PAC^m-Bayes

Narrowing the Empirical Risk Gap in the Misspecified Bayesian Regime

Warren R. Morningstar
Google Research
wmorning@google.com

Alexander A. Alemi
Google Research
alemi@google.com

Joshua V. Dillon
Google Research
jvdillon@google.com

Abstract

The Bayesian posterior minimizes the "inferential risk" which itself bounds the "predictive risk." This bound is tight when the likelihood and prior are well-specified. However since misspecification induces a gap, the Bayesian posterior predictive distribution may have poor generalization performance. This work develops a multi-sample loss (PAC^m) which can close the gap by spanning a trade-off between the two risks. The loss is computationally favorable and offers PAC generalization guarantees. Empirical study demonstrates improvement to the predictive distribution.

1 Introduction

The top and bottom of fig. 1 differ by one line of code. The traditionally inferred (approximate) posterior (top) fails to capture heteroskedastic noise in the data while the proposed generalization (bottom) succeeds. Both rows employ the same data, computation, model family, and optimization procedure; only the loss differs in that the first row is based on a average-log-likelihood and the second is based on a log-average-likelihood. Before we can understand how/why this works and return to this example (section 7 and fig. 3), we have to examine the difference between prediction and inference.

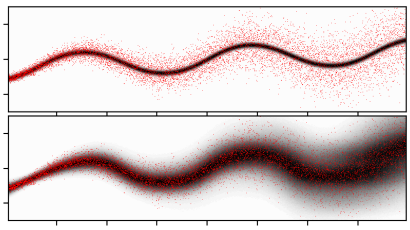

Figure 1: A misspecified Bayesian neural network changes from bad uncertainty estimation in the posterior predictive distribution (top row), to good uncertainty estimation in the (approximate) posterior predictive distribution (bottom row) with a simple (one line) change to the training loss. Cutout from fig. 3.

Pierre-Simon Laplace formulated one of the earliest Bayesian models [Laplace, 1781]. Interested in the relative birth rates of boys and girls, he derived the Beta posterior for a Bernoulli likelihood with uniform prior. He then calculated the "posterior probability" that the girl birth rate exceeds the boy rate and found it to be about 10^{-42} from which he concluded that it is "as certain as any other moral truth" that humans give birth to more boys than girls.[1]

Laplace's objective was to *infer* the parameter of his model. Broadly, science has followed suit. When a modern experiment such as the Large Hadron Collider at CERN processes terabytes of particle collision data [ATLAS Collaboration, 2012], or the Planck satellite maps the cosmic microwave background radiation [Planck Collaboration VI, 2019], they are in pursuit of the "moral truth" of some underlying parameter of the universe.

Contrast this with modern machine learning. The primary goal of machine learning is to build models that can form accurate *predictions*. We do not truly care about the value of the millionth weight in

[1] The presently accepted natural ratio is 105 males per 100 females [Ritchie, 2019]

Preprint. Under review.

a deep neural network. We do not believe the parameters of the neural network are reflecting any "moral truths".

For well-specified models the goals of inference and prediction align. For misspecified models they might not. Optimizing for inference when you'll evaluate a model's predictive performance violates the golden rule of ML: *do unto training as done unto testing*. As made clear in the work of Masegosa [2019], both Bayesian inference and Maximum Likelihood target inferential rather than predictive risks, and can make poor predictions under model misspecification.

This issue is well known [Minka, 2000, Domingos, 1997], and much prior work has been done to address it (e.g. [Berger et al., 1994, Yao et al., 2018, Bissiri et al., 2016, Grünwald et al., 2017, Jankowiak et al., 2020a,b, Masegosa, 2019, Wei et al., 2020, Sheth and Khardon, 2020]).

In this work, we introduce a tractable multi-sample bound on the true predictive risk which sometimes manifests in the striking improvement demonstrated in in figs. 1 and 3. This bound interchanges average and log and enables recovering ordinary Maximum Likelihood and the Bayesian posterior.

2 Predictive and Inferential Risks

We begin at a high level with a *statistical model*: $p(X|\theta)$ defining a distribution of each observed datum X in terms of some parameters θ. After observing n data points drawn from some *true distribution* $X^n \stackrel{\text{def}}{=} \{X_i\}_i^n \stackrel{\text{iid}}{\sim} \nu(X)$, we form a distribution of parameters, $q(\Theta|\{x_i\}_i^n)$. In principle we can then compute the *predictive distribution*:

$$p(X|\{x_i\}_i^n) = \mathbb{E}_{q(\Theta|\{x_i\}_i^n)}\left[p(X|\Theta)\right]. \tag{1}$$

(For brevity we henceforth regard q's dependence on $\{x_i\}_i^n$ as implicit.) If we had some particular application in mind, at this point we could score our model's ability to make predictions as measured by some specific *risk*, a path that would lead to the general field of Bayesian risk minimization [Berger, 1985]. To keep things simple here, lacking a specific risk, we judge the quality of our predictive distribution by measuring the relative entropy (Kullback-Leibler divergence) between the true distribution and our predicted one:

$$\mathsf{KL}[\nu(X); p(X|q)] = \mathbb{E}_{\nu(X)}\left[\log \frac{\nu(X)}{p(X|q)}\right] = \mathbb{E}_{\nu(X)}[\log \nu(X)] - \mathbb{E}_{\nu(X)}\left[\log p(X|q)\right]. \tag{2}$$

Up to a constant outside our control (the continuous entropy of the true distribution) this defines what we'll call the *true predictive risk*:

$$\mathcal{P}[q] \stackrel{\text{def}}{=} -\mathbb{E}_{\nu(X)}\left[\log \mathbb{E}_{q(\Theta)}[p(X|\Theta)]\right]. \tag{3}$$

In many cases the true predictive risk is ultimately what we care most about. Determining how accurately we can predict the future, it is often what governs how much money our model will make or how many lives it will save.

Not knowing the true distribution $\nu(X)$, we cannot directly minimize the true predictive risk. One thing we can compute is the *empirical predictive risk*:

$$\overline{\mathcal{P}}_n[q] \stackrel{\text{def}}{=} -\frac{1}{n}\sum_i^n \log \mathbb{E}_{q(\Theta)}[p(x_i|\Theta)]. \tag{4}$$

This is the observed average risk on the $\{x_i\}_i^n$ sample set. Akin to a training loss, the empirical predictive risk is a measure of how well we do at predicting the training data. If used as a target for optimization we can easily overfit. Training with this risk directly would amount to a type of *ensemble method* [Dietterich, 2000] or *non-parametric mixture* with mixing distribution $q(\Theta)$ [Wang, 2007, Lindsay, 1995].

In contrast to the predictive risks, we'll also define the *inferential risks* which focus on determining or inferring the correct values of the parameters. The *true inferential risk* (often just called the *true risk*):

$$\mathcal{R}[q] \stackrel{\text{def}}{=} -\mathbb{E}_{\nu(X)}\left[\mathbb{E}_{q(\Theta)}\left[\log p(X|\Theta)\right]\right], \tag{5}$$

and the corresponding *empirical inferential risk* (often called the *empirical risk*):

$$\overline{\mathcal{R}}_n[q] \stackrel{\text{def}}{=} -\frac{1}{n}\sum_i^n \mathbb{E}_{q(\Theta)}\left[\log p(x_i|\Theta)\right]. \tag{6}$$

For a variety of reasons, directly minimizing the inferential risk is fairly commonplace. It measures the average of the divergence between the true distribution $\nu(X)$ and the single-value parameter settings of the model $p(X|\theta)$. This is akin to doing variational optimization [Staines and Barber, 2012], and concentrates on a delta function corresponding to the best single-value parameter setting.

Jensen's inequality implies $-\log \mathbb{E}_{q(\Theta)}[p(x|\Theta)] \leq -\mathbb{E}_{q(\Theta)}[\log p(x|\Theta)]$ and so the inferential risks are upper bounds on the predictive risks:

$$\mathcal{P}[q] \leq \mathcal{R}[q] \qquad \text{and} \qquad \overline{\mathcal{P}}_n[q] \leq \overline{\mathcal{R}}_n[q]. \tag{7}$$

In this way, minimizing the inferential risk is a *valid* strategy for achieving good predictions since it minimizes an upper bound on the predictive risk. When is it a *good* strategy? When is this bound tight? Answer: If our model is well-specified [Masegosa, 2019] (Proof replicated in appendix C.1). However, in cases of model misspecification this can break down severely. We would prefer to target the true predictive risk directly, but cannot since we do not know the true data distribution. What we need is a tractable bound on the true risks.

3 PAC-Bayes

While the empirical risks $(\overline{\mathcal{P}}, \overline{\mathcal{R}})$ provide unbiased estimates of the true risks $(\mathcal{P}, \mathcal{R})$, minimizing the empirical risks do not minimize the true risks:

$$\arg\min_q \mathcal{R}[q] = \arg\min_q \mathbb{E}\left[\overline{\mathcal{R}}_n[q]\right] \neq \mathbb{E}\left[\arg\min_q \overline{\mathcal{R}}_n[q]\right].$$

Said another way, the empirical risks do not provide a *bound* on the true risks.

Despite not being a valid bound, empirical risk minimization is quite popular. Minimizing the empirical (inferential) risk over the space of all possible distributions over parameters is the well known Maximum Likelihood method. This concentrates in a delta-function-like parameter distribution with all of its mass on the maximum likelihood parameter value.

We could similarly directly optimize the empirical predictive risk, known to some as a *non-parametric mixture* [Lindsay, 1995, Wang, 2007]. In cases with bounded likelihoods this seems to perform decently well (e.g. the toy example of section 5) just as it does in the case of Maximum Likelihood. If our model is too expressive minimizing the empirical risks will quickly start to concentrate on the *empirical* data distribution rather than the *true* distribution, overfitting severely. Classic approaches prevent overfitting by limiting model capacity; by adding regularization or other tricks. If we instead had a valid bound on the true risks, we needn't worry. PAC-Bayes approaches provide such a bound.

We would really like to have some assurance that we won't overfit to our finite training data. We can formulate an upper bound on the true risks in terms of the empirical risks that nearly always hold. Such *probably approximately correct* (or PAC) bounds can be used to motivate Bayesian inference, demonstrating that the Bayesian posterior is the minimizer of a PAC-style upper bound on the true inferential risk $\mathcal{R}$ [Banerjee, 2006, Alquier et al., 2016, Guedj, 2019] (Proof replicated in appendix C.2).

In light of these results we will define the following PAC-*inferential risk* (or ELBO):

$$\widetilde{\mathcal{R}}_n[q; r, \beta] \stackrel{\text{def}}{=} \overline{\mathcal{R}}_n[q] + \frac{1}{\beta n}\mathsf{KL}\left[q(\Theta); r(\Theta)\right] = \mathbb{E}_{q(\Theta)}\left[-\frac{1}{n}\sum_i \log p(x_i|\Theta) + \frac{1}{\beta n}\log\frac{q(\Theta)}{r(\Theta)}\right]. \tag{8}$$

Aside from constants independent of q, $\widetilde{\mathcal{R}}$ is a stochastic upper bound on $\mathcal{R}$. Intuitively, this is accomplished by ensuring that our parameter distribution $q(\Theta)$ can't stray too far from a *prior* $r(\Theta)$ we chose before looking at the data. Notice that ordinary Bayesian inference corresponds to minimizing this risk for $\beta = 1$ [Knoblauch et al., 2019, Bissiri et al., 2016]. Furthermore, as $\beta \to \infty$ we recover the empirical risk $\overline{\mathcal{R}}$ and thus Maximum Likelihood.

Because $\mathcal{P} \leq \mathcal{R}$, Bayesian inference is equivalent to (almost always) minimizing an upper bound on the *true predictive risk* $\mathcal{P}$. In the case of a well-specified model, $\min \mathcal{P} = \min \mathcal{R}$ and Bayesian inference targets not only optimal inferential power but also optimal predictive power. If you have the correct model, searching for the correct single parameter setting of the model is the right thing to do.

What ought we do if our model is misspecified?

4 PAC^m-Bayes

If our model is misspecified, there may be a large gap between the minimum of the predictive and inferential risks ($\min \mathcal{P} \ll \min \mathcal{R}$) as we'll demonstrate in our experiments below.

Our central contribution is to provide a new class of bounds, analogous to the PAC-style upper bounds on the inferential risk but targeting the predictive risk more directly.

The potential gap between the predictive and inferential risks came from invoking Jensen's inequality:

$$-\log \mathbb{E}_{q(\Theta)}[p(x|\Theta)] \leq -\mathbb{E}_{q(\Theta)}[\log p(x|\Theta)]. \tag{9}$$

The core insight is to explore a family of multisample stochastic bounds: [Burda et al., 2015, Mnih and Rezende, 2016]

$$-\log \mathbb{E}_{q(\Theta^m)}[p(x|\Theta^m)] \leq -\mathbb{E}_{q(\Theta^m)}\left[\log \frac{1}{m}\sum_j^m p(x|\Theta_j)\right] \leq -\mathbb{E}_{q(\Theta)}[\log p(x|\Theta)]. \tag{10}$$

Averaging a finite number of samples from our parameter distribution provides an unbiased estimate of the predictive likelihood. Taking the log of an unbiased estimator produces a *stochastic lower bound* [Burda et al., 2014, Grosse et al., 2016] that becomes tight asymptotically.

Theorem 1. *For all $q(\Theta)$ absolutely continuous with respect to $r(\Theta)$, $X^n \overset{iid}{\sim} \nu(X)$, $\beta \in (0, \infty)$, $n, m \in \mathbb{N}$, $p(x|\theta) \in (0, \infty)$ for all $\{x \in \mathcal{X} : \nu(x) > 0\} \times \{\theta \in \mathcal{T} : r(\theta) > 0\}$, $\xi \in (0, 1)$, and $\lambda_m \in [m, \infty)$, then with probability at least $1 - \xi$:*

$$\mathcal{P}[q] \leq \widetilde{\mathcal{P}}_{n,m}[q; r, \beta] + \psi(\nu, n, m, \beta, r, \xi) \tag{11}$$

and furthermore (unconditionally):

$$\widetilde{\mathcal{P}}_{n,m}[q; r, \beta] \leq \widetilde{\mathcal{P}}_{n,m-1}[q; r, \beta] \leq \widetilde{\mathcal{P}}_{n,1}[q; r, \beta] = \widetilde{\mathcal{R}}_n[q, r, \beta] \tag{12}$$

where:

$$\widetilde{\mathcal{P}}_{n,m}[q; r, \beta] \overset{def}{=} -\frac{1}{n}\sum_i^n \mathbb{E}_{q(\Theta^m)}\left[\log\left(\frac{1}{m}\sum_j^m p(x_i|\Theta_j)\right)\right] + \frac{m}{\lambda_m}\mathsf{KL}\left[q(\Theta); r(\Theta)\right] \overset{def}{=} \mathsf{PAC}^m \tag{13}$$

$$\psi(\cdot) \overset{def}{=} \frac{1}{\lambda_m}\log \mathbb{E}_{\nu(X^n)}\,\mathbb{E}_{r(\Theta^m)}\left[e^{\lambda_m \Delta_{n,m}}\right] - \frac{1}{\lambda_m}\log \xi \tag{14}$$

$$\begin{aligned}\Delta_{n,m} &\overset{def}{=} \Delta(X^n, \Theta^m) \\ &\overset{def}{=} \frac{1}{n}\sum_i^n \log\left(\frac{1}{m}\sum_j^m p(X_i|\Theta_j)\right) - \mathbb{E}_{\nu(X)}\left[\log\left(\frac{1}{m}\sum_j^m p(X|\Theta_j)\right)\right]\end{aligned} \tag{15}$$

Proof. Proof in appendix C.3. *Sketch*: form a multisample bound on the predictive risk and apply the traditional PAC-Bayes bound. □

Our main result is in theorem 1 above. Despite looking rather complex, this PAC-Bound establishes that we are free to minimize the empirical predictive risk for any finite m, without fear of overfitting, provided we simultaneously ensure that our parameter distribution remains close to some *prior* $r(\Theta)$

which we specified independent of the data. This is nearly always an upper bound on the true risk, with an offset determined by ψ (eq. (14)), a term which measures the gap (Δ, eq. (15)) in our true and empirical inferential risks if we drew parameter values from our prior. Most crucially, this ψ is independent of $q(\Theta)$ and so for optimization purposes can be dropped as a constant. We further show (Theorem 3, Appendix C.3) that under certain assumptions, ψ is finite and therefore Theorem 1 is non-vacuous (has a $o(1)$ bounded excess) for fixed finite m. More specifically, if $\lambda = o(nm)$ then $\psi = o(m)$ and if $\lambda = o(n \log m)$ then $\psi = o(\log m)$. There are subtle issues that arise when we ask whether it is always better to increase m (for more detailed discussion see appendices B.2 and C.3).

This yields our proposed risk, $\widetilde{\mathcal{P}}_{n,m}$ (eq. (13)). Minimizing $\widetilde{\mathcal{P}}_{n,m}$ (eq. (13)) is equivalent to minimizing a stochastic upper bound on the true predictive risk $\mathcal{P}$, analogous to the relationship between $\widetilde{\mathcal{R}}$ and $\mathcal{R}$. See theorem 2 for a complete proof, though it follows directly from the traditional PAC-Bayes proof once we invoke the multisample bound. Furthermore, as we increase m, PAC^m decreases (eq. (12)).

Dropping ψ (being constant in q, though there is a lot of nuance here, see appendices B.2 and C.3), we can summarize the relationships between the risks as:

$$\mathcal{P} \lesssim \widetilde{\mathcal{P}}_{n,m} \leq \widetilde{\mathcal{P}}_{n,1} = \widetilde{\mathcal{R}}_n \gtrsim \mathcal{R} \geq \mathcal{P}. \tag{16}$$

The $\widetilde{\mathcal{R}}_n \gtrsim \mathcal{R}$ relationship is the classic PAC-Bayes result [Alquier et al., 2016], $\mathcal{R} \geq \mathcal{P}$ follows from Jensen's inequality [Masegosa, 2019], and the left hand side $\mathcal{P} \lesssim \widetilde{\mathcal{P}}_{n,m} \leq \widetilde{\mathcal{P}}_{n,1} = \widetilde{\mathcal{R}}_n$ is our contribution.

Masegosa [2019] identified the need for tighter bounds on predictive risks than $\mathcal{R}$, suggesting a family (PAC^2_T) of risks that utilize a second order Jensen bound. While estimating the variance term in PAC^2_T requires care, PAC^m is minibatch friendly, having its expectation over the parameter distribution outermost in the objective. Later in our experiments we directly compare these approaches.

We now have two knobs we can use to adjust our risk: m, the number of samples we use to estimate the predictive distribution and β, a sort of inverse temperature used for adjusting the relative strength of the likelihood and prior terms. For $m = 1$ we recover the (inferential) risks we are used to, but for $m \geq 1$ we may form tighter bounds on the true predictive risk. With $\beta = 1$ we recover traditional Bayesian inference with an equal weighting of the likelihood and prior terms, as $\beta \to \infty$ we recover purely empirical risks. For any β, including $\beta \leq 1$ we still maintain our stochastic bounds. Downweighting the KL term with respect to the prior, or *cold posteriors* has shown to be useful especially in the context of neural networks [Wenzel et al., 2020].

What do realizations of PAC^m look like in practice? In practice it amounts to a very simple change to existing variational Bayesian approaches. As can be seen in figs. 11 and 12, this can be a one line change, changing an expectation over draws from the variational posterior with a $\mathsf{logsumexp}$. Instead of scoring the average log likelihood of m draws from a variational posterior, we instead score the log of the average likelihood across m draws from a variational posterior.

5 An Illustrative Toy Example

Consider trying to fit a Normal distribution to a set of observations with a fixed unit variance but unknown mean:

$$p(x|\theta) = \mathsf{Normal}(x; \theta, 1) = (2\pi)^{-\frac{1}{2}} e^{-\frac{(x-\theta)^2}{2}}. \tag{17}$$

Imagine further that we are operating in a severe model misspecification regime. While our model is a unit variance Normal distribution, the true data distribution is a 30-70 mixture of two Normals with twice the standard deviation and separated by four times their standard deviation.

In fig. 2 (left) we show the predictive distributions that result from minimizing all of the risks discussed previously. In fig. 2 (right) we show the corresponding parameter distributions. The true data distribution is shown with the dark red curve in fig. 2 (left). The dark red tick marks on the axis show five ($n = 5$) samples which we took as our data. The fig. 2 caption lists the resulting KL divergences between the true data distribution and each of the found predictive distributions.

Minimizing $\overline{\mathcal{R}}$ (eq. (6)) is equivalent to Maximum Likelihood (grey curves), which concentrates its parameter distribution to a delta-function located at the empirical mean, and whose predictive distribution is simply a unit variance Normal distribution centered at that empirical mean.

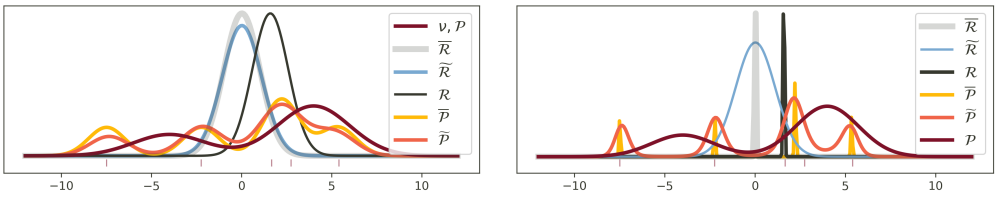

Figure 2: Toy Example: The left plot shows the resulting predictive distributions. The right plot shows the learned parameter distributions. Please see accompanying text for a full explanation. Measured in bits, the KL divergences between the true distribution and each method are: $\overline{\mathcal{R}}$: 12., $\widetilde{\mathcal{R}}$: 9.6, $\mathcal{R}$: 10.0, $\overline{\mathcal{P}}$: 0.5, $\widetilde{\mathcal{P}}$: 0.38, $\mathcal{P}$: 0.0

Minimizing $\widetilde{\mathcal{R}}$ (eq. (8), aka ELBO) is equivalent to Bayesian inference (blue curves). Here, we used a weakly informative $\mathsf{Normal}(0, 9^2)$ prior. This risk prevents the parameter distribution from collapsing onto a delta function, but the resulting predictive distribution is quite similar to the one we found with Maximum Likelihood. It is still fundamentally unimodal as minimizing $\widetilde{\mathcal{R}}$ is still fundamentally looking for the best single parameter setting of our model.

Minimizing $\mathcal{R}$ (eq. (5)) is equivalent to Bayesian inference with infinite data (black curve). Here again our parameter distribution concentrates on a delta function, this time at the true distributions mean, but the resulting predictive distribution is the best single parameter setting we could achieve, a unimodal predictive distribution that doesn't match the true distribution all that well. Notably, in this case it gives worse predictions than $\widetilde{\mathcal{R}}$.

In contrast, minimizing the predictive risks (warm colors) do not look for single parameter settings of the model. Minimizing $\overline{\mathcal{P}}$ (eq. (4)) performs a sort of clustering of the data (yellow curve). While it might seem natural to allow each data point its own delta-like contribution in the parameter distribution, two of our samples are near enough that we achieve better empirical predictive risk by combining the two points into a single contribution to the parameter distribution with twice the weight but located at the two points' mean. The resulting predictive distribution remains multimodal and achieves a much lower divergence with respect to the true distribution (0.5 bits versus the ~ 10 bits for the traditional (inferential) risks).

Minimizing $\widetilde{\mathcal{P}}$ (eq. (13), aka PAC^m, here with $m \to \infty$ see appendix E.2) has a similar qualitative effect compared to the corresponding inferential case ($\widetilde{\mathcal{R}}$). The addition of the KL penalty with respect to some prior (here the same as used in the Bayesian case) prevents the parameter distribution from collapsing to a delta-comb.

Finally, in this case, even though we have rather gross model misspecification in the sense that our model $p(X|\Theta)$ is quite unlike the true distribution for any single value of Θ, our true distribution can be expressed as an infinite mixture of our model. Minimizing the true $\mathcal{P}$ (eq. (3)) can achieve perfect predictive performance (red curve). This is achieved with a bimodal Normal distribution in parameter space which when convolved with our Normal model gives the exact bimodal Normal data distribution we chose. This is also what we achieve asymptotically from $\widetilde{\mathcal{P}}$ in the limit of infinite data.

This toy example illustrates how and when we can hope to achieve better predictive performance from $\overline{\mathcal{P}}, \widetilde{\mathcal{P}}$ than from $\overline{\mathcal{R}}, \widetilde{\mathcal{R}}$. Namely, if some mixture of our model can get closer to the true distribution than the best single setting of the parameters, we expect approaches that target the predictive risks to outperform the inferential risks by a corresponding margin.

6 Related Work

By far the work most closely related to the PAC^m bound is the PAC^2_T bound presented in Masegosa [2019]. PAC^2_T is based on a second order Jensen tightening of $\mathcal{P}$. While clearly instrumental to our work, PAC^2_T has a number of defects which PAC^m remedies. First, the variance tightening term in PAC^2_T is non-degenerate only for bounded likelihoods; PAC^m has no such restriction. Second, the

PAC^m risk, by directly targeting predictive risk satisfies the *golden rule*; the same cannot be said for PAC^2_T. Finally, in the experiments below we demonstrate that PAC^m generally matches or exceeds the test-set performance of PAC^2_T; as expected, both generally outperform ELBO.

The PAC^m proofs leverage multisample insights from the IWAE work [Burda et al., 2015]. In response, Rainforth et al. [2018] question the utility of these tighter class of bounds and demonstrate that tighter bounds on the *marginal evidence* do not help learn useful posteriors. This valuable insight does not apply to PAC^m because our bound is not on the evidence marginal $p(\{x_i\}_i^n) = \mathbb{E}_{r(\Theta)}[\prod_i^n p(x_i|\Theta)]$ but rather on the *posterior predictive distribution*, $p(X|\{x_i\}_i^n) = \mathbb{E}_{q(\Theta|\{x_i\}_i^n)}[p(X|\Theta)]$. In fact, in the limit $m \to \infty$ we already explicitly encode the idea that the actual Bayesian posterior is not directly useful.

PAC^m offers real benefits in predictive performance in cases of model misspecification, in particular, when a mixture of our model family would be a better predictive model than the model itself. If so, why not simply fit mixture models? This certainly does work, as fig. 7 demonstrates. Mixtures have proven difficult to fit in general [Morningstar et al., 2020]. The PAC^m family of risks subsume classic risks and offer theoretical generalization guarantees in cases of model misspecification, while remaining computationally tractable. For more discussion of the differences between mixtures of various sorts, please see appendix B.

7 Experimental Results

Here we demonstrate that our new risk: PAC^m, can achieve better predictive performance on a suite of tasks. In all experiments, we compare the PAC^m-Bayes risk ($\widetilde{\mathcal{P}}$) to alternative objectives, including the PAC-Inferential Risk ($\widetilde{\mathcal{R}}$), also called the Evidence Lower Bound (ELBO), since it is a lower bound on the marginal likelihood. We also compare to alternative PAC-style bounds on the predictive risk, namely the PAC^2_T objective proposed in Masegosa [2019].

Toy Experiments: We start with a series of three simple regression tasks designed to test three different flavors of model misspecification. The first task is to predict data from a sinusoidal model when the variance is underestimated. The second task is to predict data from a sinusoidal model with heteroskedastic noise, where we have assumed homoskedastic noise. The third task is to predict data from a mixture, assuming a unimodal posterior and likelihood. The exact generative models for each setup, along with the details of the predictive models and training procedures can be found in the supplement.

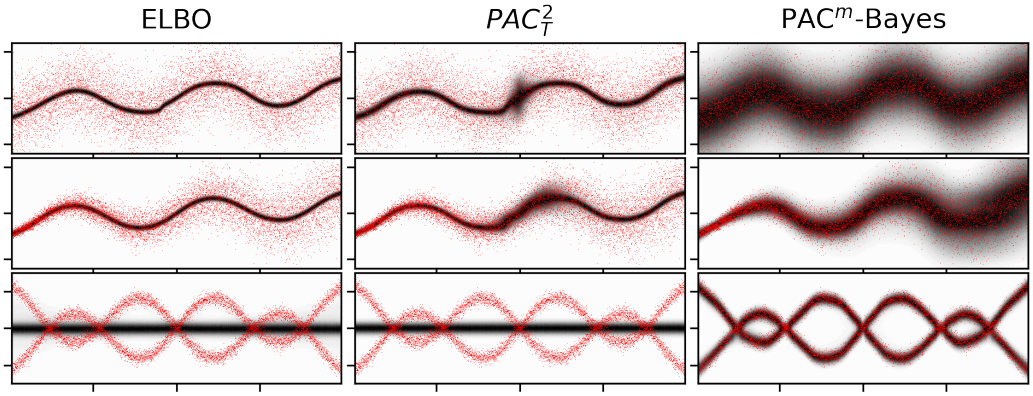

Figure 3: A comparison between ELBO, PAC^2_T, and PAC^{m}Bayes on three different toy regression problems where the model is misspecified. Datapoints are shown in red, and the black contours show the posterior predictive distribution learned by training with the loss indicated at the top of the corresponding column. Each row covers a different data generation process. All experiments use the same model family, computation, and optimization procedure, and only differ in their loss.

We show the predictive models along with the data in Figure 3. Our first observation is that in all three cases, optimizing the ELBO objective leads to a poor predictive model. This is expected behavior.

Here, because the number of training points is fairly large, ELBO causes the posterior predictive distribution to concentrate on the (true) mean of $p(y|x)$. Because the variance is underestimated, errors in the prediction of the mean are penalized more aggressively, exacerbating this concentration. In other words, incorrect specification of the model leads to increasingly overconfident (but wrong) predictions when training with ELBO. The worst-case scenario of this can be seen in the mixture experiments, when the mean of the data is often not a reasonable prediction of any of the data.

Our second observation is that while PAC^2_T does a marginally better job of accounting for the observed uncertainty (its predictive distribution is slightly wider than ELBO), it still underpredicts the variance of the data. We appear to observe that this loss results in the model expanding the tails of its predictive distribution to account for the observed variance. At the same time, it is clear that PAC^2_T still tends to concentrate its predictions on the mean of the data.

Experiment	ELBO	PAC^2_T	PAC^m
Sinusoid	46.2	2.2	**0.2**
Heteroskedastic	20.33	1.16	**0.03**
Mixture	14.21	11.57	**0.15**

Table 1: KL divergences between the learned posterior predictive model and the true predictive model. These were each computed using 1000 samples from the learned surrogate posterior distribution.

In all cases, we observe that PAC^m results in a better predictive distribution than the alternatives. We quantitatively assess the performance using the KL-Divergence between the posterior predictive distribution, and the true generative distribution. These are presented in Table 1. In all cases, ELBO performs the worst, while PAC^2_T performs better and PAC^m performs best. Interestingly, we see in Figure 3 that PAC^m recovers the multimodal posterior predictive distribution, despite the model having a unimodal posterior, prior, and likelihood. This indicates that the surrogate posterior learned by PAC^m is in some sense richer than that learned by ELBO, since it appears to exploit the architecture of the network in order to use its unimodal posterior to model multimodal data. To assess if this improvement can be simply replicated by using a more expressive posterior, we repeated the mixture experiment using a Mixture of Independent Normal distributions for the posterior in fig. 5. Here also, we find that PAC^m results in a better approximation to the true predictive model, having a lower KL divergence from the generative distribution ($\text{KL} = 0.63$) than the alternatives ($\text{KL} = 14.19$ for ELBO and $\text{KL} = 9.09$ for PAC^2_T). We also verify that all models perform well when the model is well specified in fig. 7, which for the mixture problem would mean having a two component likelihood.

Structured Prediction: We also test our objective on structured prediction tasks [e.g. Sohn et al., 2015]. For this, we train a Bayesian neural network to predict the bottom half of an image, using only the top half as an input. We test this on 3 different image datasets: MNIST [LeCun, 1998], FashionMNIST [Xiao et al., 2017], and CIFAR-10 [Krizhevsky, 2012]. For our likelihood, we use a Normal distribution where each pixel is considered independent. Following Masegosa [2019], we further fix the scale of the likelihood distribution to 1/255. These choices are interesting for two reasons. First, this setup also replicates the training setup which is often employed in training naive Variational Autoencoders [see e.g., Kingma and Welling, 2013, Tomczak and Welling, 2018], where the output variance is either fixed or shared between pixels (for non binarized images) and where all output pixels are assumed to be independent. Second, this setup is a misspecified model since we know that the pixels in the data are not independent, at least not at the granularity which we are able to capture in most models. We therefore hypothesize that PAC^m should be able to offer improvements in predictive performance.

We train models and measure performance as a function of m and the loss function used in training. For each value of m and each loss, we conduct 5 trials with different initializations to estimate the uncertainty in our final test set negative log-posterior-predictive probability. We show the results in Figure 4. We find that the performance of ELBO is roughly static in m, with much of the observed variation consistent with noise. This is consistent with our expectation. In contrast, PAC^2_T and PAC^m exhibit rapid improvement in performance with m, showing that the model is, in fact, misspecified and that these models are therefore able to offer meaningful improvement in predictive performance. Interestingly, we also appear to observe a saturation in m when the number of samples approaches the size of the batch. This could occur for two reasons. First, as we show in theorem 3, the model may ultimately cease to improve in m because the increasing value of ψ may overcome the tightening

of $\widetilde{\mathcal{P}}$. Alternatively, this could be due to empirical variance in the gradients introduced by minibatch training. This is similar to the findings from Rainforth et al. [2018] who showed that variance in the gradients results in an impedance to effective learning which eventually overcomes tightness. Alternative gradient estimators such as that from Tucker et al. [2018] may help to solve this issue.

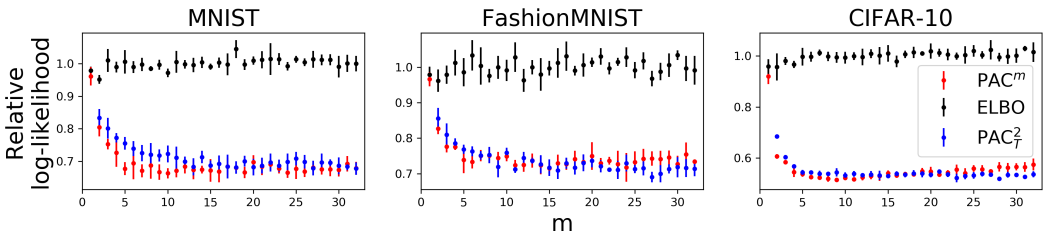

Figure 4: Test set log-posterior-predictive, relative to the mean log-posterior-predictive recovered using ELBO. Black points show models trained using ELBO, while red points show models trained using PAC^m and blue points show models trained with PAC^2_T. We see that PAC^m appears to offer the best log-posterior-predictive, but that performance either saturates or degrades as the sample number becomes to large, indicating issues with the optimization procedure.

Classification: So far, we have experimented with models where the likelihood was either purposefully misspecified in order to highlight the generalization gap introduced by minimizing $\widetilde{\mathcal{R}}$ compared to PAC^m or where we expect that it is misspecified because the assumptions we make about the output data are likely to be incorrect (e.g. pixels are likely non-independent in most images). It is unclear the degree to which this is an issue for many real-world applications where we use highly expressive deep neural network models, but in many cases we are still forced to make incorrect modeling assumptions for the sake of convenience.

It is equally interesting to consider the performance of PAC^m when the likelihood is well specified, but when other parts of the Bayesian model (the prior) are not. A good example of such a scenario is image classification, where we expect that a categorical distribution is a reasonable choice of likelihood. To test this scenario, in appendix D we present additional experiments where we use Bayesian convolutional neural networks to classify images from the datasets used in the previous section. We consider two cases: (1) being Bayesian over the weights of the model (the "global" variables), or (2) being Bayesian over the activations of the model (the "local" variables). This latter case has been explored in works such as Alemi et al. [2016]. Here we consider the same approach, except where we minimize a PAC-Bayesian bound on the predictive likelihood.

As we expect, for classification problems, we find that though the model appears to be mostly well-specified, PAC^m learns models that make better predictions at the same cost, measured in terms of the KL divergence between the posterior and the prior.

8 Conclusion

Something as simple as a one line change to a variational Bayes setup can have drastic effects. Swapping an expected log likelihood across multiple draws from a variational posterior with the log of the expected likelihood can vastly improve the predictive performance of badly misspecified models (summarized in appendix A). In this work we attempted to explain this phenomenon.

Bayesian inference minimizes a stochastic upper bound on the predictive risk but the tightness of this bound is limited by model misspecification. In this work we proposed PAC^m, a new bound on predictive risk which is asymptotically tight yet can also recover the traditional Bayesian posterior (but not simultaneously both). Moreover, minimizing this bound respects the *golden rule* by minimizing the same predictive loss we use to evaluate our model, while providing generalization guarantees. We demonstrated that PAC^m outperforms ELBO and PAC^2_T [Masegosa, 2019] on misspecified Bayesian models on a wide set of example problems.

Acknowledgments and Disclosure of Funding

We would like to thank earlier anonymous reviewers of this draft for feedback, as well as Ben Poole, Sergey Ioffe and Rif A. Saurous.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken elbo. In *International Conference on Machine Learning*, pages 159–168. PMLR, 2018.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Pierre Alquier, James Ridgway, and Nicolas Chopin. On the properties of variational approximations of gibbs posteriors. *The Journal of Machine Learning Research*, 17(1):8374–8414, 2016.

Kavosh Asadi and Michael L Littman. An alternative softmax operator for reinforcement learning. In *International Conference on Machine Learning*, pages 243–252, 2017.

ATLAS Collaboration. Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Physics Letters B*, 716(1):1 – 29, 2012. ISSN 0370-2693. doi: https://doi.org/10.1016/j.physletb.2012.08.020. URL `http://www.sciencedirect.com/science/article/pii/S037026931200857X`.

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

James O Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 1985.

James O Berger, Elías Moreno, Luis Raul Pericchi, M Jesús Bayarri, José M Bernardo, Juan A Cano, Julián De la Horra, Jacinto Martín, David Ríos-Insúa, Bruno Betrò, et al. An overview of robust bayesian analysis. *Test*, 3(1):5–124, 1994.

Pier Giovanni Bissiri, Chris C Holmes, and Stephen G Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 78(5):1103, 2016.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Accurate and conservative estimates of mrf log-likelihood using reverse annealing, 2014.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Olivier Catoni. Pac-bayesian supervised classification: the thermodynamics of statistical learning. *arXiv preprint arXiv:0712.0248*, 2007.

Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

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.

Pedro M Domingos. Why does bagging work? a bayesian account and its implications. In *KDD*, pages 155–158. Citeseer, 1997.

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.

Roger B Grosse, Siddharth Ancha, and Daniel M Roy. Measuring the reliability of mcmc inference with bidirectional monte carlo. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2451–2459. Curran Associates, Inc., 2016. URL `http://papers.nips.cc/paper/6290-measuring-the-reliability-of-mcmc-inference-with-bidirectional-monte-carlo.pdf`.

Peter Grünwald, Thijs Van Ommen, et al. Inconsistency of bayesian inference for misspecified linear models, and a proposal for repairing it. *Bayesian Analysis*, 12(4):1069–1103, 2017.

Benjamin Guedj. A primer on pac-bayesian learning, 2019.

Martin Jankowiak, Geoff Pleiss, and Jacob R. Gardner. Deep sigma point processes, 2020a.

Martin Jankowiak, Geoff Pleiss, and Jacob R. Gardner. Parametric gaussian process regressors, 2020b.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. Generalized variational inference: Three arguments for deriving new posteriors, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

Pierre-Simon Laplace. Mémoire sur les probabilités. *Mémoires de l'Académie Royale des sciences de Paris*, 1778:227–332, 1781.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Bruce G Lindsay. Mixture models: theory, geometry and applications. In *NSF-CBMS regional conference series in probability and statistics*, pages i–163. JSTOR, 1995.

Andres R Masegosa. Learning under model misspecification: Applications to variational and ensemble methods. *arXiv preprint arXiv:1912.08335v3*, 2019.

Thomas P Minka. Bayesian model averaging is not model combination. *Available electronically at http://www. stat. cmu. edu/minka/papers/bma. html*, pages 1–2, 2000.

Andriy Mnih and Danilo J. Rezende. Variational inference for monte carlo objectives, 2016.

Warren R Morningstar, Sharad M Vikram, Cusuh Ham, Andrew Gallagher, and Joshua V Dillon. Automatic differentiation variational inference with mixtures. *arXiv preprint arXiv:2003.01687*, 2020.

Kevin P Murphy. Conjugate bayesian analysis of the gaussian distribution. *def*, 1(2σ2):16, 2007.

Dan Piponi, Dave Moore, and Joshua V Dillon. Joint distributions for tensorflow probability. *arXiv preprint arXiv:2001.11819*, 2020.

Planck Collaboration VI. *Planck* 2018 results. VI. Cosmological parameters. *AAP, in press*, 2019.

Tom Rainforth, Adam R Kosiorek, Tuan Anh Le, Chris J Maddison, Maximilian Igl, Frank Wood, and Yee Whye Teh. Tighter variational bounds are not necessarily better. *arXiv preprint arXiv:1802.04537*, 2018.

Hannah Ritchie. Gender ratio. *Our World in Data*, 2019. https://ourworldindata.org/gender-ratio.

Rishit Sheth and Roni Khardon. Pseudo-bayesian learning via direct loss minimization with applications to sparse gaussian process models. In Cheng Zhang, Francisco Ruiz, Thang Bui, Adji Bousso Dieng, and Dawen Liang, editors, *Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference*, volume 118 of *Proceedings of Machine Learning Research*, pages 1–18. PMLR, 08 Dec 2020. URL `http://proceedings.mlr.press/v118/sheth20a.html`.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.

Joe Staines and David Barber. Variational optimization, 2012.

Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223. PMLR, 2018.

George Tucker, Dieterich Lawson, Shixiang Gu, and Chris J Maddison. Doubly reparameterized gradient estimators for monte carlo objectives. *arXiv preprint arXiv:1810.04152*, 2018.

Yong Wang. On fast computation of the non-parametric maximum likelihood estimate of a mixing distribution. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 69(2):185–198, 2007. ISSN 13697412, 14679868. URL `http://www.jstor.org/stable/4623262`.

Yadi Wei, Rishit Sheth, and Roni Khardon. Direct loss minimization algorithms for sparse gaussian processes, 2020.

Florian Wenzel, Kevin Roth, Bastiaan S. Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really?, 2020.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Yuling Yao, Aki Vehtari, Daniel Simpson, Andrew Gelman, et al. Using stacking to average bayesian predictive distributions (with discussion). *Bayesian Analysis*, 13(3):917–1007, 2018.

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

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We discuss several limitations of the work, in particular subtle issues related to our bound in the discussion in appendix C.3 as well as a broader set of limitations in appendix B
   (c) Did you discuss any potential negative societal impacts of your work? [No] The work is foundational in nature and so doesn't suggest specific societal impacts separate from the field in general.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Appendix C contains detailed theoretical assumptions made.
   (b) Did you include complete proofs of all theoretical results? [Yes] Appendix C contains detailed proofs.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Unclear, we are not releasing all of the experimental code to reproduce the figures exactly, but we are including code examples appendix E.1 that should allow one to replicate our study.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Appendix E gives details for all experiments run.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] For quantitative comparisons such as figs. 8 to 10 we report error bars.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Datasets used were cited.
   (b) Did you mention the license of the assets? [Yes] The datasets we used were part of `tensorflow_datasets` we mention this and point the reader to that resource for further licensing information.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Supplementary material for PAC^m-Bayes

A Quick Reference: Comparing Different Losses

Here we depict several losses closely related to PAC^m and highlight their structural similarities and differences. The likelihood is $p(y|Z)$ and the prior/posterior discrepancy term is $\frac{r(Z)}{q(Z)}$.

$$\mathsf{ELBO} \stackrel{\text{def}}{=} -\mathbb{E}_{q(Z^m)}\left[\frac{1}{m}\sum_j^m \log\left(p(y|Z_j)\right) + \frac{1}{m}\sum_j^m \log\left(\frac{r(Z_j)}{q(Z_j)}\right)\right]$$

$$\mathsf{PAC}^m \stackrel{\text{def}}{=} -\mathbb{E}_{q(Z^m)}\left[\log\left(\frac{1}{m}\sum_j^m p(y|Z_j)\right) + \frac{1}{\beta}\frac{1}{m}\sum_j^m \log\left(\frac{r(Z_j)}{q(Z_j)}\right)\right]$$

$$\mathsf{IWAE} \stackrel{\text{def}}{=} -\mathbb{E}_{q(Z^m)}\left[\log\left(\frac{1}{m}\sum_j^m p(y|Z_j)\frac{r(Z_j)}{q(Z_j)}\right)\right]$$

$$\mathsf{PAC}^2_T \stackrel{\text{def}}{=} \mathsf{ELBO} - \mathbb{E}_{q(Z^m)}\left[\mathsf{SampleVariance}(y, Z^m)\right]$$

Where we colored the average-log, log-average terms, re-framed all losses in terms of multiple samples, and where:

- ELBO is the evidence lower bound (as a loss). [Blei et al., 2017]
- IWAE is the importance weighted autoencoder loss of Burda et al. [2015].
- PAC^2_T is the loss of Masegosa [2019].

B FAQ

Below we informally address some readers' questions.

B.1 Why did you choose the names *predictive* and *inferential* risk?

These terms serve to distinguish between making the best prediction vs choosing the best model parameters. We chose the name *predictive risk* risk because this term corresponds with (among other things) the posterior *predictive* distribution. The term *inferential risk* was chosen to reflect a judgement of the likelihood of model parameters. While we realize that the machine learning community sometimes uses *inferential risk* in contexts which we call *predictive risk*, we felt that our use of the term has precedent in statistics and is therefore not unreasonable.

B.2 Is the bound non-vacuous?

For any fixed m: no. When m is fixed we have a slack term in our PAC bound similar to that seen in most other PAC bounds. As we show in appendix C.3, the bound slack can grow at best like $o(\log m)$ or–as we've opted in the paper–like $o(m)$. (If the bound grew in some polynomial of n this would indeed be a worrisome, if not vacuous result.)

The reason it is "ok" that the bound slack grows in m is because taking the limit of $m \to \infty$ is not an interesting nor recommended limit. Recall that the Bayesian formalism stipulates that $m = 1$ and that if the model is well-specified, the Bayesian posterior is *optimal*. Choosing $m > 1$ serves only as a stopgap to improve–but not cure–poor predictive risk guarantees due to the misspecified regime.

We believe that "relaxing" the Bayesian posterior in the sense of $m > 1$ is a valuable contribution to the ML and statistics communities because it is a small step toward bridging the impressive predictive performance of the frequentist methodology with the impressive explainability/transparency of the Bayesian methodology.

B.3 Can we recover the correct parameter value, if one exists?

This question–while very interesting–is intentionally out of scope for this research. One technique to answer this question would be to explore the asymptotic consistency/efficiency similar to the LeCam style proofs of maximum likelihood. However this would be a very different set of tools than the PAC approach and carry its own set of worries. I.e., "is asymptotic analysis 'reasonable?"' and "are typical regularity assumptions justified in the deep learning era?"

B.4 How is PAC^m different from using a mixture posterior?

The PAC^m loss could work *on top* of a mixture posterior (or not). That is, PAC^m simply repeatedly samples from whatever posterior family is being fit. We illustrate this in fig. 5.

In general one should expect better performance from either ELBO or PAC^m when using a richer posterior family. However, as our experiments and analysis shows, ELBO predictive risk is more sensitive to misspecification in the likelihood and/or prior.

B.5 How is PAC^m different from using a mixture likelihood?

The PAC^m loss is used to identify a distribution over the likelihood parameters, be the a likelihood mixture distribution or otherwise. In this sense, using a mixture likelihood and using PAC^m are largely orthogonal changes to the modeling setup, though they can be done to accomplish the same objectives.

That said, PAC^m seems to be particularly helpful when the likelihood and/or posterior family lacks the capacity to capture multimodality present in the data. Compare the results in fig. 3 to the results in appendix D.2. Taking the same misspecified model and switching to the PAC^m objective is enough to fit multimodal data well. Fitting a proper mixture model (as in modifying the likelihood to have two components) to the multimodal data works correctly as shown in appendix D.2. In this case the model is well specified. Details for this experiment can be found in appendix E.4. Note that when switching from the misspecified unimodal model to the well specified mixture we have increased the number of parameters in our model, additional parameters for our variational posterior to approximate. For a simple problem like this one, having our variational posterior predict two means instead of one isn't much of an additional cost, but for larger Bayesian models like Bayesian neural networks the additional burden of fitting the mixture is hard to ignore.

Overall, switching from a model to a mixture model is changing the structure of the model. Switching from ELBO to PAC^m is simply changing the loss. That simply changing the loss of a misspecified model can recover a lot of the predictive benefits of the much larger mixture model is the primary benefit PAC^m brings.

To further highlight the distinction, consider the choices you would have to make to a Bayesian mixture in order to recover the PAC^m objective. Starting with a mixture likelihood (m component likelihoods p_j with mixing weights w_j), and the ELBO objective:

$$-E_{q(Z)}\left[\log\left(\sum_j^m w_j p_j(y|Z)\right) + \frac{1}{\beta}\log\frac{r(Z)}{q(Z)}\right],$$

we will have to structure our mixture so that the parameters are non-overlapping. Let $Z^m = [Z_1, Z_2, \dots]$ denote the partition of all the component's parameters and choose a prior that factorizes in the same way:

$$-E_{q(Z^m)}\left[\log\left(\sum_j^m w_j p_j(y|Z_j)\right) + \frac{1}{\beta}\sum_j^m\log\frac{r(Z_j)}{q(Z_j)}\right].$$

One would now have to fix the mixture probabilities to be uniform ($w_j = \frac{1}{m}$) and ensure that each of the mixture components were replications of same model ($p_j = p$):

$$-E_{q(Z^m)}\left[\log\left(\frac{1}{m}\sum_j^m p(y|Z_j)\right) + \frac{1}{\beta}\sum_j^m\log\frac{r(Z^m)}{q(Z^m)}\right].$$

Further, not only would the distributional form of the variational approximation for each of the components have to be the same, but the parameters would have to be shared, i.e. the exact same variational posterior would be used for the parameters of each component of the mixture. This would generate the same objective as PAC^m (after rescaling β) at the cost of severe and specific choices. Instead, as we demonstrated, starting with a single mismatched model and simply targeting the predictive rather and inferential risks is a better way to arrive at PAC^m and a principled way to fit misspecified Bayesian models.

B.6 How is PAC^m different than training an ensemble?

Ordinarily when people train an ensemble they minimize the Empirical Inferential Risk ($\overline{\mathcal{R}}$) multiple times independently, and then average the predictions of the resulting point estimates. By targeting an inferential risk, this won't address misspecification in the way PAC^m can. One could target the Empirical Predictive Risk ($\overline{\mathcal{P}}$) directly, this is known as a non-parameteric mixture. For certain models this can perform quite well, but for rich enough model families this can severely overfit. PAC^m adds the KL regularization term that can prevent overfitting.

B.7 How is PAC^m different than IWAE?

IWAE gives a bound on the marginal likelihood, not the predictive distribution. Summarized in appendix A practically the difference is in how the ratio of the prior and posterior densities contributes. IWAE attempts to bound the marginal likelihood, in other words the prior predictive likelihood. PAC^m is a bound on the *posterior* predictive likelihood.

C Proofs

This section proves our main theoretical result (theorem 1) as well as presents additional theory relevant to PAC^m.

C.1 Relationship Between Predictive and Inferential Risks In the Presence of Model Misspecification

The following two results are adapted from Masegosa [2019] to our notation and given here for the reader's convenience. These results examine conditions under which solutions to the inferential risk, $\min_{q(\Theta)} \mathcal{R}[q]$, are equivalent to solutions to the predictive risk, $\min_{q(\Theta)} \mathcal{P}[q]$. That is, these lemmas show that model misspecification introduces a gap between *predictive risk* ($\mathcal{P}$) and *inferential risk* ($\mathcal{R}$). This gap is potentially problematic because machine learning practitioners care about $\mathcal{P}$ but minimize (an approximation of) $\mathcal{R}$.

Lemma 1. $\arg\min_{q(\Theta)} \mathcal{R}[q] \equiv \arg\min_{q(\Theta)} \mathcal{P}[q]$ *only if for any distribution* ρ *over* Θ*,*

$$\mathsf{KL}[\nu(X); p(X|\theta^{(ml)})] \leq \mathsf{KL}[\nu(X); \mathbb{E}_{\rho(\Theta)}[p(X|\Theta)]],$$

and $q^{(ml)}(\Theta) = \arg\min_{q(\Theta)} \mathcal{R}[q] \equiv \delta(\Theta - \theta^{(ml)})$ *where* δ *is the Dirac-delta distribution.*

Proof. (Sketch.) Note that,

$$\begin{aligned}\mathcal{P}[q] &= \mathsf{KL}[\nu(X), \mathbb{E}_{q(\Theta)}\, p(X|\Theta)] + \mathsf{H}[\nu(X)] \\ &\leq \mathbb{E}_{q(\Theta)}\, \mathsf{KL}[\nu(X), p(X|\Theta)] + \mathsf{H}[\nu(X)] \\ &= \mathcal{R}[q]\end{aligned}$$

where the inequality is Jensen's. Since the theorem condition implies $\mathcal{R}[q^*] \leq \mathcal{P}[q]$ then $\mathcal{R}[q^*] \leq \min_q \mathcal{P}[q] \leq \mathcal{R}[q^*]$ and the claim follows. (See Lemma 2 of Masegosa [2019] for original proof; our sketch is based on a sandwich argument.) □

Lemma 2. *If there exists a density* ρ *over* Θ *such that*

$$\mathsf{KL}[\nu(X); \mathbb{E}_{\rho(\Theta)}[p(X|\Theta)] < \mathsf{KL}[\nu(X); p(X|\theta^{(ml)}],$$

then a minimizer of $\mathcal{R}$ is not a minimizer of $\mathcal{P}$ where

$$q^{(\mathsf{ml})}(\Theta) \stackrel{def}{=} \arg\min_{q(\Theta)} \mathcal{R}[q] \equiv \delta(\Theta - \theta^{(\mathsf{ml})}),$$

where δ is the Dirac-delta distribution.

Proof. (Sketch.) The condition of this lemma implies that $q^{(\mathsf{ml})}(\Theta)$ cannot be a minimizer of $\mathcal{P}$ however it is the minimizer of $\mathcal{R}$. (See Lemma 3 of Masegosa [2019] for original proof.) □

C.2 PAC-Bayes Relationships

This section presents two well-known PAC-Bayes results as special cases of theorem 1.

Corollary 1. *Under the conditions of theorem 1, then with probability at least $1 - \xi$, $\mathcal{P}[q] \leq \widetilde{\mathcal{R}}_n[q; r, \beta] + \psi_1$.*

Proof. Immediate from theorem 1 when $m = 1$. □

Corollary 2. *The Bayesian posterior $p(\Theta|\{x_i\}_i^n) \propto r(\Theta) \prod_i^n p(x_i|\Theta)$ minimizes PAC^m when $m = \beta = 1$.*

Proof. PAC^m is equivalently PAC when $m = \beta = 1$ for which the claim is proven by Germain et al. [2016]. □

C.3 PAC^m-Bayes Theory

Theorem 2. *For all $q(\Theta)$ absolutely continuous with respect to $r(\Theta)$, $X^n \overset{iid}{\sim} \nu(X)$, $\beta \in (0, \infty)$, $n, m \in \mathbb{N}$, $p(x|\theta) \in (0, \infty)$ for all $\{x \in \mathcal{X} : \nu(x) > 0\} \times \{\theta \in \mathcal{T} : r(\theta) > 0\}$, $\xi \in (0, 1)$, and $\lambda_m \in [m, \infty)$, then with probability at least $1 - \xi$:*

$$\mathcal{P}[q] \leq \widetilde{\mathcal{P}}_{n,m}[q; r, \beta] + \psi(\nu, n, m, \beta, r, \xi) \tag{11}$$

and furthermore (unconditionally):

$$\widetilde{\mathcal{P}}_{n,m}[q; r, \beta] \leq \widetilde{\mathcal{P}}_{n,m-1}[q; r, \beta] \leq \widetilde{\mathcal{P}}_{n,1}[q; r, \beta] = \widetilde{\mathcal{R}}_n[q, r, \beta] \tag{12}$$

where:

$$\widetilde{\mathcal{P}}_{n,m}[q; r, \beta] \stackrel{def}{=} -\frac{1}{n} \sum_i^n \mathbb{E}_{q(\Theta^m)} \left[\log \left(\frac{1}{m} \sum_j^m p(x_i|\Theta_j) \right) \right] + \frac{m}{\lambda_m} \mathsf{KL}\left[q(\Theta); r(\Theta)\right] \stackrel{def}{=} \mathsf{PAC}^m \tag{13}$$

$$\psi(\nu, n, m, \beta, r, \xi) \stackrel{def}{=} \frac{1}{\lambda_m} \log \mathbb{E}_{\nu(X^n)} \mathbb{E}_{r(\Theta^m)} \left[e^{\lambda_m \Delta_{n,m}} \right] - \frac{1}{\lambda_m} \log \xi \tag{14}$$

$$\begin{aligned} \Delta_{n,m} &\stackrel{def}{=} \Delta(X^n, \Theta^m) \\ &\stackrel{def}{=} \frac{1}{n} \sum_i^n \log \left(\frac{1}{m} \sum_j^m p(X_i|\Theta_j) \right) - \mathbb{E}_{\nu(X)} \left[\log \left(\frac{1}{m} \sum_j^m p(X|\Theta_j) \right) \right] \end{aligned} \tag{15}$$

Proof. Write:

$$g(\Theta^m; X) \stackrel{\text{def}}{=} \frac{1}{m} \sum_j^m p(X|\Theta_j)$$

$$\overline{\mathcal{G}}_{n,m}[q] \stackrel{\text{def}}{=} -\frac{1}{n} \sum_i^n \mathbb{E}_{q(\Theta^m)} \left[\log g(\Theta^m; x_i) \right]$$

$$\mathcal{G}_m[q] \stackrel{\text{def}}{=} -\mathbb{E}_{\nu(X)} \mathbb{E}_{q(\Theta^m)} \left[\log g(\Theta^m; X) \right]$$

For the first claim:

Jensen's inequality implies

$$-\log \mathbb{E}_{q(\Theta^m)}\left[g(\Theta^m; X)\right] \leq \mathbb{E}_{q(\Theta^m)}\left[-\log g(\Theta^m; X)\right].$$

Applying $\mathbb{E}_{\nu(X)}$ to both sides implies $\mathcal{P}[q] \leq \mathcal{G}_m[q]$.

To complete the proof of the first claim, we now show

$$p(\mathcal{G}_m[q] \leq \widetilde{\mathcal{P}}_{n,m}[q; r, \beta] + \psi_{n,m}) \geq 1 - \xi.$$

Make the substitution, $f(\Theta^m; \{x_i\}_i^n) \stackrel{\text{def}}{=} \lambda_m \Delta(\{x_i\}_i^n, \Theta^m)$ (for some non-stochastic λ_m) to Lemma 3 ("Compression Lemma") and rearrange:

$$\begin{aligned}-\mathbb{E}_{q(\Theta^m)}\mathbb{E}_{\nu(X)}[\log g(\Theta^m; X)] \leq &-\mathbb{E}_{q(\Theta^m)}\mathbb{E}_{\nu(X|\{x_i\}_i^n)}[\log g(\Theta^m; X)] \\ &+ \tfrac{1}{\lambda_m}\mathsf{KL}[q(\Theta^m), r(\Theta^m)] + \tfrac{1}{\lambda_m}\log \mathbb{E}_{r(\Theta^m)}\left[e^{\lambda_m \Delta(\{x_i\}_i^n, \Theta^m)}\right].\end{aligned}$$

For the KL term, note that Lemma 5 ("KL-divergence iid") implies

$$\mathsf{KL}[q(\Theta^m), r(\Theta^m)] = m\,\mathsf{KL}[q(\Theta), r(\Theta)].$$

For the rightmost term (a log moment generating function conditioned on $\{x_i\}_i^n$), make substitutions $Z \stackrel{\text{def}}{=} \mathbb{E}_{r(\Theta^m)}[e^{\lambda_m \Delta(\{x_i\}_i^n, \Theta^m)}]$ and $p \stackrel{\text{def}}{=} \nu(X^n)$ to Lemma 4 ("Log Markov Inequality") to conclude:

$$\begin{aligned}\nu_{X^n}\Big(&\log \mathbb{E}_{r(\Theta^m)}\left[e^{\lambda_m \Delta(X^n, \Theta^m)}\middle| X^n\right] \\ &\leq \log \mathbb{E}_{\nu(X^n)}\mathbb{E}_{r(\Theta^m)}\left[e^{\lambda_m \Delta(X^n, \Theta^m)}\right] - \log \xi\Big) \geq 1 - \xi.\end{aligned}$$

Scale the inner inequality by $\frac{1}{\lambda_m}$ (which doesn't change the probability) and combine this result with the previous two to prove the first claim.

(This proof was inspired by Masegosa [2019].)

For the second claim:

Note that $m/\lambda_m \leq 1$ for all m hence the KL terms of $\widetilde{\mathcal{P}}_{n,m}$ and $\widetilde{\mathcal{R}}_n$–which are not otherwise a function of m–cannot grow in m and may be safely ignored. The equality $\widetilde{\mathcal{P}}_{n,1}[q; r, \beta] = \widetilde{\mathcal{R}}_n[q, r, \beta]$ is true by definition; $g(\Theta^1; X) = p(X|\Theta)$. To complete the proof it is sufficient to show $\overline{\mathcal{G}}_{n,m} \leq \overline{\mathcal{G}}_{n,m-1}$. I.e.,

$$\begin{aligned}\overline{\mathcal{G}}_{n,m}[q] &= -\frac{1}{n}\sum_i^n \mathbb{E}_{q(\Theta^m)}\left[\log \frac{1}{m}\sum_j^m p(x_i|\Theta_j)\right] \\ &= -\frac{1}{n}\sum_i^n \mathbb{E}_{q(\Theta^m)}\left[\log \frac{1}{m}\sum_j^m \frac{1}{m-1}\sum_{k\neq j}^m p(x_i|\Theta_k)\right] \\ &\leq -\frac{1}{m}\sum_j^m \frac{1}{n}\sum_i^n \mathbb{E}_{q(\Theta^m)}\left[\log \frac{1}{m-1}\sum_{k\neq j}^m p(x_i|\Theta_k)\right] \\ &= \frac{1}{m}\sum_j^m \overline{\mathcal{G}}_{n,m-1}[q] \\ &= \overline{\mathcal{G}}_{n,m-1}[q].\end{aligned}$$

The inequality is Jensen's and the second-to-last equality follows from Θ^m being independent.

(This proof is inspired by Burda et al. [2015].)

□

While Theorem 2 is technically true, additional assumptions are needed to ensure it is nonvacuous, i.e., $\psi(\nu, n, m, \beta, r, \xi) < \infty$. Theorem 3 (below) affirms this is the case when $\Delta(X, \theta)$ is *everywhere sub-gaussian* for all $\{\theta \in \mathcal{T} : r(\theta) > 0\}$ and furthermore suggests that the optimal λ is given by $\lambda^* = n\beta\sqrt{\log(\max(2, m))}$.

We emphasize that sub-gaussianity is only assumed for $n = m = 1$, yet our proof holds for $n, m \geq 1$. This assumption is similar to that made by Germain et al. [2016], however we assume $\Delta(X, \theta)$ is everywhere sub-gaussian whereas they assume $\Delta(X, \Theta)$ is *jointly sub-gaussian*. We note that their Corollaries 4 and 5 (the relevant claims) have incorrect proofs which do not obviously follow from joint sub-gaussianity; our Theorem 3 with $m = 1$ serves as a correction. As also indicated in Germain et al. [2016], our ψ's finiteness is also provable by the stronger assumption that $p(x|\theta) \in [a, b]$ where $a, b \in \mathbb{R}$ and for all $\{x \in \mathcal{X} : \nu(x) > 0\} \times \{\theta \in \mathcal{T} : r(\theta) > 0\}$. [Catoni, 2007, Alquier et al., 2016] However, we refain from making such claims, preferring the arguably more general assumptions of Theorem 3.

Theorem 3. *Making the assumptions of Theorem 2 except that $\lambda \in (0, \infty)$ and additionally that for all $\{\theta \in \mathcal{T} : r(\theta) > 0\}$, $\Delta(X, \theta)$ is sub-gaussian with standard deviation $s_\theta \in (0, s]$ for $s \in (0, \infty)$, i.e., $\log \mathbb{E}_{\nu(X)}\left[e^{\lambda\Delta(X,\theta)}\right] \leq \frac{1}{2}\lambda^2 s_\theta^2 \leq \frac{1}{2}\lambda^2 s^2$, then:*

$$\psi_{n,m} = \frac{1}{\lambda}\log \mathbb{E}_{\nu(X^n)}\,\mathbb{E}_{r(\Theta^m)}\left[e^{\lambda\Delta(X^n,\Theta^m)}\right] - \frac{1}{\lambda}\log\xi \tag{18}$$

$$\leq \left(\frac{\lambda s^2}{2n} + \frac{n\log m}{\lambda} + \log m\right) - \frac{1}{\lambda}\log\xi. \tag{19}$$

Additionally, writing $\beta = s^{-1}\sqrt{2}$ and assuming $\xi = 1$ then,

$$\lambda^* = n\beta\sqrt{\log\max(2, m)}, \tag{20}$$

minimizes Equation (19) *for $m > 1$ and is a constant when $m = 1$.*

Proof. Begin by noting that,

$$\Delta(x, \{\theta_j\}_j^m) \stackrel{\text{def}}{=} \log\frac{1}{m}\sum_j^m p(x|\theta_j) - \mathbb{E}_{\nu(X)}\left[\log\frac{1}{m}\sum_j^m p(X|\theta_j)\right] \tag{21}$$

$$\leq \max\left\{\log p(x|\theta_j)\right\}_j^m - \mathbb{E}_{\nu(X)}\left[\log\frac{1}{m}\sum_j^m p(X|\theta_j)\right] \tag{22}$$

$$= \max\left\{\log p(x|\theta_j) - \mathbb{E}_{\nu(X)}\left[\log\frac{1}{m}\sum_k^m p(X|\theta_k)\right]\right\}_j^m \tag{23}$$

$$\leq \max\left\{\log p(x|\theta_j) - \mathbb{E}_{\nu(X)}\left[\log p(X|\theta_j)\right]\right\}_j^m + \log m \tag{24}$$

$$= \max\left\{\Delta(x, \theta_j)\right\}_j^m + \log m. \tag{25}$$

The first inequality follows from the upper bound in Lemma 9. The second inequality follows from the negative of the lower bound in Lemma 9, i.e.,

$$-\log\frac{1}{m}\sum_k^m p(X|\theta_k) \leq -\max\{\log p(x|\theta_j)\}_j^m + \log m \leq -\log p(x|\theta_k) + \log m,$$

for all $k \in \{1, \ldots, m\}$.

Combining this fact and the fact that $e^{\max\{a_j\}_j^m} = \max\{e^{a_j}\}_j^m \leq \sum_j^m e^{a_j}$, implies:

$$e^{\frac{\lambda}{n}\Delta(x,\{\theta_j\}_j^m)} \leq e^{\frac{\lambda}{n}\left(\max\left\{\Delta(x,\theta_j)\right\}_j^m + \log m\right)} \tag{26}$$

$$= m^{\frac{\lambda}{n}} \max\left\{e^{\frac{\lambda}{n}\Delta(x,\theta_j)}\right\}_j^m \tag{27}$$

$$\leq m^{\frac{\lambda}{n}} \sum_j^m e^{\frac{\lambda}{n}\Delta(x,\theta_j)}. \tag{28}$$

Combining this fact with the everywhere sub-gaussianity of $\Delta(X,\theta)$ implies:

$$\log \mathbb{E}_{\nu(X^n)}\, \mathbb{E}_{r(\Theta^m)}\left[e^{\lambda \frac{1}{n}\sum_i^n \Delta(X_i,\Theta^m)}\right] \tag{29}$$

$$= \log \mathbb{E}_{r(\Theta^m)}\left[\prod_i^n \mathbb{E}_{\nu(X)}\left[e^{\frac{\lambda}{n}\Delta(X,\Theta^m)}\right]\right] \tag{30}$$

$$= \log \mathbb{E}_{r(\Theta^m)}\left[\left(\mathbb{E}_{\nu(X)}\left[e^{\frac{\lambda}{n}\Delta(X,\Theta^m)}\right]\right)^n\right] \tag{31}$$

$$\leq \log \mathbb{E}_{r(\Theta^m)}\left[\left(\mathbb{E}_{\nu(X)}\left[\sum_j^m e^{\frac{\lambda}{n}\Delta(X,\Theta_j)}\right]\right)^n\right] + \lambda \log m \tag{32}$$

$$= \log \mathbb{E}_{r(\Theta^m)}\left[\left(\sum_j^m \mathbb{E}_{\nu(X)}\left[e^{\frac{\lambda}{n}\Delta(X,\Theta_j)}\right]\right)^n\right] + \lambda \log m \tag{33}$$

$$\leq \log \mathbb{E}_{r(\Theta^m)}\left[\left(\sum_j^m e^{\frac{\lambda^2 s^2}{2n^2}}\right)^n\right] + \lambda \log m \tag{34}$$

$$= \log \mathbb{E}_{r(\Theta^m)}\left[\left(m e^{\frac{\lambda^2 s^2}{2n^2}}\right)^n\right] + \lambda \log m \tag{35}$$

$$= \frac{\lambda^2 s^2}{2n} + (\lambda + n)\log m \tag{36}$$

Adding $-\log \xi$ and scaling by $\frac{1}{\lambda}$ completes the first part of the proof.

It now remains to find the optimal λ for $\xi = 1$. For $m = 1$ we resign ourselves to finding a constant, e.g., $\lambda = n\beta\sqrt{\log 2}$ where $\beta = s^{-1}\sqrt{2}$. For $m > 1$ note that $\frac{\lambda s^2}{2n} + \frac{n \log m}{\lambda} + \log m$ is convex in $\lambda > 0$ since $m, n > 0$. Solving for the root of the gradient we find $\lambda^* = n\beta\sqrt{\log \max(2, m)}$. □

Theorem 3 indicates that $\lambda = o(n)$ is sufficient to ensure nonvacuousness of Theorem 1 for all n and a fixed, finite m. Unfortunately Theorem 3 also indicates that no choice of λ ensures $\psi_{n,m} \to 0$ as $n, m \to \infty$. That is, even for $\lambda = o(n\sqrt{\log m})$ the Theorem 1 bound loosens at rate $o(\log m)$. Worse, Theorem 1 actually assumes $\lambda = o(nm)$ to ensure monotonic tightening to the predictive risk (see λ_m assumption in Theorem 1); in this regime the Theorem 1 bound looses at rate $o(m)$. Despite these concerns we note the following facts:

1. Regardless of ψ growing at best like $o(\log m)$ or nominally like $o(m)$, Theorem 1 remains non-vacuous for any $\lambda = o(n)$ and finite m.
2. In practice we recommend choosing λ by cross-validation and for each n, m regime. This implies the n, m-parameterization is merely a theoretical consideration (especially in light of point 1 above).
3. Theorem 3 is an upper bound and may or may not be made tighter. Theorem 3 assumptions are arguably fairly weak and stronger assumptions might help, e.g., bounded likelihood or $\Delta(X, \{\theta\}_j^m)$ being everywhere sub-gaussian (as opposed to $\Delta(X, \theta)$ being everywhere sub-gaussian).

4. The practitioner would not typically use large m. Given that computational complexity also grows in m, we expect the vast majority of cases to use $m \leq 50$ and to see improvements over $m = 1$.

C.4 Lemmas

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

Lemma 3 (Compression)**.** *If $p(\Theta)$ is absolutely semicontinuous wrt $r(\Theta)$ and $\mathbb{E}_{r(\Theta)}[e^{f(\Theta)}] < \infty$, then $\mathbb{E}_{p(\Theta)}[f(\Theta)] \leq \mathsf{KL}\left[p(\Theta), r(\Theta)\right] + \log \mathbb{E}_{r(\Theta)}[e^{f(\Theta)}]$.*

Proof. Write $q(\Theta) \stackrel{\text{def}}{=} \frac{r(\Theta)e^{f(\Theta)}}{\mathbb{E}_{r(\Theta)}[e^{f(\Theta)}]}$ and note that Lemma 6 implies, $0 \leq \mathsf{KL}\left[p(\Theta), q(\Theta)\right] = \mathsf{KL}\left[p(\Theta), r(\Theta)\right] - \mathbb{E}_{p(\Theta)}[f(\Theta)] + \log \mathbb{E}_{r(\Theta)}[e^{f(\Theta)}]$. □

Proof due to Banerjee [2006], Zhang [2006].

Lemma 4 (Log Markov Inequality)**.** *For any $\xi \in (0, 1]$ and random variable $Z \sim p$ with $p(Z \leq 0) = 0$ then $p(\log Z \leq \log \mathbb{E}_p[Z] - \log \xi) \geq 1 - \xi$.*

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

Lemma 5 (KL-divergence iid)**.** *If $p(\Theta^m) \stackrel{\text{def}}{=} \prod_j^m p(\Theta_j)$ and $r(\Theta^m) \stackrel{\text{def}}{=} \prod_j^m r(\Theta_j)$, then $\mathsf{KL}[p(\Theta^m), r(\Theta^m)] = m\,\mathsf{KL}[p(\Theta), r(\Theta)]$.*

Proof. $\mathsf{KL}[p(\Theta^m), r(\Theta^m)] = \mathbb{E}_{\prod_j^m p(\Theta_j)}\left[\log \frac{\prod_j^m p(\Theta_j)}{\prod_j^m r(\Theta_j)}\right] = m\,\mathsf{KL}\left[p(\Theta), r(\Theta)\right]$. □

Lemma 6 (Gibb's Inequality)**.** *If $p(\Theta)$ is absolutely semicontinuous wrt $r(\Theta)$, then $\mathsf{KL}[p, q] \geq 0$.*

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

Lemma 7 (ψ non-negative)**.** *Under the conditions of theorem 1 and if $\mathcal{G}_m[r], \overline{\mathcal{G}}_{n,m}[r] < \infty$, then $\psi(\nu, n, m, \beta, r, \xi) \geq 0$.*

Proof. Jensen's inequality implies $e^{\mathbb{E}\,Z} \leq \mathbb{E}\,e^Z$. Applying $\log$ to both sides (a monotonically increasing function), implies $\mathbb{E}\,Z \leq \log \mathbb{E}\,e^Z$. Substitute $Z \stackrel{\text{def}}{=} \beta n m \Delta_{n,m}$ (see eq. (15)) and note $\mathbb{E}_{\nu(X^n)r(\Theta^m)}\,Z = 0$ by definition. Finally, note $\log z \leq z - 1$ for $z > 0$ implies $-\log \xi \geq 0$ for $\xi \in (0, 1]$. □

Lemma 8 (Log-Average-Exp Bound – Parametric)**.**

$$-\log \frac{1}{n}\sum_i^n e^{x_i} \leq \begin{cases} -\frac{1}{\phi}\log \frac{1}{n}\sum_i^n e^{\phi x_i} & 0 < \phi \leq 1, \\ -\frac{1}{n}\sum_i^n x_i & \phi = 0. \end{cases}$$

Proof. Write $\mathsf{lse}(x) = \log \sum_i^n e^{x_i}$ and $\mathsf{softmax}_i(a) = \exp\left(a_i - \mathsf{lse}(a)\right)$.

For the $\phi \in (0, 1]$ case, note that lse convexity and Jensen's inequality imply $\mathsf{lse}\left(\phi a + (1 - \phi)b\right) \leq \phi\,\mathsf{lse}(a) + (1 - \phi)\,\mathsf{lse}(b)$ for $a, b \in \mathbb{R}^n$. Multiplying by $-\frac{1}{\phi}$ and rearranging yields $-\mathsf{lse}(a) \leq -\frac{1}{\phi}\,\mathsf{lse}\left(\phi a + (1 - c)b\right) + \frac{1}{\phi}(1 - \phi)\,\mathsf{lse}(b)$. Substituting $a = x - \log n$ and $b = -\log n$ proves the first case.

For the $\phi = 0$ case, note that L'Hopital's rule implies:

$$\lim_{\phi \to 0} \frac{1}{\phi} \mathsf{lse}\left(\phi x - \log n\right) = \lim_{\phi \to 0} \frac{\frac{\partial}{\partial \phi} \mathsf{lse}\left(\phi x - \log n\right)}{\frac{\partial}{\partial \phi} \phi}$$
$$= \lim_{\phi \to 0} \frac{\sum_i^n \mathsf{softmax}_i\left(\phi x - \log n\right) x_i}{1} = \frac{1}{n} \sum_i^n x_i$$

since $\lim_{\phi \to 0} \mathsf{lse}(\phi x - \log n) = \lim_{\phi \to 0} \phi = 0$. The bound follows from this fact, the convexity of $-\log z$, and Jensen's inequality: $-\log \frac{1}{n} \sum_i^n e^{x_i} \leq -\frac{1}{n} \sum_i^n \log e^{x_i}$. □

Lemma 8 is potentially useful because it shows that minimizing $-\frac{1}{\phi} \log \sum_j^m p(X|\Theta_j)^\phi$ is still consistent with minimizing PAC^m (i.e., $\phi = 1$) in the sense that $\phi \in [0, 1]$ implies an upper bound. This result might be useful for mitigating some of the gradient variance observed in the Monte Carlo approximation of PAC^m for large m; this conjecture is left for future work. (We note that all experiments reported in this paper use $\phi = 1$.)

Lemma 8 similarly exists in Asadi and Littman [2017] though our proof differs slightly.

Lemma 9 (Log-Average-Exp Bound – Simple)**.**

$$\max\left(\frac{1}{n}\sum_i^n x_i, \max\{x_i\}_i^n - \log n\right) \leq \log \frac{1}{n} \sum_i^n e^{x_i} \leq \max\{x_i\}_i^n \tag{37}$$

Proof. For the upper bound, note that:

$$\log \frac{1}{n} \sum_i^n e^{x_i} = \max\{x_j\}_j^n + \log \frac{1}{m} \sum_i^n e^{x_i - \max\{x_j\}_j^n}$$
$$\leq \max\{x_j\}_j^n + \log \frac{1}{n} \sum_i^n e^0$$
$$= \max\{x_j\}_j^n$$

For the lower bound, note that:

$$\log \frac{1}{n} \sum_i^n e^{x_i} \geq \log e^{\max\{x_j\}_j^n} - \log n = \max\{x_j\}_j^n - \log n$$

and by Jensen's inequality,

$$-\log \frac{1}{n} \sum_i^n e^{x_i} \leq -\frac{1}{n} \sum_i^n \log e^{x_i}.$$

□

D Additional Experimental Results

In this section we present additional experimental results which were omitted from the main text due to space constraints.

D.1 Mixture

Figure 3 shows that PAC^m is able to accurately predict multimodal data even if the posterior, prior, and likelihood are all unimodal. To test if this is a result of an effectively more expressive posterior induced by training with the loss, we repeated this experiment, but using an explicitly multimodal posterior distribution: A mixture of two multivariate normals. Details of this experiment are presented in Appendix E.

In Figure 5, we show the predictive models learned by training with ELBO, PAC_T^2, and PAC^m. Similar to Figure 3, we find that ELBO consolidates all of its probability mass on the mean of the data,

aiming to maximize the expected log-likelihood of the data (the average squared deviation between the predicted mean and observed data). PAC^2_T improves upon this slightly because the expected log-likelihood term is in tension with the variance term which tries to maximize the difference in log-likelihood between different samples of the model.

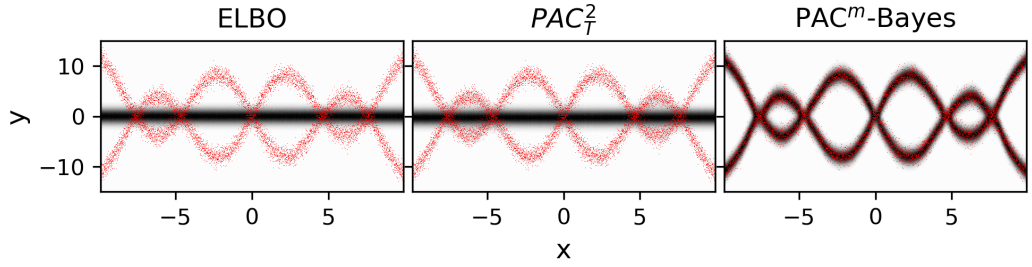

Figure 5: Similar to Figure 3, but where the surrogate posterior is multimodal. We find that the results are unchanged, despite the increased flexibility afforded to ELBO and PAC^2_T through the use of multiple modes in the posterior.

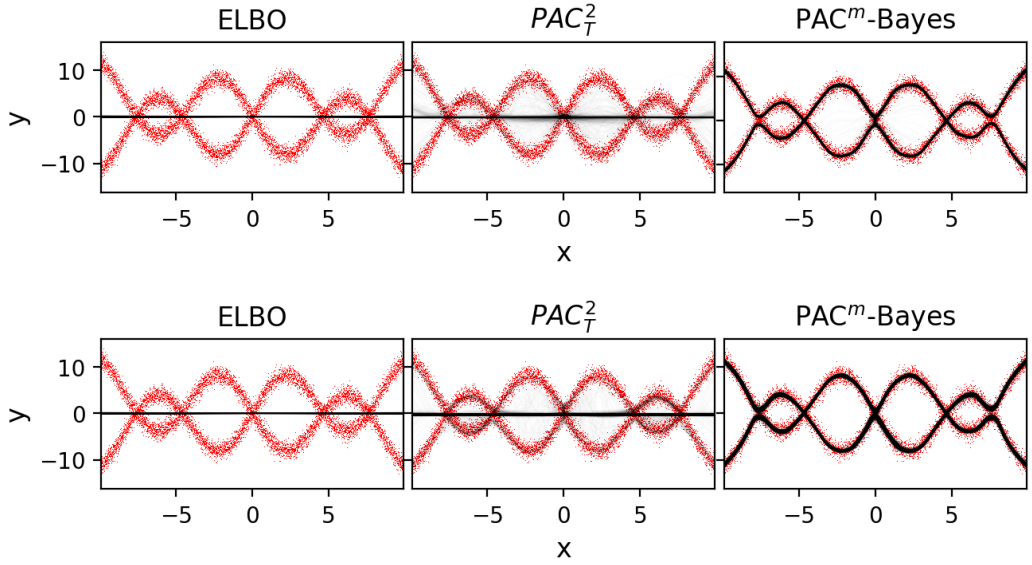

Figure 6: Visualization of the means of the likelihood predicted using samples from the posterior distribution. The top row shows the results for a unimodal posterior, and the bottom row shows the results for a multimodal posterior. We find that ELBO assigns all predictions to y=0, while PAC^2_T appears to have occasional samples that track the observed data. PAC^m only has samples which track the observed data.

Though we find that PAC^2_T does a much better job of predicting the data than ELBO (measured by the KL Divergence between the predictive model and the true generative model), this result is not obvious from looking at Figure 3 and Figure 5. In order to make this result more clear, instead of visualizing the histogram of samples from the predictive model, we instead draw 1000 samples from the posterior. For each posterior sample, we show the predicted mean of the output distribution as a function of x. Effectively, we want to see two curves tracing each mode in the output data. We show the results in Figure 6. For a 1 component model, we find that PAC^2_T places most of the probability mass on the mean, with tails that can be seen reaching toward the modes. However, we find that there are relatively few samples from the model which track any mode in the data. For a 2 component model, we find that there are proportionally more samples which track each of the modes, but these are still much less frequent than samples which merely follow the mean in the data. For reference, the peak probability density for samples near the data is roughly 30 times less than the density at the

mean. This is better than ELBO which places all of its probability near the mean. It is also worse than PAC^m, for which the 1 component model only has very few samples which fall away from either of the modes, and for which the 2 component model only has samples at each mode.

D.2 Well Specified Mixture

To show that all losses perform equivalently when the loss is well specified, we show here an experiment we ran where the likelihood is assumed to be multimodal. Here we used a mixture of normal distributions, with fixed categorical distribution and component variance. This left us to predict only the mean, similarly to the other mixture experiments. Additional details are presented in subsection E.4.

We show the results in Figure 7. As expected, since ELBO is tight for well specified models, it does a reasonable job of recovering the true predictive distribution. However, we should also note that the models which optimize bounds on the predictive risk also perform comparably. In fact, PAC^m observes a marginally lower KL Divergence from the true generative distribution. We measure KL $= 0.007$ for PAC^m, KL $= 0.017$ for ELBO, and KL $= 0.07$ for PAC^2_T. We did not evaluate if the discrepancy is simply due to variance in the optimization or if the lower KL observed from PAC^m is a result of optimizing a tighter bound.

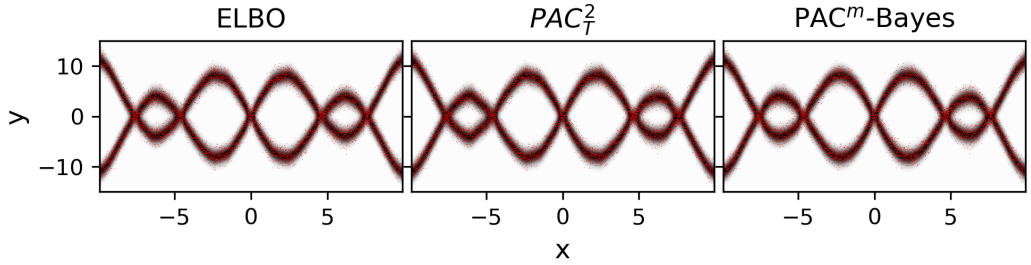

Figure 7: Similar to Figure 3, but where the likelihood is multimodal. This demonstrates performance for a well-specified model. We see that in this scenario, all three losses recover a good predictive model.

D.3 Bayesian Neural Network - Stochastic Weights

For classification experiments using stochastic weights, we give the model the full images from each dataset and attempt to predict the output class. For this, we assume the following graphical model:

$$\Theta \sim r(\Theta) \tag{38}$$

$$\text{for } i = 1 \dots n:$$

$$y_i \sim p(Y_i | z_T(x_i, \Theta)) \tag{39}$$

where z_T is the output of a T-layer neural network where each layer's parameters are specified by a partitioning of the random vector θ. For example, if z_T is a multilayer perceptron, it might be defined by the recurrence $z_t(x, \theta) = a_{t-1}(z_{t-1}(x, \theta))w_t + b_t$. where $\{(w_t, b_t)\}_t^T$ is partition of vector θ and with appropriately reshaped members and where $a(\cdot)w$ is a (row-) vector-matrix product.

We experimented with classification using a Bayesian Neural Network on several popular benchmarking datasets. Experimental details can be seen in subsection E.5. Similar to Alemi et al. [2018], we evaluate our models as a function of the constant β by producing the relationship between predictive negative log-likelihood (distortion) and KL divergence in the model (rate), a measure of compression of the model. For global experiments, we show the results in Figure 8. At the end of the day, all models achieve a comparable accuracy (within the experimental uncertainty). However, we find that PAC^{m}and PAC^2_T do so at lower rate than ELBO, and also have the dominant Pareto-frontier in the information, indicating that it may be doing a better job of distilling useful information from the data.

D.4 Bayesian Neural Network - Stochastic Activations

We also consider classification using a different, and non-traditional, type of Bayesian Neural Network wherein we treat the activations of an intermediate layer in the model as the random variables. This

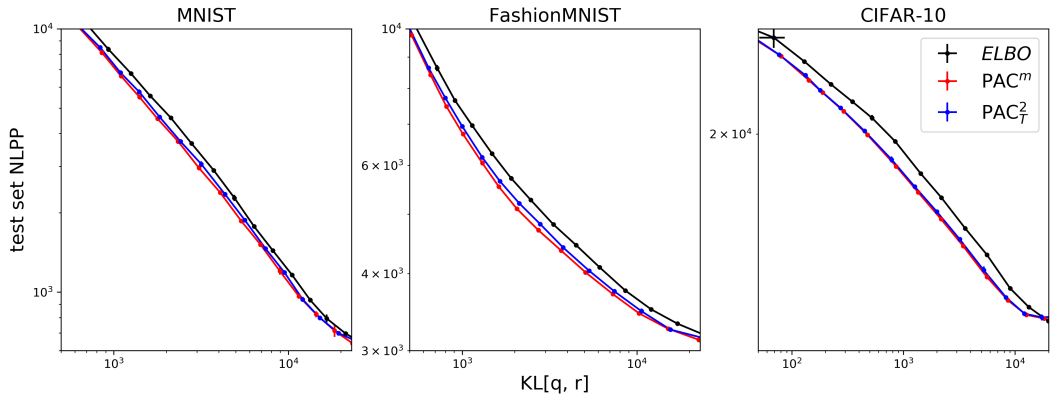

Figure 8: Predictive negative log-likelihood as a function of the KL divergence between the learned posterior and the prior (a measurement of the information content contained in the posterior distribution). The Pareto frontier for each model is shown as the solid line, as measured by the indicated points. Lower and to the left is "better." While all models have comparable performance, we find that models which optimize PAC-Bayesian bounds on the predictive likelihood do a better job of distilling information from the dataset, and therefore require fewer bits to produce equivalent accuracies. PAC^m performs best.

corresponds to the following graphical model.

$$\begin{aligned} &\text{for } i = 1 \dots n: \\ &\quad Z_i \sim r(Z) && (40) \\ &\quad y_i \sim p(Y_i|Z_i) && (41) \end{aligned}$$

In this formulation neither evidence x_i nor deep neural network are directly present in the assumed generative process. Rather, these ideas appear only in the construction of the surrogate posterior, i.e., $Z_i \sim q(Z|x_i, \theta)$. For example, one might assume $q(Z_i|x_i, \theta) \equiv \mathsf{Normal}(\mu_x, \sigma_x)$ where μ_x, σ_x are computed from two outputs of a DNN evaluated on x_i and using parameters θ (both of which are regarded as being non-stochastic). This type of setup is familiarized by Variational Autoencoders Kingma and Welling [2013], and in deep variational information bottleneck Alemi et al. [2016] models which use this graphical model to optimize for the log-evidence (or a bound on mutual informations) to set up either an unsupervised generative model (VAE) or a supervised predictive model (VIB). For these experiments, we follow this previous work and use a deep neural network as an "encoder" which predicts the parameters of the posterior distribution, and a "decoder" which uses the latent variable to define $p(Y_i|Z_i)$. As in Kingma and Welling [2013], we use the *reparameterization trick* to differentiate through the posterior sampling, which facilitates the optimization of the encoder parameters.

Experimental details are presented in subsection E.5. To evaluate performance, we show the Negative log-posterior-predictive probability as a function of the KL divergence between the posterior and the prior. The results are shown in Figure 9. Notably, we find that PAC^m has a Pareto frontier which advances noticeably beyond the other alternatives. This means that the model needs to learn less information in the posterior in order to make reasonable predictions on the data. We further show the classification accuracy as a function of KL in Figure 10. We find that both PAC^m and PAC^2_T appear to require significantly less information in the latent representation in order to make useful predictions. PAC^m still appears to have the dominant Pareto frontier in this space, though it is often ambiguous that it performs "better" than PAC^2_T in this space. However, it still appears that both outperform ELBO which appears to undergo posterior collapse at relative high rates, as indicated by the sudden sharp decrease in accuracy.

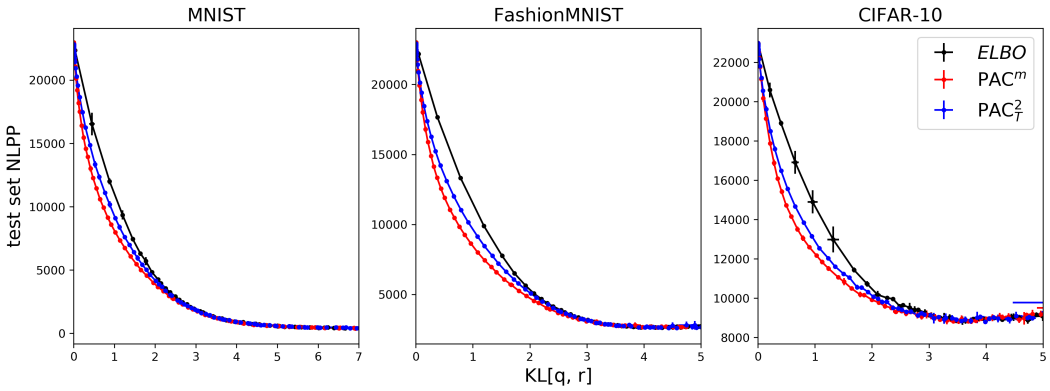

Figure 9: Similar to Figure 8, but where the posterior is defined over activations of an intermediate layer of the network, rather than all of the weights. Similar to before, lower and to the left is "better." We find that in this context, PAC^m clearly has the dominant pareto frontier.

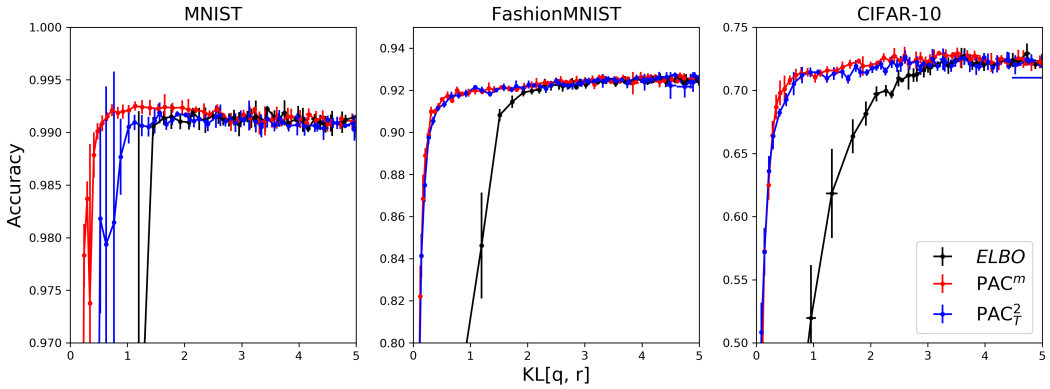

Figure 10: Classification accuracy as a function of the KL Divergence between the posterior and the prior. PAC^m and PAC^2_T consistently offer higher accuracy as a function of KL (i.e. for more *compressed* posteriors), with PAC^m appearing to do slightly better on MNIST. Sharp decreases in classification accuracy, along with corresponding large uncertainty in final accuracy correspond to a sudden collapse of the posterior which occurs for sufficiently large β.

E Experimental Details

E.1 Example Code

Here we provide example code for computing each loss. In all cases, we assume that one has a posterior, prior, and likelihood, where the posterior and prior are over the weights of the model, and the likelihood is a function which takes in weights and inputs and returns a probability distribution over outputs. All of the following use tensorflow and tensorflow probability Abadi et al. [2016], Dillon et al. [2017]. Additional arguments are xy and y; the inputs to the model and outputs from the model, m; the number of samples to draw from the posterior, β; the weight to place on the KL penalty, and n; the number of examples in the dataset.

Note that PAC^m and ELBO are almost identical. The only difference between the two is that PAC^m uses a log-mean-exp over the sample dimensions to get the negative log-posterior-predictive probability rather than the expected negative log-likelihood. Note also that this is not the case with PAC^2_T, which relies on the additional computation of a complicated variance term. This term has memory and compute cost which scales in the number of samples, though this will likely be sub-dominant to the memory cost of the forward pass in the network itself. It also relies on several tricks to encourage stability, and for the likelihood to be bounded in order for it to not converge to $-\infty$.

```
def elbo(prior, likelihood, posterior, x, y, m, beta, n):
  w = posterior.sample(m)
  ll = likelihood(x, w).log_prob(y)
  kl = tf.reduce_mean(
      posterior.log_prob(w) - prior.log_prob(w),
      axis=0)
  nll = -tf.reduce_mean(ll, axis=(0, 1))
  return nll + kl / (beta * n)
```

Figure 11: TF Probability [Dillon et al., 2017] implementation of ELBO loss for a unimodal global latent variable models (e.g., BNN).

```
def pacm(prior, likelihood, posterior, x, y, m, beta, n):
  w = posterior.sample(m)
  ll = likelihood(x, w).log_prob(y)
  kl = tf.reduce_mean(
      posterior.log_prob(w) - prior.log_prob(w),
      axis=0)
  nlpp = -tf.reduce_mean(
      tfp.math.reduce_logmeanexp(ll, axis=0),
      axis=0)
  return nlpp + kl / (beta * n)
```

Figure 12: TF Probability [Dillon et al., 2017] implementation of PAC^m loss for a unimodal global latent variable models (e.g., BNN). Note that this is identical to ELBO, with the exception of the use of the negative log-posterior-predictive rather than the negative log-likelihood.

```
def pac2t(prior, likelihood, posterior, x, y, m, beta, n,
    smoothing_constant=0.1):
  w = posterior.sample(m)
  ll = likelihood(x, w).log_prob(y)
  nll = -tf.reduce_mean(ll, axis=(0, 1))
  kl = tf.reduce_mean(posterior.log_prob(w) - prior.log_prob(w), axis=0)
  # We now compute the Masegosa "variance."
  lmx = tf.stop_gradient(
      tf.reduce_max(ll, axis=0, keepdims=True) + smoothing_constant)
  ll_max_centered = ll - lmx
  al = tfp.math.reduce_logmeanexp(ll_max_centered, axis=0)
  h = 2. * tf.stop_gradient(al / (1 - tf.math.exp(al))**2 +
      1. / (tf.math.exp(al) * (1 - tf.math.exp(al))))
  var1 = h * tf.math.exp(2 * ll_mac_centered)
  var2 = tf.math.reduce_mean(
      h * tf.math.exp(
          ll_max_centered[tf.newaxis] +
          ll_max_centered[:, tf.newaxis]),
      axis=0)
  variance = tf.math.reduce_mean(var1 - var2, axis=(0, 1))
  return nll - variance + kl / (beta * n)
```

Figure 13: TF Probability [Dillon et al., 2017] implementation of PAC^2_T loss for a unimodal global latent variable models (e.g., BNN).

E.2 Toy Model

The toy problem in fig. 2 was as described in section 5. The true data distribution came from a 30-70 mixture of two Normal distributions, with a variance of 1 and means at -2 and 2. The model was a standard Normal with fixed unit variance, the only learned parameter being the mean. Five datapoints were drawn, as indicated by the hash marks near the axis in the figures. The inferential risks were determined analytically, as the solution takes on the closed form [Murphy, 2007].

For the PAC-predictive risk, the posterior was found numerically with an iterative procedure. The sought after parameter distribution was represented by the values the density took on a grid with 500 points from -30 to 30. If we take $m \to \infty$ in $\widetilde{\mathcal{P}}_{n,m}$ in eq. (13), we have:

$$\widetilde{\mathcal{P}}_{n,\infty}[q; r, \beta] = -\frac{1}{n}\sum_{i}^{n} \log\left(\int d\theta\, q(\theta) p(x_i|\theta)\right) + \frac{1}{\beta n}\mathsf{KL}\left[q(\Theta); r(\Theta)\right] \tag{42}$$

Trying to minimize this functional with respect to $q(\Theta)$ using calculus of variations (along with the constraint that $q(\Theta)$ integrates to 1) suggests an iterative procedure to find the optimal parameter distribution:

$$q^{n+1}(\Theta) \propto r(\Theta) \exp\left(\beta \sum_i \frac{p(x_i|\Theta)}{p^{(n)}(x_i)}\right) \tag{43}$$

$$p^{(n+1)}(x_i) = \alpha p^{(n)}(x_i) + (1-\alpha)\int d\theta\, q^{(n+1)}(\theta) p(x_i|\theta). \tag{44}$$

We iterated these equations numerically, representing the parameter distribution as the values it took on a grid of 500 points between -30, and 30. eq. (43) sets the new estimate for the parameter distribution in terms of the current estimate for the data point marginal likelihoods. Notice the proportionality here, as we then numerically normalized the density after setting it to the right hand side of eq. (43). Then in eq. (44) we update our estimates of the data point marginal likelihoods, which act as sort of weights for the generalized Boltzmann distribution that is our parameter distribution. For the figure in the paper the mixing fraction α was set to 0.9.

The empirical predictive risk was minimized numerically. For the empirical predictive risk, an explicit mixture was fit, in this instance a 300 component Normal distribution, all with fixed unit variance. This is akin to searching for a 300 component atomic posterior distribution ($q(\Theta) = \sum_i \lambda_i \delta(\Theta - \theta_i)$). This was minimized with `adagrad` trained until it reached a fixed point to within a tolerance of 10^{-5}. Repeated runs all gave the same result. Though this was assuming the parameter distribution was itself atomic, experiments with a setup as was done for the PAC-predictive risk verified that the parameter distribution quickly does collect to an delta-comb.

E.3 Sinusoid

As mentioned in section 7, for our first experiment we tried to predict data drawn from the following sinusoid model:

$$\begin{aligned} &\text{for } i = 1 \ldots n: \\ &\quad \mu_{x_i} = 7\sin\left(\frac{3x_i}{4}\right) + \frac{x_i}{2} && (45) \\ &\quad Y_i \sim \mathsf{Normal}(\mu_{x_i}, 10). && (46) \end{aligned}$$

We generate data for 10^4 evenly spaced values of $x \in [-10.5, 10.5]$.

For our neural network, we use a two layer MLP with 20 hidden units, and a hyperbolic tangent activation function. We use a Normal distribution for the posterior, with both mean and variance as trainable variables. The initial values of the means were set to 0, and the initial variances were set to 1. The variances were constrained to be positive using the $\exp$ bijector available in tensorflow probability. We similarly use $\mathsf{Normal}(0, 1)$ for the prior over each weight and bias. For the likelihood, we use the MLP to predict the mean of a normal distribution, whose variance is fixed to 1.

We train all models using Adam [Kingma and Ba, 2014] with a learning rate of 0.01 and no learning rate decay. We use full batch training, for 10^5 steps. For fig. 3, we used $m = 100$ samples from the posterior during training, and $\beta = 1$ for all models. For all models, we evaluate performance using the log-posterior predictive, constructed using 10^3 samples from the posterior.

E.4 Mixture Experiments

For our second experiment, we use data generated from a two component mixture distribution:

$$\text{for } i = 1 \dots n:$$

$$\mu_{x_i} = 7 \sin\left(\frac{3x_i}{4}\right) + \frac{x_i}{2} \tag{47}$$

$$Z_i \sim \mathsf{Rademacher} \tag{48}$$

$$Y_i \sim \mathsf{Normal}(Z_i \mu_{x_i}, 1) \tag{49}$$

The model setup was largely similar to the Sinusoid experiment described in appendix E.3, but with one major difference: For these experiments we added an additional hidden layer to the networks to aid in expressiveness. We also used Exponential Linear Unit (ELU) activations instead of tanh to facilitate gradient propagation more easily. For these experiments we used both a unimodal posterior, as well as a mixture posterior, but the underlying distribution was implemented similarly to the sinusoid (i.e. a Normal distribution with learnable mean and variance). When considering a multimodal posterior, we fixed the component probabilities to 0.5 and used stratified sampling to integrate over the discrete categorical random variable. For unimodal likelihoods, we used a normal distribution whose mean was predicted by the model, and which had a fixed variance of 1. When considering a mixture likelihood, we compared situations with both 1 and 2 components in the posterior. The MLP was set up to predict the means of a two component mixture of Gaussian distributions, whose component probabilities were fixed to 0.5, and whose component variances were fixed to 1.

All models were again trained using Adam with an initial learning rate of 0.01, but this time we added a small amount of learning rate decay with a decay rate of 0.5 and a decay timescale of 10^5 steps. Because the model was only trained for 10^5 steps, the learning rate only undergoes one half-life. We did not study if holding the learning rate fixed changed the results at all, though it is doubtful that it did. We employed full batch training, and used $m = 100$ samples from the posterior. To evaluate the models qualitatively, as in fig. 3, we used 10^5 samples from the posterior to construct the predictive distribution. For each sample, we computed a forward pass for 10^3 evenly spaced values of x and drew a single sample from the resulting likelihood. This gave us 10^5 samples from the predictive distribution for each x. We then computed the 1-d histogram of the predictive distribution at each x, which we used to display the predictive models as in fig. 3. To quantitatively evaluate models, we used 10^4 samples from the posterior to construct the predictive distribution, and then computed the KL divergence from the known generative distribution for each of the (x, y) pairs in an independently generated test set.

E.5 Image Experiments

E.5.1 Structured Prediction

For the structured prediction experiments, we attempt to solve the following problem: Given the top half of an image, predict the bottom half of the image. We follow the setup in Masegosa [2019], for this experiment which we now describe. We use the experimental TFP Neural Networking toolbox (`tfp.experimental.nn`, [Dillon et al., 2017]) and TFP Joint Distributions [Piponi et al., 2020] to compactly specify BNNs. For the posterior and prior, we used Normal distributions. The posterior had learnable variables to represent the mean and variance, while the mean and variance in the prior are assumed to be constant. The model predicts the means of independent Normal distributions with fixed variance of $1/255$ which we use as our likelihood for each pixel. We assume that all pixels are independent and therefore ignore covariance between pixels in the output distribution. Note that this assumption is (purposefully) incorrect; images generally have strong correlations between adjacent spatial pixels and between the colors within a single pixel. For the network architecture, we used a 3-layer MLP with 50 hidden units, and ELU activations. We therefore fed our input images to the model as flattened vectors. For CIFAR-10, we converted the image to grayscale to reduce the number of pixels and simplify the model.

All models were trained using Adam with an initial learning rate of 0.001, decayed by 0.5 every 10^5 steps. We train models for 500 epochs. We used a batch size of 128 during training. We tested performance on the heldout evaluation set as a function of m, ranging from $m = 1$ to $m = 32$, where

each m corresponds to the number of samples used during training. Reconstruction performance was quantified using the log posterior-predictive (nlpp), which we measure using 100 samples from the posterior. We train 5 different models independently, with different random initializations and different shufflings of the training set in order to obtain the uncertainties in the final performance of the model.

E.5.2 Classification - Stochastic Weights

For our likelihood, we use a categorical distribution with 10 possible outcomes (all image datasets we consider have 10 output classes). Similar to previous experiments, we used Normal distributions for both the Posterior and the Prior, where the posterior uses variables to represent the location and scale of the normal distribution, and where the prior uses fixed values, both of which were initialized or fixed to 0 for the location and 1 for the scale respectively. We used the same architecture as the structured prediction experiments: A 3-layer MLP with 50 hidden units and ELU activations. All input images were normalized to the [-1, 1] interval before being passed to the first layer of the network.

Similar to the structured prediction experiments, we trained with a batch size of 128. We optimized our model for 100 epochs using Adam, with an initial learning rate of 10^{-4}, which we decayed by a factor of 0.5 every 10^5 steps. We study performance as a function of β, and fix the number of samples used during training to $m = 4$. We consider 33 values of c spaced logarithmically and ranging from $\left[10^{-3} - 10^{3}\right]$. We evaluate performance by computing both the log-posterior-predictive nlpp, on the heldout evaluation set. We use 100 samples to construct the posterior predictive distribution.

E.5.3 Classification - Stochastic Activations

For this model, we use the latent embedding z_i to codify each example in a 16 dimensional latent space. To increase the capacity of the model and since the memory cost is much lower, we consider a convolutional neural network for the encoder. This layer uses the following architecture[2]: 4 convolutional layers, followed by 2 dense layers. For each layer, we use LeakyReLU activations. Alternating convolutional layers use a stride of 2. The last dense layer predicts the parameters of the posterior. All images were normalized to the range [-1, 1] prior to being passed to the network.

For the posterior, we used a Multivariate Normal Distribution. The encoder predicts the location and the cholesky decomposition of the precision matrix. For the prior, we used a Multivariate Normal distribution with mean 0, and an identity covariance. For the likelihood, we used a categorical distribution. During training, we used a batch size of 128. We optimized our model for 100 epochs, using Adam with an initial learning rate of 10^{-4}, which was decayed by a factor of 0.5 every 10^5 training steps. Similar to our stochastic weights experiments, we evaluated performance as a function of β using 100 log-spaced bins between 10^{-3} and 10. We evaluate performance by computing both the nlpp and the Accuracy on the evaluation set. For both, we used 100 samples from the posterior to construct the predictive distribution.

[2]This architecture follows the encoder from the VAE example at `https://www.tensorflow.org/probability/examples/Probabilistic_Layers_VAE`