VIB is Half Bayes

Alexander A. Alemi ALEMI@GOOGLE.COM
Warren R. Morningstar WMORNING@GOOGLE.COM
Ben Poole POOLEB@GOOGLE.COM
Ian Fischer IANSF@GOOGLE.COM
Joshua V. Dillon JVDILLON@GOOGLE.COM

Abstract

In discriminative settings such as regression and classification there are two random variables at play, the inputs X and the targets Y. Here, we demonstrate that the Variational Information Bottleneck can be viewed as a compromise between fully empirical and fully Bayesian objectives, attempting to minimize the risks due to finite sampling of Y only. We argue that this approach provides some of the benefits of Bayes while requiring only some of the work.

1. Introduction

Big models, big data, and maximum likelihood training are a proven recipe for learning powerful and generalizable neural network models. But training such large models on small data results in overfitting and poor performance. How can we achieve good performance from limited data?

Bayesian inference presents one such mechanism. Bayesian inference can be seen as minimizing a PAC-style upper bound on generalization performance from finite data (Morningstar et al., 2020; Masegosa, 2020; Germain et al., 2016). However, exactly performing Bayesian inference is costly, requiring careful tuning of MCMC methods or expressive variational distributions to match the posterior (Betancourt, 2018; Zhang et al., 2018).

Here, we show that training on multiple outputs Y for each input X can be beneficial, and derive a training objective which provides these benefits without actually having to collect multiple outputs for each input. The resulting objective matches the Variational Information Bottleneck (VIB) (Alemi et al., 2016), and provides a tractable alternative to Bayesian inference that loses some of the guarantees but retains much of the qualitative and quantitative performance.

2. Preliminaries

Consider training a neural network with parameters θ to output a stochastic representation $z \sim q(Z|x, \theta)$ for each input $x \in X$. From the representation, we can predict the target y with a fixed (parameter-free) *classifier* or *regressor* $p(y|z)$. At test time, we will form the predictive distribution $q(y|x, \theta) = \int \mathrm{d}z\, q(z|x, \theta)p(y|z)$.

Assuming the true data comes from some joint distribution $\nu(X, Y)$, we aim to learn a predictive distribution that is as close as possible to the true conditional distribution

$\nu(Y|X)$, as measured by the expected conditional KL divergence:[1]

$$\mathsf{E}_{\nu(X)}\left[\mathsf{KL}[\nu(Y|X), q(Y|X,\theta)]\right] = \int \mathrm{d}x\,\mathrm{d}y\,\nu(x,y)\log\frac{\nu(y|x)}{q(y|x,\theta)} \overset{\text{def}}{=} \mathcal{P}(\theta) - \mathsf{H}[\nu(Y|X)], \tag{1}$$

where $\mathsf{H}[\nu(Y|X)]$ is the true conditional entropy and we call $\mathcal{P}(\theta)$ the *true predictive risk*:

$$\mathcal{P}(\theta) \overset{\text{def}}{=} \mathsf{E}_{\nu(X,Y)}\log\left(-\,\mathsf{E}_{q(Z|X,\theta)}[p(Y|Z)]\right). \tag{2}$$

Invoking Jensen's inequality, we can upper bound $\mathcal{P}(\theta)$ with the *true classification risk*:

$$\mathcal{P}(\theta) \leq \mathcal{C}(\theta) \overset{\text{def}}{=} \mathsf{E}_{\nu(X,Y)}\,\mathsf{E}_{q(Z|X,\theta)}[-\log p(Y|Z)]. \tag{3}$$

$\mathcal{C}(\theta)$ measures how well we can predict the targets given a sample from our representation z of each input $x \in X$. Unlike $\mathcal{P}(\theta)$ which contains a log of an expectation, we can compute unbiased estimates of $\mathcal{C}$ using Monte-Carlo.

While we don't know the true distribution, we have access to n paired samples from this distribution, a *dataset* $D_n = \{(x_i, y_i)\}_i^n$. We need ways to approximate the true classification risk (eq. (3)) while only using a finite number of samples.

Maximum Likelihood (ML) tries to minimize the *empirical classification risk*:

$$\widehat{C}(\theta; D_n) \overset{\text{def}}{=} \mathsf{E}_{\hat{\nu}_n(X,Y)}\,\mathsf{E}_{q(Z|X,\theta)}[-\log p(Y|Z)], \tag{4}$$

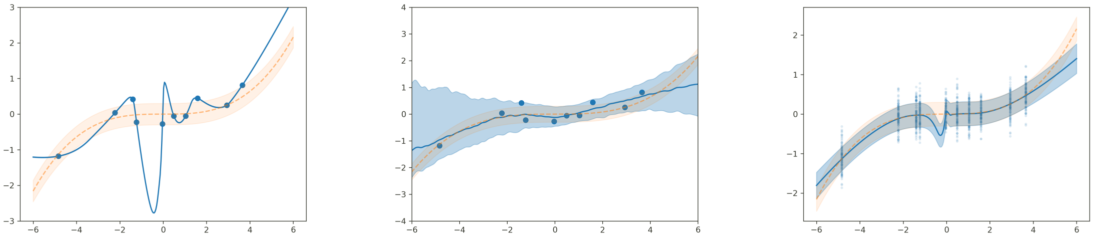
approximating the expectation with respect to the true distribution with an average over the observed samples. From the perspective of variational optimization, ML can concentrate on the deterministic representation $q(Z|x,\theta) = \delta(z - f_\theta(x))$ which best predicts the observed target y for each input x. Unfortunately, ML with finite samples provides no guaranteed relationship to the true classification risk. In other words, eq. (4) is neither an upper nor a lower bound on eq. (3).

As an illustration, in fig. 1(*a*) we show what happens if we try to minimize eq. (4) using a neural network with a two dimensional representation $z = (\mu, \sigma^2)$. This is used to parameterize the mean and standard deviation of a conditional normal distribution $y \sim \mathsf{Normal}(\mu(x), \sigma^2(x))$. The true model in this case consists of x values uniformly distributed from -5 to 5, and ys that are cubic in the xs with fixed standard deviation: $y \sim \mathsf{Normal}(x^3/100, 0.3^2)$. This true distribution is shown in orange. The 10 sampled (x, y) pairs the model was trained on are shown as the blue dots. The neural network quickly learns to set its predictive standard deviation to a small value and overfits to the samples.

By approximating the true distribution in eq. (3) with the average over the empirical samples, instead of concentrating on the *true* distribution $\nu(X,Y)$, we've had our network attempt to model the *empirical* distribution $\hat{\nu}_n(x,y) = \frac{1}{n}\sum_i^n \delta(X - x_i)\delta(Y - y_i)$. The neural network did well at the task it was asked to do, but not at the task we wanted.

The traditional way to improve the situation is to try to fit a Bayesian neural network. Instead of assuming that the parameters of the neural network take on particular values,

[1] Since much of the paper is concerned with the differences between taking expectations with respect to the true distribution versus the empirical distribution, we're using a blue E to denote expectations with respect to the true data distribution and red E to denote expectations with respect to the empirical distribution to increase the visibility of this distinction.



(a) Maximum Likelihood

(b) Bayesian neural network (HMC)

(c) ML with Multiple Target Samples

Figure 1: A simple demonstration of (a) a neural network overfitting, (b) its corresponding Bayesian neural network doing much better, and (c) the neural network trained with multiple target samples also doing very well. The true distribution is shown in orange, the network's predictive distribution in blue. All three models were fit using the same 10 $x \in X$ samples shown as the blue dots. The first two models were trained with a single y for each $x \in X$. ML with Multiple Target Samples is trained with many Y samples for each $x \in \mathcal{X}$.

we treat the parameters θ as a random variable themselves and compute the posterior over the parameters. This noticeably improves the predictive distribution, as can be seen in fig. 1(b), but this improvement comes at great computational cost. In this example, doing Hamiltonian Monte Carlo to generate samples from the posterior for the 2178 parameters of the neural network took 8 times longer than training the maximum likelihood model.

3. Multiple Target Samples

In the typical setup of a discriminative task, we have a finite sample of pairs (x, y) from the true joint distribution $\nu(X, Y)$. This amounts to a single sample for the target for each input, a single draw from each of the conditional distributions $\nu(Y|x)$. Hypothetically, what would happen if we kept the same 10 X samples we used above in figs. 1(a) and 1(b) but collected many Y samples for each? In fig. 1(c) we show the result of training precisely the same neural network as in fig. 1(a) in this new setup.

With access to many Y samples, the network learns to match the true predictive distribution nearly exactly at those sampled points. At the same time, the neural network does a reasonable job of interpolating between the sampled points while maintaining a good degree of predictive uncertainty. When asked to extrapolate outside of the data, the quality of the predictive distribution diminishes noticeably. Overall, asking the neural network to match a *half-empirical* distribution $\nu(Y|X)\hat{\nu}_n(X) = \nu(Y|X)\frac{1}{n}\sum_i^n \delta(X - x_i)$ has produced reasonably good results. This multiple target setup is similar to problem settings that use *soft targets*, such as teacher-student learning setups (Hinton et al., 2014), which have proven effective. Similarly, for image classification tasks, using multiple label samples can lead to improved fits (Peterson et al., 2019).

Is there some way to provide the sorts of guarantees Bayesian inference provides, but only with respect to the finite number of Y samples for each X? Is there some kind of compromise position we could adopt that achieves performance similar to that in fig. 1(c) without requiring actually collecting additional target samples for each input?

4. VIB as PAC-Bayes

One way to view the source of the Bayesian guarantees is that Bayesian inference optimizes a PAC style upper bound on the *true risk* (Masegosa, 2020; Morningstar et al., 2020). By penalizing the posterior from being too distinct from the prior, Bayesian inference *probably* won't overfit (PAC stands for *Probably* Approximately Correct, and the bound that Bayesian inference optimizes holds with high probability even with a finite training sample).

We can invoke the same PAC-Bayes bound as in the Bayesian case, but only on the inner expectation over *targets* demonstrating that with probability at least $1 - \xi_X$ (see appendix A for proofs):

$$\mathcal{C}(\theta) \leq \mathsf{E}_{\nu(X)} \Bigg[\mathsf{E}_{q(Z|X,\theta)} \left[\mathsf{E}_{\hat{\nu}_n(Y|X)}[-\log p(Y|Z)]\right] + \tau\, \mathsf{KL}\left[q(Z|X,\theta), r(Z)\right] \\ + \tau \log \mathsf{E}_{r(Z)}\, \mathsf{E}_{\nu(Y|X)} \left[e^{\frac{1}{\tau}\left(\left(\mathsf{E}_{\nu(Y|X)}[-\log p(Y|Z)]\right) - \left(\mathsf{E}_{\hat{\nu}_n(Y|X)}[-\log p(Y|Z)]\right)\right)} \right] - \log \xi_X \Bigg]. \quad (5)$$

While this is a lot to unpack, notice that all of the terms on the second line are constants with respect to the neural network's representation $q(Z|X, \theta)$, and so can be dropped in its objective. Equation (5) (nearly always) provides an upper bound on the *true classification risk*, However, it is still intractable as it includes an expectation over $\nu(X)$.

With this observation, we could instead adopt a mixed approach. Why not take the Bayesian strategy of minimizing an upper bound with respect to the conditional expectation of targets $\nu(Y|X)$ while using the bold Maximum Likelihood strategy of a drop-in Monte Carlo estimate for the expectation over inputs X? Doing so gives us:

$$\mathsf{VIB}(\theta) \stackrel{\text{def}}{=} \mathsf{E}_{\hat{\nu}_n(X)}\, \mathsf{E}_{q(Z|X,\theta)} \left[\mathsf{E}_{\hat{\nu}_n(Y|X)} \left[-\log p(Y|Z)\right] + \tau \log \frac{q(Z|X,\theta)}{r(Z)} \right]. \quad (6)$$

This objective is equivalent to the Variational Information Bottleneck (VIB) objective of Alemi et al. (2016). The VIB objective was originally motivated as being a variatonal upper bound on the Information Bottleneck objective (Tishby et al., 1999):

$$\max I(Z;Y) - \tau I(Z;X). \quad (7)$$

$I(X;Y) \stackrel{\text{def}}{=} \mathsf{E}_{p(X,Y)} \left[\log \frac{p(X,Y)}{p(X)p(Y)}\right]$ is the *mutual information* between X and Y. The Information Bottleneck aims to find a representation Z that is as maximally informative about the target Y as possible ($I(Z;Y)$), subject to a constraint on how *expensive* that representation is, measured by how many bits about the input it retains ($I(Z;X)$).

Here we have stumbled upon an alternative motivation of the same objective, showing that the VIB objective can be seen as *half Bayesian*. VIB attempts to protect against overfitting on a finite number of sampled targets for each input without addressing potentially overfitting to the finite number of sampled inputs themselves. It tries to concentrate on the half-empirical distribution of fig. 1(c). The VIB objective does not itself provide any bound on the true classification risk, just as Maximum Likelihood does not. Yet, VIB style objectives have been shown to improve model's generalization and robustness (Fischer and Alemi, 2020).

Where building a traditional Bayesian neural network requires a distribution over all of the parameters of the network, solving eq. (6) only requires a distribution over the *output activations* of the network. This is a much lower dimensional space and much easier to deal with computationally. In the VIB setup, the output of the neural network is made an explicit distribution on the representation space, often chosen to be a Gaussian distribution for simplicity.

Notice that in this interpretation, we are not allowed to learn either the *classifier* distribution $p(Y|Z)$ or the *prior* or *marginal* $r(Z)$ using eq. (5), as both of those distributions appear in the second line but are dropped in the objective (eq. (6)). In this way this half-Bayesian interpretation differs from the existing VIB literature, where both $p(Y|Z)$ and $r(Z)$ are thought to be variational approximations that are free to be fit simultaneously with the representation $q(Z|X, \theta)$. If the data were split, or there were additional holdout data, these could be used to refine either $p(Y|Z)$ or $r(Z)$ similar to the setup in Dziugaite et al. (2020).

If we wanted to generate a fully valid bound on the true classification risk, we could continue the road we are on and simple apply another PAC-Bound to eq. (5), this time with respect to the *parameters* of the encoding distribution $q(Z|X, \theta)$. See appendix A for the full details, but dropping the constant terms with regards to the objective we obtain a fully *Bayesian variational information bottleneck*:

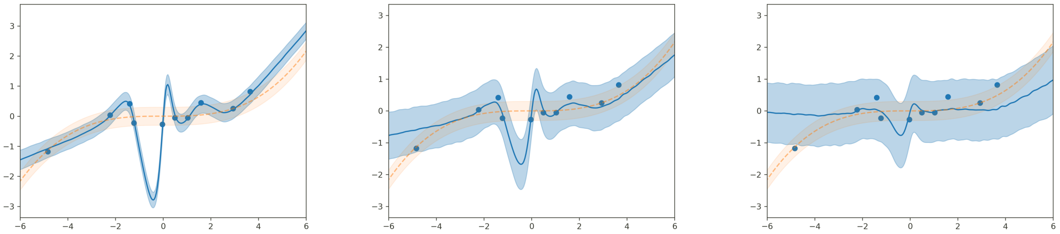
$$\mathsf{BVIB}[q(\Theta)] \overset{\text{def}}{=} \mathsf{E}_{\hat{\nu}_n(X)}\, \mathsf{E}_{q(\Theta)}\, \mathsf{E}_{q(Z|X,\Theta)} \left[\mathsf{E}_{\hat{\nu}_n(Y|X)} \left[-\log p(Y|Z) \right] + \tau \log \frac{q(Z|X,\Theta)}{r(Z)} + \frac{\gamma}{n} \log \frac{q(\Theta)}{r(\Theta)} \right] \tag{8}$$

Realizing eq. (8) could be as simple as adding weight decay to the parameters of the representation in eq. (6). Objectives of this sort also appeared in Alemi and Fischer (2018), where again they were motivated from an alternative, information theoretical perspective.

5. Demonstration

To illustrate that this can work, in fig. 2 we show the result of fitting the VIB objective (eq. (6)) to the same 10 data points as in figs. 1 and 1(c), using the same random network initialization. The results are sensitive to the choice of τ, so we show several values near the best performing models. Full experimental details can be found in appendix B. Figure 2(b) in particular has a similar predictive distribution to fig. 1(c), while only having access to a single target sample for each of the 10 input samples shown.

This qualitative sense that the VIB methods are doing well can be verified quantitatively. In table 1 we show the computed KL divergences between the true conditional distribution $\nu(Y|X)$ and the predictive distributions $q(Y|X, \theta) = \int \mathrm{d}z\, q(z|X, \theta) p(Y|z)$ for each method. This conditional KL can then be computed in expectation both with respect to the empirical X distribution $\hat{\nu}(X)$ (simply the average on the 10 samples), or in expectation with respect to the true $\nu(X)$, marginalizing from $x = -5$ to $x = 5$ uniformly. This assesses how well



(a) VIB $\tau = 10^3$ (b) VIB $\tau = 10^4$ (c) VIB $\tau = 10^5$

Figure 2: A simple demonstration that VIB can learn to capture uncertainty in much the same way that we could if we trained with multiple target samples as in fig. 1(c). The three figures show different values for τ. All three models have the same failure as seen in fig. 1(a) and fig. 1(c) because they were all initialized with the same random seed. This highlights the inherent risk of training with the empirical sample – the model can make arbitrary errors away from the observed data.

the methods did at learning the predictive distribution both on the values they were given ($\overline{\mathsf{KL}}$) as well as on all values (KL). The VIB approaches are competitive with the fully Bayesian model, while being significantly cheaper to optimize. The VIB models did not take noticeably longer to train than the ML model. We give additional experimental results on MNIST *classification* in appendix C.

	Determ	MultiY	Bayes	VIB 10^3	VIB 10^4	VIB 10^5
KL	3850	0.0993	0.195	1130	1.08	1.38
$\overline{\mathsf{KL}}$	1090	4.39×10^{-4}	0.330	1.85	0.763	1.22

Table 1: True and Empirical KL divergences for the predictive distribution from each method on the toy problem. All KLs are measured in bits. The large value for $\overline{\mathsf{KL}}$ for Determ is due to the fact that we know the true σ^2 for ν – even interpolating the sampled points doesn't protect against a large empirical risk.

6. Conclusion

We've demonstrated that on a simple problem we can provide most of the benefits of Bayesian inference for signficantly less work. The Variational Information Bottleneck method of Alemi et al. (2016) can be thought of as a half-Bayesian approach that offers some assurance that it won't too severely overfit, but only with regards to the finite sampling of the targets in a discriminative modeling task.

References

Alexander A. Alemi and Ian Fischer. TherML: Thermodynamics of machine learning, 2018.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck, 2016.

Arindam Banerjee. On bayesian bounds. In *Proceedings of the 23rd international conference on Machine learning*, pages 81–88, 2006.

Michael Betancourt. A conceptual introduction to hamiltonian monte carlo, 2018.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Joshua V Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A Saurous. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

Gintare Karolina Dziugaite, Kyle Hsu, Waseem Gharbieh, Gabriel Arpino, and Daniel M. Roy. On the role of data in pac-bayes bounds, 2020.

Ian Fischer and Alexander A. Alemi. CEB improves model robustness. *Entropy*, 22(10): 1081, Sep 2020. ISSN 1099-4300. doi: 10.3390/e22101081. URL http://dx.doi.org/10.3390/e22101081.

Pascal Germain, Francis Bach, Alexandre Lacoste, and Simon Lacoste-Julien. Pac-bayesian theory meets bayesian inference. In *Advances in Neural Information Processing Systems*, pages 1884–1892, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Dark knowledge. *Presented as the keynote in BayLearn*, 2, 2014.

Junpeng Lao, Christopher Suter, Ian Langmore, Cyril Chimisov, Ashish Saxena, Pavel Sountsov, Dave Moore, Rif A Saurous, Matthew D Hoffman, and Joshua V Dillon. tfp.mcmc: Modern markov chain monte carlo tools built for modern hardware. *arXiv preprint arXiv:2002.01184*, 2020.

Andres Masegosa. Learning under model misspecification: Applications to variational and ensemble methods. *Advances in Neural Information Processing Systems*, 33, 2020.

Warren R. Morningstar, Alexander A. Alemi, and Joshua V. Dillon. PAC^m-Bayes: Narrowing the empirical risk gap in the misspecified bayesian regime, 2020.

Joshua C. Peterson, Ruairidh M. Battleday, Thomas L. Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust, 2019.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proc. Allerton Conf. on Communication, Control, and Computing*, pages 368–377, Monticello, IL, September 1999.

Cheng Zhang, Judith Butepage, Hedvig Kjellstrom, and Stephan Mandt. Advances in variational inference, 2018.

Tong Zhang. Information-theoretic upper and lower bounds for statistical estimation. *IEEE Transactions on Information Theory*, 52(4):1307–1321, 2006.

Appendix A. Theory

In this appendix we prove the claims in the paper.

Suppose $(X,Y)^n \overset{\text{iid}}{\sim} \nu(Y|X)\nu(X)$; write their empirical distributions as,

$$\hat{\nu}_n(Y,X) = \frac{1}{n}\sum_i^n \delta(y_i - Y)\delta(x_i - X) \tag{9}$$

$$\hat{\nu}_n(X) = \int_{\mathcal{Y}} \mathrm{d}\mu(y)\, \hat{\nu}_n(y,X) = \frac{1}{n}\sum_i^n \delta(x_i - X) \tag{10}$$

$$\hat{\nu}_n(Y|X) = \frac{\hat{\nu}_n(Y,X)}{\hat{\nu}_n(X)} = \frac{\sum_i^n \delta(y_i - Y)\delta(x_i - X)}{\sum_i^n \delta(x_i - X)}. \tag{11}$$

For notational simplicity, we regard $\hat{\nu}_n(Y|X)$ as 0 (for all y) for $X \notin \{x_i\}_i^n$. Depending on its context, the symbol δ denotes either the Dirac or Kronecker delta function. (These two caveats are our only notational abuses in the paper.)

Theorem 1 *For all $q(\Theta)$ absolutely continuous with respect to $r(\Theta)$, $q(Z|X,\Theta)$ absolutely continuous with respect to $r(Z)$ for all $\{\Theta \in \mathcal{T} : q(\Theta) > 0\}$ and $\{X \in \mathcal{X} : \nu(X) > 0\}$, $(X,Y)^n \overset{iid}{\sim} \nu(Y|X)\nu(X)$, $\beta_X, \beta \in \mathbb{R}_+$, $n \in \mathbb{N}$, and $\xi_X, \xi \in (0,1]$, then with probability at least $1 - \max(\xi_X, \xi)$:*

$$\begin{aligned}
&- \mathsf{E}_{\nu(X)\nu(Y|X)} \log \mathsf{E}_{q(\Theta)q(Z|X,\Theta)}\left[p(Y|Z)\right] && (12)\\
&\quad\le -\frac{1}{n}\sum_i^n \mathsf{E}_{q(\Theta)q(Z|x_i,\Theta)}\left[\log p(y_i|Z)\right] && (13)\\
&\qquad + \frac{1}{\beta_X}\frac{1}{n}\sum_i^n \mathsf{E}_{q(\Theta)}\,\mathsf{KL}\left[q(Z|x_i,\Theta), r(Z)\right] && (14)\\
&\qquad + \frac{1}{\beta}\frac{1}{n}\,\mathsf{KL}\left[q(\Theta), r(\Theta)\right] && (15)\\
&\qquad + \mathsf{E}_{\nu(X)}\left[\psi_X\left(\nu(Y|X), r(Z), p(Y|Z), \beta_X, \xi_X\right)\right] && (16)\\
&\qquad + \psi\left(\nu(X)\nu(\hat{\nu}_n(Y|X)|X), r(\Theta)r(Z), q(Z|X,\Theta)p(Y|Z), n, \beta_X, \beta, \xi\right) && (17)
\end{aligned}$$

where,

$$\begin{aligned}\psi_X(\nu(Y|X),&r(Z),p(Y|Z),\beta_X,\xi_X) = \\ &= \frac{1}{\beta_X}\log \mathsf{E}_{r(Z)}\,\mathsf{E}_{\nu(Y|X)}\left[e^{\beta_X\Delta_X}\right] - \frac{\log\xi_X}{\beta_X} \\ \psi(\nu(X)\nu(\hat{\nu}_n(Y|X)),&\,r(\Theta)r(Z),q(Z|X,\Theta)p(Y|Z),n,\beta_X,\beta,\xi) = \\ &= \frac{1}{\beta n}\log \mathsf{E}_{r(\Theta)}\,\mathsf{E}_{\nu(X)}\,\mathsf{E}_{\nu(\hat{\nu}_n(Y|X))}\left[e^{\beta n\Delta}\right] - \frac{\log\xi}{\beta n}\end{aligned}$$

and where Δ_X, Δ are defined by Equations 21 and 22 and $\nu(\hat{\nu}_n(Y|X))$ is the true probability of the empirical conditional measure.

Note that neither ψ_X nor ψ are a function of $q(\Theta)$ and that quantities 13, 14, and 15 are not a function of the unknowable true data generating distribution, $\nu(Y|X)\nu(X)$.

Proof

First, with probability at least $1-\xi_X$ we have:

$$\begin{aligned}&-\mathsf{E}_{\nu(X)\nu(Y|X)}\log \mathsf{E}_{q(\Theta)}\,\mathsf{E}_{q(Z|X,\Theta)}\left[p(Y|Z)\right] && (18)\\ &\quad\leq -\mathsf{E}_{q(\Theta)}\,\mathsf{E}_{\nu(X)}\,\mathsf{E}_{q(Z|X,\Theta)}\,\mathsf{E}_{\nu(Y|X)}\log\left[p(Y|Z)\right] && (19)\\ &\quad\underset{\xi_x}{\lesssim} \mathsf{E}_{q(\Theta)}\,\mathsf{E}_{\nu(X)}\Bigg[-\mathsf{E}_{q(Z|X,\Theta)}\,\mathsf{E}_{\hat{\nu}_n(Y|X)}\log\left[p(Y|Z)\right] \\ &\qquad\qquad + \frac{1}{\beta_X}\mathsf{KL}\left[q(Z|X,\Theta),r(Z)\right] \\ &\qquad\qquad + \psi_X\left(\nu(Y|X),\beta_X,r(Z),\xi_X\right)\Bigg] && (20)\end{aligned}$$

Inequality 19 follows from Jensen's inequality and inequality 20 holds with probability at least $1-\xi_X$ and follows from applying Lemmas 2 and 3 to:

$$\Delta_X = \mathsf{E}_{\nu(Y|X)}\left[-\log p(Y|Z)\right] - \mathsf{E}_{\hat{\nu}_n(Y|X)}\left[-\log p(Y|X)\right]. \qquad (21)$$

Again applying Lemmas 2 and 3 to:

$$\begin{aligned}\Delta = \mathsf{E}_{\nu(X)\hat{\nu}_n(Y|X)}&\left[-\mathsf{E}_{q(Z|X,\Theta)}\log\left[p(Y|Z)\right] + \frac{1}{\beta_X}\mathsf{KL}\left[q(Z|X,\Theta),r(Z)\right]\right] - \\ \mathsf{E}_{\hat{\nu}_n(X)\hat{\nu}_n(Y|X)}&\left[-\mathsf{E}_{q(Z|X,\Theta)}\log\left[p(Y|Z)\right] + \frac{1}{\beta_X}\mathsf{KL}\left[q(Z|X,\Theta),r(Z)\right]\right]. \qquad (22)\end{aligned}$$

we conclude that with probability at least $1-\max(\xi_X,\xi)$*:*

$$\begin{aligned}
&[\text{Equation } 20] \\
&\underset{\xi_X,\xi}{\lesssim} -\mathsf{E}_{q(\Theta)}\,\mathsf{E}_{\hat{\nu}_n(X,Y)}\,\mathsf{E}_{q(Z|X,\Theta)}\log\left[p(Y|Z)\right] \\
&\qquad + \frac{1}{\beta_X}\,\mathsf{E}_{q(\Theta)}\,\mathsf{E}_{\hat{\nu}_n(X)}\left[\mathsf{KL}\left[q(Z|X,\Theta),r(Z)\right]\right] \\
&\qquad + \frac{1}{\beta n}\,\mathsf{KL}\left[q(\Theta),r(\Theta)\right] \\
&\qquad + \mathsf{E}_{\nu(X)}\left[\psi_X\left(\nu(Y|X),r(Z),p(Y|Z),\beta_X,\xi_X\right)\right] \\
&\qquad + \psi\left(\nu(X)\nu(\hat{\nu}_n(Y|X)),r(\Theta)r(Z),q(Z|X,\Theta)p(Y|Z),n,\beta_X,\beta,\xi\right).
\end{aligned}$$

The proof is completed by expanding occurrences of $\mathsf{E}_{\hat{\nu}_n}$ *as a summation.* ■

The proof of Theorem 1 is similar to twice applying the technique of Morningstar et al. (2020) (with $m=1$). The Morningstar et al. (2020) proof followed arguments similar to Masegosa (2020) which itself followed arguments similar to Germain et al. (2016).

Note that in the text body we used $\tau=\frac{1}{\beta_X}$ and $\gamma=\frac{1}{\beta}$.

We also note that one can use Lemma 5 to rewrite $\mathsf{E}_{\hat{\nu}_n}$ expectations as conditional averages (e.g., this could be done to eq. (5)).

A.1. Lemmas

In this section we present several Lemmas used to simplify this paper's proofs. The Lemmas are well-known and are given here for the reader's convenience.

Lemma 2 (Compression) *If* $p(\Theta)$ *is absolutely semicontinuous wrt* $r(\Theta)$ *and* $\mathsf{E}_{r(\Theta)}[e^{f(\Theta)}]<\infty$*, then* $\mathsf{E}_{p(\Theta)}[f(\Theta)]\le\mathsf{KL}\left[p(\Theta),r(\Theta)\right]+\log\mathsf{E}_{r(\Theta)}[e^{f(\Theta)}]$.

Proof *Write* $q(\Theta)\overset{def}{=}\frac{r(\Theta)e^{f(\Theta)}}{\mathsf{E}_{r(\Theta)}[e^{f(\Theta)}]}$ *and note that Lemma 4 implies,* $0\le\mathsf{KL}\left[p(\Theta),q(\Theta)\right]=\mathsf{KL}\left[p(\Theta),r(\Theta)\right]-\mathsf{E}_{p(\Theta)}[f(\Theta)]+\log\mathsf{E}_{r(\Theta)}[e^{f(\Theta)}]$. ■

Proof due to Banerjee (2006); Zhang (2006).

Lemma 3 (Log Markov Inequality) *For any* $\xi\in(0,1]$ *and random variable* $Z\sim p$ *with* $p(Z\le 0)=0$ *then* $p(\log Z\le\log\mathsf{E}_p[Z]-\log\xi)\ge 1-\xi$.

Proof *Markov's inequality states that* $p(Z>t)\le\frac{\mathsf{E}_p[Z]}{t}$ *for non-negative random variable* $Z\sim p$ *and* $t>0$*. Substituting* $t=\frac{\mathsf{E}_p[Z]}{\xi}$ *implies* $p(Z>\frac{\mathsf{E}_p[Z]}{\xi})\le\xi$*. Combining this with the fact that* $\log$ *is a non-decreasing bijection implies* $p(\log Z>\log\mathsf{E}_p[Z]-\log\xi)\le\xi$*. Examining the complement interval completes the proof.* ■

Lemma 4 (Gibb's Inequality) *If* $p(\Theta)$ *is absolutely semicontinuous wrt* $r(\Theta)$*, then* $\mathsf{KL}[p,q]\ge 0$.

Proof $\mathsf{KL}[p,q]=-\mathsf{E}_{p(x)}\left[\log\frac{q(x)}{p(x)}\right]\ge-\log\mathsf{E}_{p(x)}\left[\frac{q(x)}{p(x)}\right]=-\log 1=0$ *where the inequality is Jensen's.* ■

Lemma 5 (Conditional Empirical Expectation) *Assuming $X \in \{x_i\}_i^n$, then:*

$$\mathsf{E}_{\hat{\nu}_n(Y|X)}[f(Y,X)] = \frac{1}{\sum_i^n \delta(x_i - X)} \sum_i^n \delta(x_i - X) f(y_i, x_i). \tag{23}$$

Proof $\mathsf{E}_{\hat{\nu}_n(Y|X)}[f(Y,X)] = \int_{\mathcal{Y}} d\mu(y)\, \hat{\nu}_n(y|X) f(y,X) = \int_{\mathcal{Y}} d\mu(y) \sum_i^n \frac{\delta(y_i - y)\delta(x_i - X)}{\sum_j^n \delta(x_j - X)} f(y,X) = \sum_i^n \delta(x_i - X) \frac{\int_{\mathcal{Y}} d\mu(y)\, \delta(y_i - y) f(y,X)}{\sum_j^n \delta(x_j - X)} = \frac{1}{\sum_i^n \delta(x_i - X)} \sum_i^n \delta(x_i - X) f(y_i, x_i)$. *The δ function is either the Dirac or Kronecker delta function, depending on whether measure μ is continuous or discrete.* ■

Appendix B. Experimental Details

For figs. 1(*a*) to 1(*c*) and 2(*a*) to 2(*c*) all experiments were done with JAX (Bradbury et al., 2018).

The true data distribution was taken to be

$$X \sim \mathsf{Uniform}(-5, 5)$$
$$Y|X \sim \mathsf{Normal}\left(\frac{x^3}{100}, 0.3^2\right).$$

Ten samples were taken for the data distribution. The predictive network consisted of two fully connected layers with 32 hidden units followed by an elu activation. The final layer was a linear layer with 2 outputs, the first of which was taken as the mean, and the second generated the standard deviation of the predictive model with a softplus activation and a minimum value of 0.01: ($\sigma^2 = 0.01 + \mathsf{softplus}(x)$).

A standard Lecun style truncated normal initialization scheme was used for the kernels, and the biases were initialized to be zero. The initial parameter variance was increased by a factor of 5, which was found to be important to get the one dimensional networks to converge well on the range and domain of the toy problem. All problems used the same initial parameters and the same adabelief optimizer with a cosine decay schedule on the learning rate starting at 10^{-3} and ending at 0 after 100k steps, the length of each optimization run.

To sample from the Bayesian neural network, tensorflow_probability's JAX backend Hamiltonian Monte Carlo sampler was used (Dillon et al., 2017; Lao et al., 2020). In particular 1000 results were generated from the chain with 10k burn-in steps, dual averaging step size adaptation with a step size of 10^{-3} and 100 leapfrog steps, 9000 adaptation steps and a target acceptance probability of 0.7. The initialization distribution used for the neural network experiments was taken to be the prior distribution, both for the kernel and bias parameters.

For the VIB experiments, the same neural network as above was used to form the representation $q(Z|X)$. The classifier network $p(Y|Z)$ was taken to be a Normal distribution with a small fixed variance: $\mathsf{Normal}(z, 0.01^2)$. The marginal $r(Z)$ was set to be a fixed unit Normal: $\mathsf{Normal}(0, 1)$.

All KL divergences were computed using 100k samples of data points and 10k samples from the any intermediate distributions as required.

Appendix C. MNIST Experiments

In the main paper, we demonstrate that VIB learns a model which retains uncertainty about the targets Y, even if it provides no guarantees about generalization in X. For an additional illustrative comparison, we train models on the MNIST dataset. Specifically, we compare the predictive models of a deterministic deep neural network and a VIB model having the same architecture. We use a parameter free decoder, with a categorical likelihood, multivariate normal prior, and multivariate normal posterior. For the prior, we assume zero mean and identity covariance, while we predict the full covariance matrix for the posterior. The deterministic model replicates this setup, but simply predicts the categorical likelihood rather than the posterior, and it has no prior. We train models for 50 epochs with a learning rate of 0.001, which is decayed by half every 5000 steps. We further use a batch size of 128. For the VIB model, we evaluate the objective using 4 samples from the posterior and use $\tau = 0.005$ as a weighting for the KL penalty.

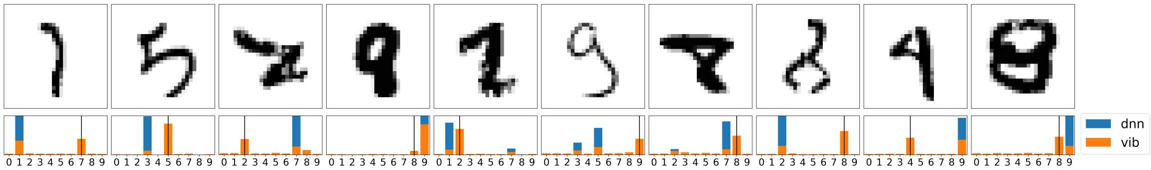


Figure 3: A selection of images from the MNIST dataset which are misclassified, either by a deterministic model or by a VIB model. The lower panel shows the class probabilities predicted by the model for a VIB and a deterministic model. We find that the deterministic model tends to assign high probability to a single class, while a VIB model tends to incorporate uncertainty between multiple classes.

We evaluate models using the classification accuracy as well as the log-likelihood of the test set. To compute the log-likelihood of VIB models, we marginalize over 1000 samples from the posterior. Both models produce similar final test set accuracies (99.2% for the deterministic model, and 99.4% for the VIB model). We further find that the test set log-likelihood for the VIB model is higher (-508 versus -786 for the deterministic model). However, this later finding is heavily dependent on the τ used in training: We find that the VIB model can measure a lower log-likelihood if the τ multiplier is larger. Note that the accuracy is more robust to this hyperparameter, and we find that the VIB model consistently observes higher accuracy than the deterministic model over the range of τ we explored.

In addition to showing that VIB leads to models which have a higher test log-likelihood and accuracy, we also examined if the resulting predictive models learned by VIB incorporate more uncertainty into the labels than do those which are trained with a deterministic network. For this, in fig. 3 we show 10 images from the test set, each of which is classified incorrectly by either the VIB model or by the deterministic model. Below each image, we show the class probabilities predicted by the deterministic and VIB models, and indicate the true class with a vertical line. Our main observation is that many of the deterministic models overpredict the probability of a label, with 7 out of 10 images being assigned a class probability greater than 95%. We also find that in these same situations, VIB often folds additional probability into other classes one of which is typically the correct class. This, when combined with the higher test set log-likelihood agrees with our findings that VIB facilitate generalization over the label distribution.