926 ± 0.010	−108,806.007 ± 174.5	−2.923 ± 0.009	**−108,597.5** ± 158.4
Wine	−0.968 ± 0.079	−0.963 ± 0.027	−0.973 ± 0.054	−1346.1 ± 18.0	−0.968 ± 0.056	**−1311.8** ± 17.4

On these benchmarks, refined VI consistently improves both the ELBO and the MLL estimates over VI. For refined VI, the ELBO_{ref} cannot be calculated exactly, but ELBO_{aux} provides a lower bound to it, which we can estimate using Equation (13). Note that the gains in MLL are small in this case. Nevertheless, refined VI is one of the best performing approaches on 7 out of the 9 datasets.

We examine the performance on commonly used image classification benchmarks (Table 2) using LeNet5 architecture [15]. We use the local reparametrization trick [14] for the dense layers and Flipout [16] for the convolutional layers to reduce the gradient noise. We do not use data augmentation in order to stay consistent with the Bayesian framework.

Table 2. Refining improves the ELBO across all image classification benchmarks. Results on image classification benchmarks with the LeNet-5 architecture, *without data augmentation*. Metrics: marginal log-likelihood (MLL, higher is better), accuracy (Acc, higher is better), and the evidence lower bound (ELBO higher is better). Means and standard deviations are shown. Bolded numbers indicate the highest ELBO (ELBO_{aux} is a lower bound to ELBO_{ref}, which is the true ELBO) and underlined numbers indicate the highest MLL.

	Deep Ensemble	**MNF**	**VI**		**Refined VI (This Work)**	
	MLL & Acc	**MLL & Acc**	**MLL & Acc**	**ELBO**	**MLL & Acc**	**ELBO_{aux}**
mnist	−0.017 ± 0.001 99.4% ± 0.0	−0.034 ± 0.002 99.1% ± 0.1	−0.032 ± 0.001 99.1% ± 0.1	−7618.5 ± 47.5	−0.025 ± 0.001 99.2% ± 0.0	**−6310.8** ± 42.3
fashion_mnist	−0.201 ± 0.002 93.1% ± 0.1	−0.255 ± 0.004 90.7% ± 0.2	−0.255 ± 0.003 90.7% ± 0.1	−22,830.3 ± 232.6	−0.241 ± 0.004 91.3% ± 0.2	**−20,438.9** ± 79.6
cifar10	−0.791 ± 0.009 76.3% ± 0.3	−0.795 ± 0.013 72.8% ± 0.6	−0.815 ± 0.004 72.3% ± 0.5	−57,257.8 ± 299.5	−0.768 ± 0.007 73.5% ± 0.5	**−50,989.2** ± 238.9

On the classification benchmarks, we again are able to confirm that the refinement step consistently improves both the ELBO and the MLL over VI, with the MLL differences being more significant here than in the previous experiments. Refined VI is unable to outperform deep ensembles in classification accuracy, but it does outperform them in MLL on the largest dataset, CIFAR10.

To demonstrate the performance on larger scale models, we apply the refining algorithm to residual networks [17] with 20 layers (based on Keras's ResNet implementation).

We look at two models: a standard ResNet, where inference is done over every residual block and a hybrid model (ResNet Hybrid [18]), where inference is only done over the final layer of each residual block, and every other layer is treated as a regular layer. For this model, we used a batch-size of 256 and we decayed the learning rate starting from 0.001 over 200 epochs. We used 10 auxiliary variables each reducing the prior variance by a factor of 0.5. Results are shown in Table 3.

Table 3. Results on CIFAR10 with the ResNet architecture, *without data augmentation*. We observe that our method not only improves significantly in MLL over the VI baseline, but it also significantly improves in accuracy over the strong ensemble baseline. Metrics: marginal log-likelihood (MLL, higher is better), accuracy (Acc, higher is better), and the evidence lower bound (ELBO higher is better). Note that the non-hybrid and the hybrid models are equivalent when trained deterministically. The best MLL result is highlighted in bold.

	Deep Ensemble		**VI**		**Refined VI (This Work)**	
	MLL	**Acc**	**MLL**	**Acc**	**MLL**	**Acc**
ResNet	−0.698	82.7%	−0.795	72.6%	**−0.696**	75.5%
ResNet + BatchNorm	**−0.561**	83.6%	−0.672	77.6%	−0.593	79.7%
ResNet Hybrid	−0.698	82.7%	−0.465	84.2%	**−0.432**	85.8%
ResNet Hybrid + BatchNorm	−0.561	83.6%	−0.465	84.0%	**−0.423**	85.6%

Batch normalization [19] provides a substantial improvement for VI though, this improvement interestingly disappears for the hybrid model. The refined hybrid model outperforms the recently proposed natural gradient VI method by [20] in both MLL and accuracy, but it is still behind some non-Bayesian uncertainty estimation methods [21].

4.2. Computational Costs

When introducing a novel algorithm for variational inference, we must discuss the computational costs. The computational complexity grows linearly with both K and M, resulting in an overall $O(KM)$ runtime. The memory requirement is $O(M)$ as it grows linearly with M. For the neural network models, the computational cost of generating the posterior samples is ∼30% of the cost of training the initial mean-field approximation (LeNet-5/CIFAR10 on an NVIDIA P100 GPU using TensorFlow). In practice, we recommend tuning the number of auxiliary variables for the given application; using more auxiliary variables always improves the posterior approximation, but they come with additional computational overhead.

4.3. Thompson Sampling

Generating posterior samples for Thompson sampling [22,23] in a contextual bandit problem is an ideal use case for the refinement algorithm. Refinement allows one to trade-off computational complexity for a higher quality approximation to the posterior. This can be ideal for Thompson sampling where more expensive objectives often warrant spending time computing better approximations.

Thompson sampling works by sampling a hypothesis from the approximate posterior to decide on each action. This balances exploration and exploitation, since probable hypotheses are tested more frequently than improbable ones. In each step,

1. Sample $\mathbf{w} \sim q_\phi(\mathbf{w})$;
2. Take action $\arg\max_a \mathbb{E}_{p(r|c,a,\mathbf{w})}[r]$, where r is the reward that is determined by the context c, the action a taken, and the unobserved model parameters $\mathbf{w}$;
3. Observe reward r and update the approximate posterior $q_\phi(\mathbf{w})$.

We look at the mushroom task [9,24], where the agent is presented with a number of mushrooms that they can choose to eat or pass. The mushrooms are either edible or poisonous. Eating an edible mushroom always yield a reward of 5, while eating a poisonous mushroom yield a reward 5 with probability 50% and −35 with probability 50%. Passing a mushroom gives no reward.

To predict the distribution of the rewards, the agent uses a neural network with 23 inputs and two outputs. The inputs are the 22 observed attributes of the mushrooms and the proposed action (1 for eating and 0 for passing). The output is the mean expected reward. The network has a standard feed-forward architecture with two hidden layers containing 100 hidden units each, with ReLU activations throughout. For the prior, we used a standard Gaussian distribution over the weights.

For the variational posterior, we use a mean-field Gaussian approximation that we update for 500 iterations after observing each new reward. For the updates, we use batches of 64 randomly sampled rewards with an Adam optimizer with learning rate 10^{-3}. In refined sampling, we used two auxiliary variables: $\mathbf{w} = \mathbf{a}_1 + \mathbf{a}_2$ with $p(\mathbf{a}_1) = \mathcal{N}(0, 0.7)$ and $p(\mathbf{a}_2) = \mathcal{N}(0, 0.3)$. To obtain a high quality sample for prediction, we first draw $\mathbf{a}_1$ using the main variational approximation and then refine the posterior over $\mathbf{a}_2$ for 500 iterations. After using the refined sample for prediction, we discard it and update the main variational approximation using the newly observed reward (for 500 iterations). In our experiments, we used three posterior samples to calculate the expected reward, which helps to emphasize exploitation compared to using a single sample.

As baselines, we show the commonly used ϵ-greedy algorithm, where the agent takes the action with the highest expected reward according to the maximum-likelihood solution with probability $1 - \epsilon$, and takes a random action with probability ϵ.

We measure the performance using the cumulative regret. The cumulative regret measures the difference between our agent and an omniscient agent that makes the optimal choice each time. Lower regret indicates better performance. Figure 5 depicts the results. We see that the refined agent has lower regret throughout, which shows that the higher quality posterior samples translate to improved performance. Until about 3000 iterations, the ϵ-greedy algorithms perform well, but they are overtaken by Thompson sampling as the posterior tightens and the agent shifts focus to exploitation.

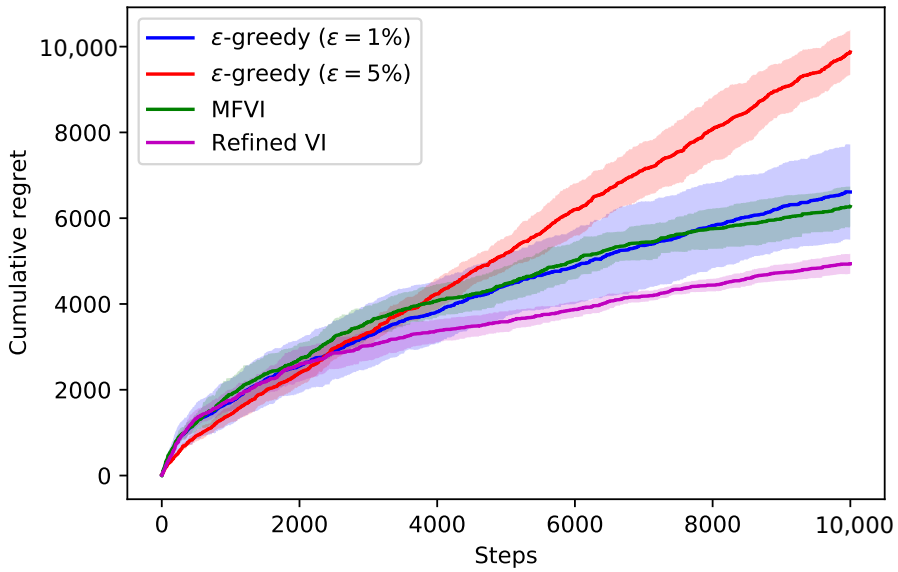

Figure 5. The performances of ϵ-greedy, Mean-field VI, and Refined VI on the mushrooms contextual bandit task. Lower regret is better. The mean and standard deviations are shown from 5 runs with different random seeds.

5. Related Works

Although, in theory, the Bayesian approach can accurately capture uncertainty, in practice, we find that exact inference is computationally infeasible in most scenarios, and thus, we have to resort to approximate inference methods. There is a wealth of research on approximate inference methods; here, we focus on works closely related to this paper.

Variational inference [25] tries to approximate the true posterior distribution over parameters with a variational posterior from a simple family of distributions. Mean-field VI, which for neural networks traces back to [26], uses independent Gaussian distributions over the parameters to try to capture the posterior. The advantage of the mean-field approximation is that the network can be efficiently trained using the reparameterization trick [27], and the variational posterior has a proper density over the parameter space, which then can be used across tasks, such as continual learning [20,28] and contextual bandits [29]. Recently, [10] showed that normalizing flows can be used to further increase the flexibility of the variational posterior. [30] provide a detailed survey of recent advances in VI.

Our method is a novel variant of the auxiliary variable approaches to VI [31,32] that increase the flexibility of the variational posterior through the use of auxiliary variables. The key distinction, however, is that instead of trying to train a complex variational approximation over the joint distribution, we iteratively train simple mean-field approximations at the sampled values of the auxiliary variables. Although this poses an $O(MK)$ overhead (K is the number of auxiliary variables and M is the number of posterior samples) over mean-field VI, the training itself remains straightforward and efficient. The introduction of every new auxiliary variable increases the flexibility of the posterior approximation. In contrast to MCMC methods, it is unclear whether the algorithm approaches the true posterior in the limit of infinitely many auxiliary variables.

There are also numerous methods that start with an initial variational approximation and refine it through a few MCMC steps [33–35]. The distinction from our algorithm is that we refine the posterior starting at large scale and iteratively move towards smaller scale refinements, whereas these methods only refine the posterior at the scale of the MCMC steps [36–38] used boosting to refine the variational posterior, where they iteratively added parameters, such as mixture components to minimize the residual of the ELBO. Our method does not add parameters at training time but instead iteratively refines the samples through the introduction of auxiliary variables. We do not include these in our baselines since they have yet to be applied to Bayesian multi-layer neural networks.

Further related works include methods that iteratively refine the posterior in latent variable models [39–42]. These methods, however, focus on reducing the amortization gap and do not increase the flexibility of the variational approximation.

Lastly, there are non-Bayesian strategies for estimating epistemic uncertainty in deep learning. Bootstrapping [43] and deep ensembles [11] may be the most promising. Deep ensembles, in particular, have been demonstrated to achieve strong performance on benchmark regression and classification problems and uncertainty benchmarks including out-of-distribution detection [11] and prediction under distribution shift [18]. Both methods rely on constructing a set of independently trained models to estimate the uncertainty. Intuitively, the amount of disagreement across models reflects the uncertainty in the ensemble prediction. In order to induce diversity among the ensemble members, bootstrapping subsamples the training set for each member while deep ensembles use the randomness in weight initialization and mini-batch sampling.

6. Conclusions

In this work, we investigated a novel method for generating samples from a highly flexible posterior approximation, which works by starting with a mean-field approximation and locally refining it in selected regions. We demonstrated that the samples are able to capture dependencies and multi-modality. Furthermore, we showed both theoretically and empirically that the method always improves the ELBO of the initial mean-field approximation and demonstrated its improvement on a hierarchical inference problem, a deep learning benchmark and a contextual bandit task.

Author Contributions: Conceptualization, M.H. and J.G.; methodology, M.H., J.S., D.T., J.G. and J.M.H.-L.; software, M.H., J.S. and D.T.; writing—original draft preparation, M.H.; writing—review and editing, M.H., J.S., D.T., J.G. and J.M.H.-L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by EPSRC.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Analytical Forms of $q_{\boldsymbol{\phi}_{k-1}}(\mathbf{a}_k)$ and $q_{\boldsymbol{\phi}_{k-1}}(\mathbf{w}|\mathbf{a}_k)$

For a diagonal Gaussian prior distribution $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \sigma^2 I)$ ($\mathbf{0}$ denotes the $\mathrm{d}_{\mathbf{w}}$ dimensional zero vector and I denotes the $\mathrm{d}_{\mathbf{w}} \times \mathrm{d}_{\mathbf{w}}$ identity matrix where $\mathrm{d}_{\mathbf{w}}$ is the dimensionality of $\mathbf{w}$), we have $\mathbf{w} = \sum_{k=1}^{K} \mathbf{a}_k$, $p(\mathbf{a}_k) = \mathcal{N}(\mathbf{a}_k|\mathbf{0}, \sigma_k^2 I)$ for $k \in \{1, \ldots, K\}$ such that $\sum_{k=1}^{K} \sigma_k^2 = \sigma^2$.

The forms of approximate posterior over the auxiliary variables $q_{\phi_{k-1}}(\mathbf{a}_k)$ and the conditionals $q_{\phi_{k-1}}(\mathbf{w}|\mathbf{a}_k)$ can be computed in closed form. We only derive the result in the univariate case, but extending to the diagonal covariance case is straightforward.

First, let $p(a_k) = \mathcal{N}(a_k \,|\, \mu_k, \sigma_k^2)$. Now, define $b_k = \sum_{i=1}^{k} a_i$, $m_k = \sum_{i=k+1}^{K} \mu_i$ and $s_k^2 = \sum_{i=k+1}^{K} \sigma_i^2$. Since $z = \sum_{k=1}^{K} a_k$, using the formula for the conditional distribution of sums of Gaussian random variables (For Gaussian random variables X, Y with means μ_x, μ_y and variances σ_x^2, σ_y^2 and $Z = X + Y$, $p(x|z)$ is normally distributed with mean $\mu_x + (z - \mu_x - \mu_y)\frac{\sigma_x^2}{\sigma_x^2+\sigma_y^2}$ and variance $\frac{\sigma_x^2\sigma_y^2}{\sigma_x^2+\sigma_y^2}$), we obtain

$$p(a_k \,|\, a_{1:k-1}, w) = \mathcal{N}\left(a_k \;\middle|\; \mu_k + (w - b_{k-1} - m_{k-1})\frac{\sigma_k^2}{s_{k-1}^2}, \frac{s_k^2\sigma_k^2}{s_{k-1}^2}\right). \tag{A1}$$

Recall that

$$q_{\phi_{k-1}}(a_k) = \int p(a_k|a_{1:k-1}, w) q_{\phi_{k-1}}(w)\, \mathrm{d}w\,, \tag{A2}$$

and assume that we have already calculated $q_{\phi_{k-1}}(w) = \mathcal{N}\left(w \,|\, \nu_{k-1}, \rho_{k-1}^2\right)$. Notice that the quantity of interest is an integral of Gaussian densities, and hence after some algebraic manipulation, we obtain

$$q_{\phi_{k-1}}(a_k) = \mathcal{N}\left(a_k \;\middle|\; \mu_k + (\nu_{k-1} - b_{k-1} - m_{k-1})\frac{\sigma_k^2}{s_{k-1}^2}, \frac{s_k^2\sigma_k^2}{s_{k-1}^2} + \rho_{k-1}^2\frac{\sigma_k^4}{s_{k-1}^4}\right). \tag{A3}$$

Regarding $q_{\phi_{k-1}}(w|a_k)$, we have

$$q_{\phi_{k-1}}(w|a_k) = \frac{p(a_k \,|\, a_{1:k-1}, w) q_{\phi_{k-1}}(w)}{q_{\phi_{k-1}}(a_k)} \tag{A4}$$

using Bayes' rule. Again, we see that the desired quantity is a product of Gaussians, which we can derive to arrive at

$$q_{\phi_{k-1}}(w|a_k) = \mathcal{N}\left(w \;\middle|\; \frac{(a_k - \mu_k)\rho_{k-1}^2 s_{k-1}^2 + (b_{k-1} + m_{k-1})\sigma_k^2\rho_{k-1}^2 + \nu_{k-1}s_k^2 s_{k-1}^2}{\sigma_k^2\rho_{k-1}^2 + s_{k-1}^2 s_k^2}, \frac{\rho_{k-1}^2 s_{k-1}^2 s_k^2}{\sigma_k^2\rho_{k-1}^2 + s_{k-1}^2 s_k^2}\right). \tag{A5}$$

References

1. Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; Mané, D. Concrete problems in AI safety. *arXiv* **2016**, arXiv:1606.06565.
2. Gelman, A.; Carlin, J.B.; Stern, H.S.; Dunson, D.B.; Vehtari, A.; Rubin, D.B. *Bayesian Data Analysis*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2013.
3. Carpenter, B.; Gelman, A.; Hoffman, M.D.; Lee, D.; Goodrich, B.; Betancourt, M.; Brubaker, M.; Guo, J.; Li, P.; Riddell, A. Stan: A probabilistic programming language. *J. Stat. Softw.* **2017**, *76*, 1–32. [CrossRef]
4. Yao, Y.; Vehtari, A.; Simpson, D.; Gelman, A. Yes, but Did It Work?: Evaluating Variational Inference. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5581–5590.
5. Kucukelbir, A.; Ranganath, R.; Gelman, A.; Blei, D. Automatic variational inference in Stan. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 568–576.
6. Hoffman, M.D.; Gelman, A. The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.* **2014**, *15*, 1593–1623.
7. Salvatier, J.; Wiecki, T.V.; Fonnesbeck, C. Probabilistic programming in Python using PyMC3. *PeerJ Comput. Sci.* **2016**, *2*, e55. [CrossRef]
8. Graves, A. Practical variational inference for neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Granada, Spain, 12–14 December 2011; pp. 2348–2356.
9. Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; Wierstra, D. Weight uncertainty in neural network. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 1613–1622.
10. Louizos, C.; Welling, M. Multiplicative normalizing flows for variational Bayesian neural networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 2218–2227.
11. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6402–6413.
12. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
13. Hernández-Lobato, J.M.; Adams, R. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 1861–1869.
14. Kingma, D.P.; Salimans, T.; Welling, M. Variational Dropout and the Local Reparameterization Trick. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2015.
15. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. In *The Handbook of Brain Theory and Neural Networks*; A Bradford Book; MIT Press: Cambridge, MA, USA, 1995.
16. Wen, Y.; Vicol, P.; Ba, J.; Tran, D.; Grosse, R. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv* **2018**, arXiv:1803.04386.
17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
18. Ovadia, Y.; Fertig, E.; Ren, J.; Nado, Z.; Sculley, D.; Nowozin, S.; Dillon, J.V.; Lakshminarayanan, B.; Snoek, J. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–9 December 2019.
19. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
20. Osawa, K.; Swaroop, S.; Jain, A.; Eschenhagen, R.; Turner, R.E.; Yokota, R.; Khan, M.E. Practical Deep Learning with Bayesian Principles. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
21. Wen, Y.; Tran, D.; Ba, J. BatchEnsemble: An Alternative Approach to Efficient Ensemble and Lifelong Learning. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
22. Thompson, W.R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* **1933**, *25*, 285–294. [CrossRef]
23. Hernández-Lobato, J.M.; Requeima, J.; Pyzer-Knapp, E.O.; Aspuru-Guzik, A. Parallel and distributed Thompson sampling for large-scale accelerated exploration of chemical space. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1470–1479.
24. Guez, A. Sample-Based Search Methods for Bayes-Adaptive Planning. Ph.D. Thesis, UCL (University College London), London, UK, 2015.
25. Hinton, G.; Van Camp, D. Keeping neural networks simple by minimizing the description length of the weights. In Proceedings of the 6th Ann. ACM Conf. on Computational Learning Theory, Santa Cruz, CA, USA, 26–28 July 1993.
26. Peterson, C. A mean field theory learning algorithm for neural networks. *Complex Syst.* **1987**, *1*, 995–1019.
27. Kingma, D.P.; Welling, M. Auto-encoding variational Bayes. *arXiv* **2013**, arXiv:1312.6114.
28. Nguyen, C.V.; Li, Y.; Bui, T.D.; Turner, R.E. Variational continual learning. *arXiv* **2017**, arXiv:1710.10628.
29. Riquelme, C.; Tucker, G.; Snoek, J.R. Deep Bayesian Bandits Showdown. In Proceedings of the International Conference on Representation Learning, Vancouver, BC, Canada, 30 April–3 May 2018.

30. Zhang, C.; Butepage, J.; Kjellstrom, H.; Mandt, S. Advances in variational inference. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 2008–2026. [CrossRef] [PubMed]
31. Agakov, F.V.; Barber, D. An auxiliary variational method. In Proceedings of the International Conference on Neural Information Processing, Calcutta, India, 22–25 November 2004; pp. 561–566.
32. Ranganath, R.; Tran, D.; Blei, D. Hierarchical variational models. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; pp. 324–333.
33. Salimans, T.; Kingma, D.; Welling, M. Markov chain monte carlo and variational inference: Bridging the gap. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 1218–1226.
34. Zhang, Y.; Hernández-Lobato, J.M.; Ghahramani, Z. Ergodic measure preserving flows. *arXiv* **2018**, arXiv:1805.10377.
35. Ruiz, F.; Titsias, M. A Contrastive Divergence for Combining Variational Inference and MCMC. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 5537–5545.
36. Guo, F.; Wang, X.; Fan, K.; Broderick, T.; Dunson, D.B. Boosting variational inference. *arXiv* **2016**, arXiv:1611.05559.
37. Miller, A.C.; Foti, N.J.; Adams, R.P. Variational Boosting: Iteratively Refining Posterior Approximations. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
38. Locatello, F.; Dresdner, G.; Khanna, R.; Valera, I.; Raetsch, G. Boosting Black Box Variational Inference. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018.
39. Hjelm, D.; Salakhutdinov, R.R.; Cho, K.; Jojic, N.; Calhoun, V.; Chung, J. Iterative refinement of the approximate posterior for directed belief networks. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 4691–4699.
40. Cremer, C.; Li, X.; Duvenaud, D. Inference suboptimality in variational autoencoders. *arXiv* **2018**, arXiv:1801.03558.
41. Kim, Y.; Wiseman, S.; Miller, A.C.; Sontag, D.; Rush, A.M. Semi-amortized variational autoencoders. *arXiv* **2018**, arXiv:1802.02550.
42. Marino, J.; Yue, Y.; Mandt, S. Iterative amortized inference. *arXiv* **2018**, arXiv:1807.09356.
43. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]