5th Symposium on Advances in Approximate Bayesian Inference, 2023 1–31

Variational Bayesian Last Layers

James Harrison[1], **John Willes**[2], **Jasper Snoek**[1]
[1] *Google DeepMind,* [2] *Vector Institute*

Abstract

We introduce a deterministic variational formulation for training Bayesian last layer neural networks. This yields a sampling-free, single-forward pass objective that effectively improves network uncertainty representation. Our variational Bayesian last layer can be trained and evaluated inexpensively, with only quadratic complexity in last layer width, and is thus (nearly) computationally free to add to existing architectures.

1. Introduction

Well-calibrated uncertainty quantification is essential for robust decision-making with machine learning systems. However, many methods for improving uncertainty quantification in deep learning (including Bayesian methods) have seen limited application due to their complexity over standard deep learning. For example, methods such as sampling-based mean field variational inference (Blundell et al., 2015), Markov chain Monte Carlo (MCMC) methods (Papamarkou et al., 2022; Neal, 1995; Izmailov et al., 2021), and comparatively simple heuristics such as Bayesian dropout (Gal and Ghahramani, 2016) all have substantially higher computational cost than baseline networks. Single-pass methods (so named because only one network evaluation is required) often require substantial modifications to network architectures, regularization, or training and evaluation procedures, even for the simplest such models (Liu et al., 2022; Wilson et al., 2016b; Kristiadi et al., 2021).

In this work, we investigate variational learning of Bayesian last layer (BLL) neural networks. In contrast to previous Bayesian deep learning formulations, BLL models consider only the uncertainty over the output layer of the network. While this is relatively restrictive in representational ability, it enables desirable features such as exact inference under certain distributional assumptions and exact likelihood computation. To train these variational BLL (VBLL) models, we derive deterministic lower bounds on the marginal likelihood. In contrast to standard variational training strategies, our bounds can be computed without sampling and may be trained with standard mini-batch training strategies. Moreover, we show that all resulting training loss terms may be computed with complexity no greater than standard neural network training complexity.

2. Bayesian Last Layer Neural Networks

We first introduce Bayesian last layer models which maintain a non-point posterior only for the last layer in a neural network. These models correspond to Bayesian (linear or logistic) regression or Bayesian Gaussian discriminant analysis (for each of the three models we present, respectively) with learned features. The subscript $t \in \mathbb{N}$ is used to index the data, and we assume T total data points.

2.1. Regression

The canonical BLL model for the regression case[1] is $\boldsymbol{y}_t = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_t, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}_t$ where $\boldsymbol{\phi} : \mathbb{R}^{N_x} \times \Theta \to \mathbb{R}^{N_\phi}$ is a vector of neural network features; the input $\boldsymbol{\theta} \in \Theta$ is the weights of the neural network[2]. We will typically write $\boldsymbol{\phi}_t := \boldsymbol{\phi}(\boldsymbol{x}_t, \boldsymbol{\theta})$ for notational convenience and refer to these parameters as *features* because they define the map from inputs to the feature embedding on which the BLL operates. The term $\boldsymbol{\varepsilon}_t$ is assumed to be normally distributed with zero mean and covariance Σ, and these noise terms are i.i.d. across realizations.

We specify a Gaussian prior $p(\boldsymbol{w}) = \mathcal{N}(\bar{\boldsymbol{w}}_0, S_0)$, assumed independent of the noise $\boldsymbol{\varepsilon}_t$. Posterior inference in the BLL model is analytically tractable for a fixed set of features. The marginal likelihood may be computed either via direct computation or by iterating over the dataset. Fixing a distribution over $\boldsymbol{w}$ of the form $\mathcal{N}(\bar{\boldsymbol{w}}, S)$ (and writing $\boldsymbol{\eta} = (\bar{\boldsymbol{w}}, S)$) the posterior predictive is

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\eta}, \boldsymbol{\theta}) = \mathcal{N}(\bar{\boldsymbol{w}}^\top \boldsymbol{\phi}_t, \boldsymbol{\phi}_t^\top S \boldsymbol{\phi}_t + \Sigma). \tag{1}$$

2.2. Discriminative Classification

In this subsection we introduce a BLL model that closely matches standard classification neural networks. We assume N_y classes, and we aim to predict the correct class given an input $\boldsymbol{x}_t$. We write T_y for the number of data points belonging to class $\boldsymbol{y}$. Our model takes the form

$$\boldsymbol{z}_t = W \boldsymbol{\phi}(\boldsymbol{x}_t, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}_t, \qquad p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, W, \boldsymbol{\theta}) = \mathrm{softmax}(\boldsymbol{z}_t) \tag{2}$$

where $\boldsymbol{z}_t \in \mathbb{R}^{N_y}$ are the logits. These are also interpreted as unnormalized joint data-label log likelihoods (Grathwohl et al., 2020), where

$$\boldsymbol{z}_t = \log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid W, \boldsymbol{\theta}) - Z(W, \boldsymbol{\theta}) \tag{3}$$

where $Z(W, \boldsymbol{\theta})$ is a normalizing constant, independent of the data. The term $\boldsymbol{\varepsilon}_t \in \mathbb{R}^{N_y}$ is a zero-mean Gaussian noise term with variance Σ. Typically in logistic regression this noise term is ignored, although it has seen use to model label noise (Collier et al., 2021). We include it to unify the presentation, and the variance can be assumed zero as necessary.

As in the regression case, we specify a Gaussian prior for W. In contrast with the regression setting, exact inference and computation of the posterior predictive is not analytically tractable in this model. We refer to this model—consisting of multinominal Bayesian logistic regression on learned neural network features—as discriminative classification, as logistic regression is a classical discriminative learning algorithm.

2.3. Generative Classification

The second classification model we consider is the *generative classification* model, so-called due to its similarity to classical generative models such as Gaussian discriminant analysis. We again assume deterministic input features parameterized by a neural network, which we write as $\boldsymbol{\phi}_t$. We will assume that in this feature space, the features associated with each class

1. We present a scalar output version of regression, and defer the multivariate output case to the appendix.
2. We discuss only scalar-output regression in the body of the paper; we defer details on the multivariate case to the appendix.

are normally distributed. Placing a normal prior on the means of these feature distributions and a (conjugate) Dirichlet prior on class probabilities, we have a model of the form

$$\boldsymbol{\rho} \sim \mathrm{Dir}(\boldsymbol{\alpha}_0) \qquad \bar{\boldsymbol{\phi}}[\boldsymbol{y}_t] \sim \mathcal{N}(\boldsymbol{\mu}_0[\boldsymbol{y}_t], S_0[\boldsymbol{y}_t]) \tag{4}$$

$$\boldsymbol{y}_t \sim \mathrm{Cat}(\boldsymbol{\rho}) \qquad \boldsymbol{\phi}_t \mid \boldsymbol{y}_t \sim \mathcal{N}(\bar{\boldsymbol{\phi}}[\boldsymbol{y}_t], \Sigma). \tag{5}$$

In this model, $\boldsymbol{\mu}_0[\boldsymbol{y}_t] \in \mathbb{R}^{N_\phi}$ and $S_0[\boldsymbol{y}_t] \in \mathbb{R}^{N_\phi \times N_\phi}$ are the prior mean and covariance over $\bar{\boldsymbol{\phi}}[\boldsymbol{y}_t] \in \mathbb{R}^{N_\phi}$, the mean embedding for each class[3]. We use the square bracket notation to index the statistics for each class; thus, we parameterize $\boldsymbol{\mu}_0[k]$ and $S_0[k]$ for $k = 1, \ldots, N_y$. The terms $\boldsymbol{\rho} \in \mathcal{P}_{N_y}$ correspond to class probabilities, and where $\mathcal{P}_{N_y}$ denotes the probability simplex embedded in $\mathbb{R}^{N_y}$. These class probabilities are in turn used in the categorical distribution over the class. We will write $\bar{\boldsymbol{\phi}} := \{\bar{\boldsymbol{\phi}}[1], \ldots, \bar{\boldsymbol{\phi}}[N_y]\}$.

For a distribution over model parameters

$$p(\boldsymbol{\rho}, \bar{\boldsymbol{\phi}} \mid \boldsymbol{\eta}) = \mathrm{Dir}(\boldsymbol{\alpha}) \prod_{k=1}^{N_y} \mathcal{N}(\boldsymbol{\mu}[k], S[k]) \tag{6}$$

for which we write $\boldsymbol{\eta} = \{\boldsymbol{\alpha}, \boldsymbol{\mu}, S\}$, we have

$$p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{\eta}) = \mathcal{N}(\boldsymbol{\mu}[\boldsymbol{y}], C[\boldsymbol{y}]), \qquad p(\boldsymbol{y} = k \mid \boldsymbol{\eta}) = \frac{\boldsymbol{\alpha}[k]}{\sum_{k'=1}^{N_y} \boldsymbol{\alpha}[k']} \tag{7}$$

via analytical marginalization, and where $C[\boldsymbol{y}] := \Sigma + S[\boldsymbol{y}]$. To compute the predictive over class labels, we apply Bayes' rule, yielding

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\eta}) = \mathrm{softmax}_{\boldsymbol{y}}(\log p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{\eta}) + \log p(\boldsymbol{y} \mid \boldsymbol{\eta})). \tag{8}$$

Here,

$$\log p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{\eta}) = -\frac{1}{2}((\boldsymbol{\phi} - \boldsymbol{\mu}[\boldsymbol{y}])^\top C[\boldsymbol{y}]^{-1}(\boldsymbol{\phi} - \boldsymbol{\mu}[\boldsymbol{y}]) + \mathrm{logdet} C[\boldsymbol{y}] + c) \tag{9}$$

where c is a constant, shared for all classes, that may be ignored due to the shift-invariance of the softmax. Grouping the log determinant term with the class prior yields a bias term. Instead of a linear transformation of the input features to obtain a class logit, we instead have a quadratic transformation. This formulation generalizes standard classifier architectures (Harrison, 2021), in which we have quadratic decision regions as opposed to linear ones.

2.4. Inference and Training in BLL Models

BLL models have seen growing popularity in recent years, ironically driven in part by a need for compatibility with increasingly deep models (Snoek et al., 2015; Azizzadenesheli et al., 2018; Harrison et al., 2018; Weber et al., 2018; Riquelme et al., 2018; Harrison et al., 2020; Ober and Rasmussen, 2019; Kristiadi et al., 2020; Thakur et al., 2020; Watson et al., 2020, 2021; Daxberger et al., 2021a; Willes et al., 2022; Sharma et al., 2022; Schwöbel et al., 2022; Zhang et al., 2021; Moberg et al., 2019; Fiedler and Lucia, 2023). Exact marginalization enables computationally efficient treatment of uncertainty, as well as resulting in lower-variance training objectives compared to sampling-based Bayesian models. The standard

3. With some abuse of notation, the term $\boldsymbol{y}_t$ may refer both to an integer class index and a one-hot encoding, depending on the context.

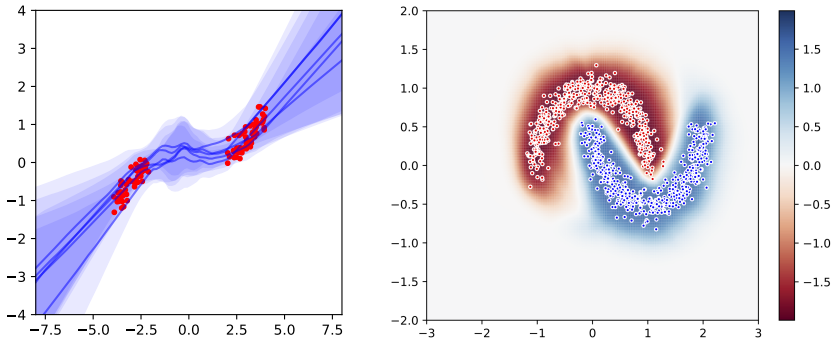

Figure 1: **Left**: a variational BLL (VBLL) regression model with BNN features trained on 50 data points generated from a re-scale cubic with Gaussian noise. The plot shows the 95% predictive credible region under the variational posterior for several sampled feature weights. **Right**: Visualizing (re-scaled) $p(\boldsymbol{x} \mid \boldsymbol{y} = 1) - p(\boldsymbol{x} \mid \boldsymbol{y} = 0)$ predicted by a trained generative VBLL model on the half moon dataset, showing good sensitivity to Euclidean distance and sensible embedding densities.

objective for training in BLL models is the (log) marginal likelihood (Harrison et al., 2018), via gradient descent on

$$\frac{1}{T} \log p(Y \mid X, \boldsymbol{\theta}) \tag{10}$$

where X, Y denote stacked data across t. We include a factor of $1/T$ to enable better comparison with standard, non-Bayesian, training pipelines (typically based on average loss over mini-batches) and across dataset sizes. This training objective can be problematic, however: gradient computation requires computing the full marginal likelihood, and mini-batches do not yield unbiased gradient estimators as in standard training with an arbitrary loss function.

3. Deterministic Stochastic Variational Inference for BLL Networks

To exploit exact marginalization while avoiding full marginal likelihood computation, we will turn to stochastic variational inference (Hoffman et al., 2013). In particular, we aim to jointly compute an approximate last layer posterior and optimize network weights by maximizing lower bounds on marginal likelihood. As such, we will avoid distributional assumptions made in the previous section. We write the last layer parameters as $\boldsymbol{\xi}$ and aim to find an approximate posterior $q(\boldsymbol{\xi} \mid \boldsymbol{\eta})$ parameterized by $\boldsymbol{\eta}$. Concretely, throughout this section we will develop bounds of the form

$$\frac{1}{T} \log p(Y \mid X, \boldsymbol{\theta}) \geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}, \Sigma) - \frac{1}{T} \mathrm{KL}(q(\boldsymbol{\xi} \mid \boldsymbol{\eta}) || p(\boldsymbol{\xi})) \tag{11}$$

where $\mathcal{L}$ is architecture dependent and developed in the remainder of this section.

3.1. Regression

We consider the log marginal likelihood $\log p(Y \mid X, \boldsymbol{\theta})$, with marginalized parameters $\boldsymbol{\xi} = \{\boldsymbol{w}\}$, and have the following lower bound.

Theorem 1 *Let $q(\boldsymbol{\xi} \mid \boldsymbol{\eta}) = \mathcal{N}(\bar{\boldsymbol{w}}, S)$ denote the variational posterior for the BLL model defined in Section 2.1. Then, (11) holds with*

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}, \Sigma) = \frac{1}{T} \sum_{t=1}^{T} \left(\log \mathcal{N}(\boldsymbol{y}_t \mid \bar{\boldsymbol{w}}^\top \boldsymbol{\phi}_t, \Sigma) - \frac{1}{2} \boldsymbol{\phi}_t^\top S \boldsymbol{\phi}_t \Sigma^{-1} \right). \tag{12}$$

The proof for this result and all others is available in Appendix B. When $q(\boldsymbol{\xi} \mid \boldsymbol{\eta}) = p(\boldsymbol{\xi} \mid Y, X)$ and distributional assumptions are satisfied, this lower bound is tight (this may be shown by direct substitution). This correspondence between the variational and true posterior for appropriately-chosen variational families is well known—see (Knoblauch et al., 2019) for a thorough discussion. We note that a similar objective for regression models was developed in (Watson et al., 2021).

3.2. Discriminative Classification

In the discriminative classification case, the parameters are $\boldsymbol{\xi} = \{W\}$. We will assume a diagonal covariance matrix Σ, and write $\sigma_i^2 := \Sigma_{ii}$. We will fix a mean field variational posterior of the form $q(W \mid \boldsymbol{\eta}) = \prod_{k=1}^{N_y} q(\boldsymbol{w}[k] \mid \boldsymbol{\eta}) = \prod_{k=1}^{N_y} q(\bar{\boldsymbol{w}}[k], S[k])$, where $\boldsymbol{w}[k]$ denotes the k'th row of W. This factorization retains dense covariances for each class, but sacrifices cross-class covariances. While we only present this factorized variational posterior, a similar training objective may be derived with a fully dense variational posterior. Under the mean field variational posterior, we have the following bound on the marginal likelihood.

Theorem 2 *Let $q(W \mid \boldsymbol{\eta}) = \prod_{k=1}^{N_y} \mathcal{N}(\bar{\boldsymbol{w}}[k], S[k])$ denote the variational posterior for the discriminative classification model defined in Section 2.2. Then, (11) holds with*

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}, \Sigma) = \frac{1}{T} \sum_{t=1}^{T} \left(\boldsymbol{y}_t^\top \bar{W} \boldsymbol{\phi}_t - \mathrm{LSE}_k \left[\bar{\boldsymbol{w}}[k]^\top \boldsymbol{\phi}_t + \frac{1}{2} (\boldsymbol{\phi}_t^\top S[k] \boldsymbol{\phi}_t + \sigma_k^2) \right] \right) \tag{13}$$

Here, $\mathrm{LSE}(\cdot)$ denotes the log-sum-exp function. In contrast to the regression case, this lower bound is a lower bound on the standard ELBO (due to two applications of Jensen's inequality) and the bound is not tight. We have reduced variance (which would be induced by sampling logit values before the softmax in standard SVI (Ovadia et al., 2019)) for bias due to this lower bound. Our proof leverages the same double application of Jensen's inequality used by (Blei and Lafferty, 2007). We note that tighter analytically tractable lower bounds exist for the logistic regression model (Depraetere and Vandebroek, 2017; Knowles and Minka, 2011), although for simplicity of the resulting algorithm we use the above lower bound.

3.3. Generative Classification

In the generative classification case, the parameters are $\boldsymbol{\xi} = \{\bar{\boldsymbol{\phi}}, \boldsymbol{\rho}\}$. In this setting, the Dirichlet posterior over class probabilities $p(\boldsymbol{\rho} \mid Y)$ can be computed exactly with one pass over the data by simply counting class occurrences. We therefore only consider a variational posterior of the form $q(\boldsymbol{\xi} \mid \boldsymbol{\eta}, Y) = q(\bar{\boldsymbol{\phi}} \mid \boldsymbol{\eta})$ for the class embeddings, where $q(\bar{\boldsymbol{\phi}} \mid \boldsymbol{\eta}) = \prod_{k=1}^{N_y} \mathcal{N}(\boldsymbol{\mu}[k], S[k])$. This yields the following lower bound.

Theorem 3 *Let $q(\bar{\boldsymbol{\phi}} \mid \boldsymbol{\eta}) = \prod_{k=1}^{N_y} \mathcal{N}(\boldsymbol{\mu}[k], S[k])$ denote the variational posterior over class embeddings for the generative classification model defined in Section 2.3. Let $p(\boldsymbol{\rho} \mid Y) =$*

	Accuracy (↑)	ECE (↓)	NLL (↓)	SVHN AUC (↑)	CIFAR-100 AUC (↑)
DNN	95.8 ± 0.19	0.028 ± 0.028	0.183 ± 0.007	0.946 ± 0.005	0.893 ± 0.001
SNGP	95.7 ± 0.14	0.017 ± 0.003	$\mathbf{0.149 \pm 0.005}$	0.960 ± 0.004	$\mathbf{0.902 \pm 0.003}$
D-VBLL	$\mathbf{96.4 \pm 0.12}$	0.022 ± 0.001	0.160 ± 0.001	$\mathbf{0.969 \pm 0.004}$	$\mathbf{0.900 \pm 0.004}$
G-VBLL	$\mathbf{96.3 \pm 0.06}$	0.021 ± 0.001	0.174 ± 0.002	0.925 ± 0.015	0.804 ± 0.006
G-VBLL-MAP	–	–	–	0.950 ± 0.006	0.893 ± 0.003
BNN	96.0 ± 0.08	0.033 ± 0.001	0.333 ± 0.014	0.957 ± 0.004	0.844 ± 0.013
D-VBLL BNN	95.9 ± 0.15	0.058 ± 0.019	0.238 ± 0.036	0.832 ± 0.026	0.744 ± 0.010
G-VBLL BNN	95.9 ± 0.16	$\mathbf{0.009 \pm 0.001}$	0.229 ± 0.010	0.917 ± 0.005	0.779 ± 0.009

Table 1: Results for Wide ResNet-28-10 on CIFAR-10.

$Dir(\boldsymbol{\alpha}_T)$ denote the exact Dirichlet posterior over class probabilities, with $\boldsymbol{\alpha}_T$ denoting the Dirichlet posterior concentration parameters. Then, (11) holds with

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}, \Sigma) = \frac{1}{T}\sum_{t=1}^{T}\Big(\log \mathcal{N}(\boldsymbol{\phi}_t \mid \boldsymbol{\mu}[\boldsymbol{y}_t], \Sigma) - \frac{1}{2}\mathrm{tr}(\Sigma^{-1}S[\boldsymbol{y}_t]) + \log \boldsymbol{\alpha}_T[\boldsymbol{y}_t] \\ - \mathrm{LSE}_k[\log \mathcal{N}(\boldsymbol{\phi}_t \mid \boldsymbol{\mu}[k], \Sigma + S[k]) + \log \boldsymbol{\alpha}_T[k]]\Big) \tag{14}$$

This training objective is again a lower bound on the ELBO, and is not tight. The first Dirichlet term (in the upper line) vanishes in gradient computation, but the second term inside the log-sum-exp function does not. In the case that the posterior concentration parameters are equal for all classes (as in the case of a balanced dataset), the concentration parameter can be pulled out of the $\mathrm{LSE}(\cdot)$ (due to the equivariance of log-sum-exp under shifts) and can be ignored.

3.4. Training and Prediction in VBLL Models

We propose two methods to learn the features in VBLL models. First, we can jointly optimize the last layer variational posterior together with MAP estimation of the features. While one may expect this to result in substantial over-concentration for weak feature priors, in practice we observe that stochastic regularization due to mini-batch optimization prevents overconcentration. Throughout this work, we will place simple isotropic zero-mean Gaussian priors on feature weights (yielding weight decay regularization) and a canonical inverse-Wishart prior on Σ. For Gaussian priors (as developed throughout this section) the KL regularization term can be computed in closed form. Second, we can combine last layer SVI with variational feature learning (Blundell et al., 2015), corresponding to approximate collapsed VI (Teh et al., 2006). Details on both methods are presented in the appendix.

4. Experiments and Discussion

We investigate all three VBLL models. All results and experimental details are available in the appendix. For regression, we present results on the UCI regression datasets (Dua and Graff, 2017), where VBLL models show strong results compared to both single-pass and multi-pass baselines. For classification, we present results on CIFAR-10 and CIFAR-100. CIFAR-10 results are presented in Table 1, above. Both the generative and discriminative models perform well, including compared to SNGP (Liu et al., 2022), which is an established and effective single-pass last layer uncertainty quantification method. Overall, VBLL methods provide consistent improvements to predictive error (as measured through accuracy or MSE) and calibration, while the cost associated with their use is near-zero for even reasonably sized neural networks.

References

Laurence Aitchison. A statistical theory of cold posteriors in deep neural networks. *arXiv preprint arXiv:2008.05912*, 2020.

Kamyar Azizzadenesheli, Emma Brunskill, and Animashree Anandkumar. Efficient exploration through Bayesian deep q-networks. *arXiv:1802.04412*, 2018.

David Blackwell. Conditional expectation and unbiased sequential estimation. *The Annals of Mathematical Statistics*, 1947.

David M Blei and John D Lafferty. A correlated topic model of science. *The annals of applied statistics*, 2007.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning (ICML)*, 2015.

George EP Box and George C Tiao. *Bayesian inference in statistical analysis*, volume 40. John Wiley & Sons, 2011.

Mark Collier, Basil Mustafa, Efi Kokiopoulou, Rodolphe Jenatton, and Jesse Berent. Correlated input-dependent label noise in large-scale image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux-effortless Bayesian deep learning. *Neural Information Processing Systems (NeurIPS)*, 2021a.

Erik Daxberger, Eric Nalisnick, James U Allingham, Javier Antorán, and José Miguel Hernández-Lobato. Bayesian deep learning via subnetwork inference. In *International Conference on Machine Learning (ICML)*, pages 2510–2521. PMLR, 2021b.

Nicolas Depraetere and Martina Vandebroek. A comparison of variational approximations for fast inference in mixed logit models. *Computational Statistics*, 2017.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Michael Dusenberry, Ghassen Jerfel, Yeming Wen, Yian Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. Efficient and scalable bayesian neural nets with rank-1 factors. In *International Conference on Machine Learning (ICML)*, 2020.

Sebastian Farquhar, Michael A Osborne, and Yarin Gal. Radial bayesian neural networks: Beyond discrete support in large-scale bayesian deep learning. In *Artificial Intelligence and Statistics (AISTATS)*, 2020.

Felix Fiedler and Sergio Lucia. Improved uncertainty quantification for neural networks with bayesian last layer. *arXiv preprint arXiv:2302.10975*, 2023.

Andrew YK Foong, Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. 'in-between' uncertainty in bayesian neural networks. *arXiv preprint arXiv:1906.11537*, 2019.

Vincent Fortuin. Priors in bayesian deep learning: A review. *International Statistical Review*, 2022.

Vincent Fortuin, Adrià Garriga-Alonso, Sebastian W Ober, Florian Wenzel, Gunnar Rätsch, Richard E Turner, Mark van der Wilk, and Laurence Aitchison. Bayesian neural network priors revisited. *arXiv:2102.06571*, 2021.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, 2016.

Seymour Geisser. Bayesian estimation in multivariate analysis. *The Annals of Mathematical Statistics*, 1965.

Ryan Giordano, Tamara Broderick, and Michael I Jordan. Covariances, robustness and variational bayes. *Journal of Machine Learning Research*, 2018.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations (ICLR)*, 2020.

Peter Grünwald and Thijs Van Ommen. Inconsistency of bayesian inference for misspecified linear models, and a proposal for repairing it. *Bayesian Analysis*, 2017.

James Harrison. *Uncertainty and efficiency in adaptive robot learning and control*. PhD thesis, Stanford University, 2021.

James Harrison, Apoorva Sharma, and Marco Pavone. Meta-learning priors for efficient online Bayesian regression. *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2018.

James Harrison, Apoorva Sharma, Chelsea Finn, and Marco Pavone. Continuous meta-learning without tasks. *Neural Information Processing Systems (NeurIPS)*, 2020.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv:1610.02136*, 2016.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *Uncertainty in Artificial Intelligence (UAI)*, 2013.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.

Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are bayesian neural network posteriors really like? In *International Conference on Machine Learning (ICML)*, 2021.

David Jacobson. Optimal stochastic linear systems with exponential performance criteria and their relation to deterministic differential games. *IEEE Transactions on Automatic Control*, 1973.

Sanyam Kapoor, Wesley J Maddox, Pavel Izmailov, and Andrew G Wilson. On uncertainty, tempering, and data augmentation in bayesian classification. In *Neural Information Processing Systems (NeurIPS)*, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations (ICLR)*, 2014.

Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. Generalized variational inference: Three arguments for deriving new posteriors. *arXiv preprint arXiv:1904.02063*, 2019.

David Knowles and Tom Minka. Non-conjugate variational message passing for multinomial and binary regression. In *Neural Information Processing Systems (NeurIPS)*, 2011.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being Bayesian, even just a bit, fixes overconfidence in relu networks. In *International Conference on Machine Learning (ICML)*, 2020.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Learnable uncertainty under laplace approximations. In *Uncertainty in Artificial Intelligence (UAI)*, 2021.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Neural Information Processing Systems (NeurIPS)*, 2017.

Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai. When gaussian process meets big data: A review of scalable gps. *IEEE transactions on neural networks and learning systems*, 2020.

Jeremiah Zhe Liu, Shreyas Padhy, Jie Ren, Zi Lin, Yeming Wen, Ghassen Jerfel, Zack Nado, Jasper Snoek, Dustin Tran, and Balaji Lakshminarayanan. A simple approach to improve single-model deep uncertainty via distance-awareness. *arXiv preprint arXiv:2205.00403*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv:1711.05101*, 2017.

David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 1992.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

John Moberg, Lennart Svensson, Juliano Pinto, and Henk Wymeersch. Bayesian linear regression on deep representations. *arXiv:1912.06760*, 2019.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 2020.

Eric Thomas Nalisnick. *On priors for Bayesian neural networks*. University of California, Irvine, 2018.

Radford M Neal. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1995.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Sebastian W Ober and Carl Edward Rasmussen. Benchmarking the neural linear model for regression. *arXiv preprint arXiv:1912.08416*, 2019.

Sebastian W Ober, Carl E Rasmussen, and Mark van der Wilk. The promises and pitfalls of deep kernel learning. In *Uncertainty in Artificial Intelligence (UAI)*, 2021.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Neural Information Processing Systems (NeurIPS)*, 2019.

Theodore Papamarkou, Jacob Hinkle, M Todd Young, and David Womble. Challenges in markov chain monte carlo for bayesian neural networks. *Statistical Science*, 2022.

Tim Pearce, Russell Tsuchida, Mohamed Zaki, Alexandra Brintrup, and Andy Neely. Expressive priors in Bayesian neural networks: Kernel combinations and periodic functions. In *Uncertainty in Artificial Intelligence (UAI)*, 2020.

C Radhakrishna Rao. Information and the accuracy attainable in the estimation of statistical parameters. In *Breakthroughs in statistics*. Springer, 1992.

Carl Edward Rasmussen. *Gaussian processes in machine learning*. Springer, 2004.

Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *Neural Information Processing Systems (NeurIPS)*, 2019.

Jie Ren, Stanislav Fort, Jeremiah Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. A simple fix to mahalanobis distance for improving near-ood detection. *arXiv:2106.09022*, 2021.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep Bayesian bandits showdown. In *International Conference on Learning Representations (ICLR)*, 2018.

Pola Schwöbel, Martin Jørgensen, Sebastian W Ober, and Mark Van Der Wilk. Last layer marginal likelihood for invariance learning. In *Artificial Intelligence and Statistics (AISTATS)*, 2022.

Mrinank Sharma, Sebastian Farquhar, Eric Nalisnick, and Tom Rainforth. Do Bayesian neural networks need to be fully stochastic? *arXiv preprint arXiv:2211.06291*, 2022.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Neural Information Processing Systems (NeurIPS)*, 2017.

Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Prabhat, and Ryan Adams. Scalable Bayesian optimization using deep neural networks. *International Conference on Machine Learning (ICML)*, 2015.

Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger Grosse. Functional variational bayesian neural networks. *arXiv:1903.05779*, 2019.

Yee Teh, David Newman, and Max Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *Neural Information Processing Systems (NeurIPS)*, 2006.

Sujay Thakur, Cooper Lorsung, Yaniv Yacoby, Finale Doshi-Velez, and Weiwei Pan. Uncertainty-aware (una) bases for Bayesian regression using multi-headed auxiliary networks. *arXiv preprint arXiv:2006.11695*, 2020.

George C Tiao and Arnold Zellner. On the bayesian estimation of multivariate regression. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1964.

Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial Intelligence and Statistics (AISTATS)*, 2009.

Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning (ICML)*, 2020.

Yixin Wang and David Blei. Variational bayes under model misspecification. *Neural Information Processing Systems (NeurIPS)*, 2019.

Joe Watson, Jihao Andreas Lin, Pascal Klink, and Jan Peters. Neural linear models with functional gaussian process priors. In *Advances in Approximate Bayesian Inference (AABI)*, 2020.

Joe Watson, Jihao Andreas Lin, Pascal Klink, Joni Pajarinen, and Jan Peters. Latent derivative Bayesian last layer networks. In *Artificial Intelligence and Statistics (AISTATS)*, 2021.

Noah Weber, Janez Starc, Arpit Mittal, Roi Blanco, and Lluís Màrquez. Optimizing over a Bayesian last layer. In *NeurIPS workshop on Bayesian Deep Learning*, 2018.

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? In *International Conference on Machine Learning (ICML)*, 2020.

John Willes, James Harrison, Ali Harakeh, Chelsea Finn, Marco Pavone, and Steven Waslander. Bayesian embeddings for few-shot open world recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2022.

Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In *Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Andrew G Wilson, Zhiting Hu, Russ R Salakhutdinov, and Eric P Xing. Stochastic variational deep kernel learning. In *Neural Information Processing Systems (NeurIPS)*, 2016a.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial Intelligence and Statistics (AISTATS)*, 2016b.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Xueting Zhang, Debin Meng, Henry Gouk, and Timothy M Hospedales. Shallow Bayesian meta learning for real-world few-shot recognition. In *International Conference on Computer Vision (ECCV)*, 2021.

Appendix A. The Multivariate Regression Model

In the multivariate regression case, we consider a model of the form

$$\boldsymbol{y}_t = W\boldsymbol{\phi}_t + \boldsymbol{\varepsilon}_t \tag{15}$$

and place a matrix normal (Tiao and Zellner, 1964; Geisser, 1965) prior on W, with $W \sim \mathcal{MN}(\bar{W}_0, I, S_0)$. For a discussion of the matrix normal distribution, we refer the reader to (Box and Tiao, 2011).

Given the matrix normal prior and the above model, the posterior is also matrix normal. We thus fix a matrix normal variational posterior. In Appendix B.2, we obtain an ELBO of the form

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}, \Sigma) = \frac{1}{T}\sum_{t=1}^{T}\left(\log \mathcal{N}(\boldsymbol{y}_t \mid \bar{W}\boldsymbol{\phi}_t, \Sigma) - \frac{1}{2}\boldsymbol{\phi}_t^\top S \boldsymbol{\phi}_t \mathrm{tr}(\Sigma^{-1})\right). \tag{16}$$

for $\boldsymbol{\eta} = \{\bar{W}, S\}$, and we use this as a training objective.

For a parameter distribution $\mathcal{MN}(\bar{W}, I, S)$, prediction in this model is analytically tractable and is

$$p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{\eta}, \boldsymbol{\theta}) = \mathcal{N}(\bar{W}\boldsymbol{\phi}_t, \boldsymbol{\phi}_t^\top S \boldsymbol{\phi}_t I + \Sigma). \tag{17}$$

Appendix B. Proofs and Further Theoretical Results

B.1. Helper Results

Our first result builds on results from the variational Gaussian process literature (Titsias, 2009; Hensman et al., 2013).

Lemma 4 *Let $q(\boldsymbol{\mu}) = \mathcal{N}(\bar{\boldsymbol{\mu}}, S)$ and $p(\boldsymbol{y} \mid X, \boldsymbol{\mu}) = \mathcal{N}(X\boldsymbol{\mu}, \Sigma)$ with $\boldsymbol{y} \in \mathbb{R}^N, \bar{\boldsymbol{\mu}}, \boldsymbol{\mu} \in \mathbb{R}^M$, $X \in \mathbb{R}^{N\times M}$, and $S, \Sigma \in \mathbb{R}^{M\times M}$. Then*

$$\mathbb{E}_{q(\boldsymbol{\mu})}[\log p(\boldsymbol{y} \mid X, \boldsymbol{\mu})] = \log p(\boldsymbol{y} \mid X, \bar{\boldsymbol{\mu}}) - \frac{1}{2}\mathrm{tr}(\Sigma^{-1} X S X^\top). \tag{18}$$

Proof We have

$$\begin{aligned}
\mathbb{E}_{q(\boldsymbol{\mu})}[\log p(\boldsymbol{y} \mid X\boldsymbol{\mu})] &= -\frac{1}{2}\mathbb{E}_{q(\boldsymbol{\mu})}[c + \mathrm{logdet}(2\pi\Sigma) + (\boldsymbol{y} - X\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{y} - X\boldsymbol{\mu})] && (19)\\
&= -\frac{1}{2}\left(c + \mathrm{logdet}(2\pi\Sigma) + \mathbb{E}_{q(\boldsymbol{\mu})}[(\boldsymbol{y} - X\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{y} - X\boldsymbol{\mu})]\right) && (20)\\
&= -\frac{1}{2}\left(c + \mathrm{logdet}(2\pi\Sigma) + (\boldsymbol{y} - X\bar{\boldsymbol{\mu}})^\top \Sigma^{-1}(\boldsymbol{y} - X\bar{\boldsymbol{\mu}}) + \mathrm{tr}(\Sigma^{-1} X S X^\top)\right) && (21)
\end{aligned}$$

where $c \in \mathbb{R}$ is a constant and where the last line follows from the fact that $\boldsymbol{y} - X\boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{y} - X\bar{\boldsymbol{\mu}}, XSX^\top)$. The first two terms form the desired log density. ■

Based on this result, we can state a straightforward corollary for generative classification.

Corollary 5 *Let $q(\boldsymbol{\mu}) = \mathcal{N}(\bar{\boldsymbol{\mu}}, S)$ and $p(\boldsymbol{y} \mid \boldsymbol{\mu}) = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ with $\boldsymbol{y}, \bar{\boldsymbol{\mu}}, \boldsymbol{\mu} \in \mathbb{R}^N$, $S, \Sigma \in \mathbb{R}^{N\times N}$. Then*

$$\mathbb{E}_{q(\boldsymbol{\mu})}[\log p(\boldsymbol{y} \mid \boldsymbol{\mu})] = \log p(\boldsymbol{y} \mid \bar{\boldsymbol{\mu}}) - \frac{1}{2}\mathrm{tr}(\Sigma^{-1} S). \tag{22}$$

Proof This result follows from Lemma 4 by simply choosing $X = I$. ■

We can also present a variant for multivariate classification.

Corollary 6 *Let* $q(W) = \mathcal{MN}(\bar{W}, I, S)$ *and* $p(\boldsymbol{y} \mid \boldsymbol{x}, W) = \mathcal{N}(W\boldsymbol{x}, \Sigma)$ *with* $\boldsymbol{y} \in \mathbb{R}^M, \bar{W}, W \in \mathbb{R}^{M\times N}$*;* $x \in \mathbb{R}^N$*;* $S \in \mathbb{R}^{N\times N}$*; and* $\Sigma \in \mathbb{R}^{M\times M}$*. Then*

$$\mathbb{E}_{q(W)}[\log p(\boldsymbol{y} \mid \boldsymbol{x}, W)] = \log p(\boldsymbol{y} \mid \boldsymbol{x}, \bar{W}) - \frac{1}{2}\boldsymbol{x}^\top S\boldsymbol{x}\,\mathrm{tr}(\Sigma^{-1}). \tag{23}$$

Proof Our proof closely follows that of Lemma 4. Expanding the likelihood in the expectation, we have

$$\mathbb{E}_{q(W)}[\log p(\boldsymbol{y} \mid \boldsymbol{x}, W)] = \log p(\boldsymbol{y} \mid \boldsymbol{x}, \bar{W}) - \frac{1}{2}\mathbb{E}_W[\boldsymbol{x}^\top (W - \bar{W})^\top \Sigma^{-1}(W - \bar{W})\boldsymbol{x}] \tag{24}$$

Leveraging the matrix normal identity

$$\mathbb{E}_{W\sim\mathcal{MN}(\bar{W},V,U)}[W^\top AW] = U\mathrm{tr}(A^\top V) + \bar{W}^\top A\bar{W} \tag{25}$$

and the fact that $W - \bar{W} \sim \mathcal{MN}(0, I, S)$, we have

$$\mathbb{E}[(W - \bar{W})^\top \Sigma^{-1}(W - \bar{W})] = S\mathrm{tr}(\Sigma^{-1}) \tag{26}$$

which completes the proof. ■

Lemma 7 *Let* $p(\boldsymbol{x} \mid \bar{\boldsymbol{x}}) = \mathcal{N}(\bar{\boldsymbol{x}}, \Sigma)$*, and let* $\bar{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{\mu}, S)$*. Then,*

$$\mathbb{E}_{\bar{\boldsymbol{x}}}[p(\boldsymbol{x} \mid \bar{\boldsymbol{x}})] = \mathcal{N}(\boldsymbol{\mu}, \Sigma + S). \tag{27}$$

Proof We build upon (Jacobson, 1973) and note

$$\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},S)}[\exp(-\frac{1}{2}\boldsymbol{x}^\top\Sigma^{-1}\boldsymbol{x})] = \sqrt{\frac{\det(S^{-1})}{\det(S^{-1}+\Sigma^{-1})}}\exp(-\frac{1}{2}\boldsymbol{\mu}^\top S^{-1}(S-(\Sigma^{-1}+S^{-1})^{-1})S^{-1}\boldsymbol{\mu}). \tag{28}$$

Note, by Woodbury's identity

$$S^{-1}(S - (\Sigma^{-1} + S^{-1})^{-1})S^{-1} = (S + \Sigma)^{-1} \tag{29}$$

Let $\boldsymbol{z} := \boldsymbol{x} - \bar{\boldsymbol{x}}$, then $\boldsymbol{z}_t \sim \mathcal{N}(\boldsymbol{x} - \boldsymbol{\mu}, S)$. We then have

$$\mathbb{E}[p(\boldsymbol{x} \mid \bar{\boldsymbol{x}})] = \mathbb{E}[\exp(-\frac{1}{2}\|\boldsymbol{x} - \bar{\boldsymbol{x}}\|^2_{\Sigma^{-1}} + \frac{1}{2}\mathrm{logdet}(2\pi\Sigma^{-1}))] \tag{30}$$

$$= \mathbb{E}[\exp(-\frac{1}{2}\boldsymbol{z}^\top\Sigma^{-1}\boldsymbol{z})]]\exp(\frac{1}{2}\mathrm{logdet}(2\pi\Sigma^{-1})) \tag{31}$$

For the expectation we apply (28). We simplify the determinant term of (28) as

$$\sqrt{\frac{\det(S^{-1})}{\det(S^{-1}+\Sigma^{-1})}} = \exp(-\frac{1}{2}\mathrm{logdet}(I + S\Sigma^{-1})) \tag{32}$$

Combining, we have

$$\mathbb{E}[\exp(-\frac{1}{2}\boldsymbol{z}^\top\Sigma^{-1}\boldsymbol{z})]] = \exp(-\frac{1}{2}(\|\boldsymbol{x}-\boldsymbol{\mu}\|^2_{(S+\Sigma)^{-1}} + \operatorname{logdet}(I+S\Sigma^{-1})) \tag{33}$$

We have two log determinant terms, from (31) and the above. We can combine them as

$$\begin{aligned}\frac{1}{2}\operatorname{logdet}(2\pi\Sigma^{-1}) - \frac{1}{2}\operatorname{logdet}(I+S\Sigma^{-1}) &= -\frac{1}{2}(\operatorname{logdet}(\frac{1}{2\pi}\Sigma) + \operatorname{logdet}(I+S\Sigma^{-1})) &(34)\\ &= -\frac{1}{2}\operatorname{logdet}((\frac{1}{2\pi}\Sigma)(I+S\Sigma^{-1})) &(35)\\ &= -\frac{1}{2}\operatorname{logdet}(\frac{1}{2\pi}\Sigma + \frac{1}{2\pi}S) &(36)\end{aligned}$$

Combining all terms completes the proof. ■

B.2. Proof of Theorem 1

Theorem 1 *Let $q(\boldsymbol{\xi} \mid \boldsymbol{\eta}) = \mathcal{N}(\bar{\boldsymbol{w}}, S)$ denote the variational posterior for the BLL model defined in Section 2.1. Then, (11) holds with*

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}, \Sigma) = \frac{1}{T}\sum_{t=1}^{T}\left(\log\mathcal{N}(\boldsymbol{y}_t \mid \bar{\boldsymbol{w}}^\top\boldsymbol{\phi}_t, \Sigma) - \frac{1}{2}\boldsymbol{\phi}_t^\top S\boldsymbol{\phi}_t\Sigma^{-1}\right). \tag{37}$$

Proof First,

$$\begin{aligned}\log p(Y \mid X, \boldsymbol{\theta}) &= \log\mathbb{E}_{p(\boldsymbol{\xi})}[p(Y \mid X, \boldsymbol{\xi}, \boldsymbol{\theta})] &(38)\\ &= \log\mathbb{E}_{q(\boldsymbol{\xi}\mid\boldsymbol{\eta})}[p(Y \mid X, \boldsymbol{\xi}, \boldsymbol{\theta})\frac{p(\boldsymbol{\xi}}{q(\boldsymbol{\xi} \mid \boldsymbol{\eta})}] &(39)\\ &\geq \mathbb{E}_{q(\boldsymbol{\xi}\mid\boldsymbol{\eta})}[\log p(Y \mid X, \boldsymbol{\xi}, \boldsymbol{\theta})] - \mathrm{KL}(q(\boldsymbol{\xi} \mid \boldsymbol{\eta})||p(\boldsymbol{\xi})) &(40)\\ &= \sum_{t=1}^{T}\mathbb{E}_{q(\boldsymbol{\xi}\mid\boldsymbol{\eta})}[\log p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{\xi}, \boldsymbol{\theta})] - \mathrm{KL}(q(\boldsymbol{\xi} \mid \boldsymbol{\eta})||p(\boldsymbol{\xi})). &(41)\end{aligned}$$

Note that the first term in the last line is the log of a Normal distribution. Applying Lemma 5, we have

$$\mathbb{E}_{q(\boldsymbol{\xi}\mid\boldsymbol{\eta})}[\log p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{\xi}, \boldsymbol{\theta})] = \log p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{\xi}, \boldsymbol{\theta}) - \frac{1}{2}\boldsymbol{\phi}_t^\top S\boldsymbol{\phi}_t\Sigma^{-1} \tag{42}$$

which completes the proof. ■

We can also state the following corollary for the multivariate case.

Corollary 8 *Let $q(\boldsymbol{\xi} \mid \boldsymbol{\eta}) = \mathcal{MN}(\bar{W}, I, S)$ denote the variational posterior for the multivariate BLL model defined in Appendix A. Then, (11) holds with*

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}, \Sigma) = \frac{1}{T}\sum_{t=1}^{T}\left(\log\mathcal{N}(\boldsymbol{y}_t \mid \bar{W}\boldsymbol{\phi}_t, \Sigma) - \frac{1}{2}\boldsymbol{\phi}_t^\top S\boldsymbol{\phi}_t\operatorname{tr}(\Sigma^{-1})\right). \tag{43}$$

Proof The proof follows the proof of Theorem 1, applying Corollary 6 instead of Lemma 5. ■

B.3. Proof of Theorem 2

Theorem 2 *Let* $q(W \mid \boldsymbol{\eta}) = \prod_{k=1}^{N_y} \mathcal{N}(\bar{\boldsymbol{w}}[k], S[k])$ *denote the variational posterior for the discriminative classification model defined in Section 2.2. Then,* (11) *holds with*

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}, \Sigma) = \frac{1}{T} \sum_{t=1}^{T} \left(\boldsymbol{y}_t^\top \bar{W} \boldsymbol{\phi}_t - \mathrm{LSE}_k \left[\bar{\boldsymbol{w}}[k]^\top \boldsymbol{\phi}_t + \frac{1}{2} (\boldsymbol{\phi}_t^\top S[k] \boldsymbol{\phi}_t + \sigma_k^2) \right] \right) \tag{44}$$

Proof We construct an ELBO via

$$\begin{aligned} \log p(Y \mid X, \boldsymbol{\theta}) &= \log \mathbb{E}_{p(\boldsymbol{\xi})}[p(Y \mid X, \boldsymbol{\theta}, \boldsymbol{\xi})] && (45) \\ &\geq \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta})}[\log p(Y \mid X, \boldsymbol{\theta}, \boldsymbol{\xi})] - \mathrm{KL}(q(\boldsymbol{\xi} \mid \boldsymbol{\eta}) || p(\boldsymbol{\xi})) && (46) \\ &= \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta})}[\sum_{t=1}^{T} \log \mathrm{softmax}_y(\log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\theta}, \boldsymbol{\xi}))] - \mathrm{KL}(q(\boldsymbol{\xi} \mid \boldsymbol{\eta}) || p(\boldsymbol{\xi})) && (47) \end{aligned}$$

Expanding the log-softmax term, we have

$$\begin{aligned} &\mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta})} \left[\sum_t \log \mathrm{softmax}_{\boldsymbol{y}_t}(\log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\theta}, \boldsymbol{\xi})) \right] = && (48) \\ &\sum_t \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta})}[\log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\theta}, \boldsymbol{\xi}))] - \sum_t \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta})}[\mathrm{LSE}_{\boldsymbol{y}_t}[\log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\theta}, \boldsymbol{\xi})]. \end{aligned}$$

As previously, under the variational posterior these likelihoods factorize across the data. The first term may be directly evaluated, yielding

$$\mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta})}[\log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\theta}, \boldsymbol{\xi}))] = \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta})}[\boldsymbol{w}[\boldsymbol{y}_t]^\top] \boldsymbol{\phi}_t = \bar{\boldsymbol{w}}[\boldsymbol{y}_t]^\top \boldsymbol{\phi}_t. \tag{49}$$

The second term (containing the log-sum-exp) can not be computed exactly, and so we will bound this term for both the discriminative and generative classifiers. Via Jensen's inequality, we have

$$- \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta})}[\mathrm{LSE}_{\boldsymbol{y}_t}[\log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\theta}, \boldsymbol{\xi})] \geq - \log \sum_{\boldsymbol{y}_t} \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta})}[\exp(\log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\theta}, \boldsymbol{\xi}))] \tag{50}$$

In the case of the discriminative model, we follow (Blei and Lafferty, 2007) and note that for each row i

$$\mathbb{E}_{\boldsymbol{w}_i \sim \mathcal{N}(\bar{\boldsymbol{w}}_i, S_i)}[\exp(\boldsymbol{w}_i^\top \boldsymbol{\phi}_t + \boldsymbol{\varepsilon}_t[i])] = \exp(\bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi}_t + \frac{1}{2} (\boldsymbol{\phi}_t^\top S_i \boldsymbol{\phi}_t + \sigma_i^2)) \tag{51}$$

which relies on assumed independence of rows of W (although relaxation of this assumption is possible). Combining these results yields a lower bound on the ELBO, which is itself a lower bound on the marginal likelihood. ■

B.4. Proof of Theorem 3

Theorem 3 *Let $q(\bar{\boldsymbol{\phi}} \mid \boldsymbol{\eta}) = \prod_{k=1}^{N_y} \mathcal{N}(\boldsymbol{\mu}[k], S[k])$ denote the variational posterior over class embeddings for the generative classification model defined in Section 2.3. Let $p(\boldsymbol{\rho} \mid Y) = Dir(\boldsymbol{\alpha}_T)$ denote the exact Dirichlet posterior over class probabilities, with $\boldsymbol{\alpha}_T$ denoting the Dirichlet posterior concentration parameters. Then, (11) holds with*

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}, \Sigma) = \frac{1}{T}\sum_{t=1}^{T}\big(\log \mathcal{N}(\boldsymbol{\phi}_t \mid \boldsymbol{\mu}[\boldsymbol{y}_t], \Sigma) - \frac{1}{2}\mathrm{tr}(\Sigma^{-1}S[\boldsymbol{y}_t]) + \log \boldsymbol{\alpha}_T[\boldsymbol{y}_t] \\ - \mathrm{LSE}_k[\log \mathcal{N}(\boldsymbol{\phi}_t \mid \boldsymbol{\mu}[k], \Sigma + S[k]) + \log \boldsymbol{\alpha}_T[k]]\big) \tag{52}$$

Proof Note that

$$\log p(Y \mid X, \boldsymbol{\theta}) \geq \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta})}[\log p(Y \mid X, \boldsymbol{\xi}, \boldsymbol{\theta})] - \mathrm{KL}(q(\boldsymbol{\xi} \mid \boldsymbol{\eta})||p(\boldsymbol{\xi})) \tag{53}$$

where

$$\mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta})}[\log p(Y \mid X, \boldsymbol{\xi}, \boldsymbol{\theta})] = \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta})}\left[\log p(X \mid Y, \boldsymbol{\theta}, \boldsymbol{\xi}) - \log p(X \mid \boldsymbol{\theta}, \boldsymbol{\xi})\right] + \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta})}[\log p(Y \mid \boldsymbol{\xi})] \tag{54}$$

All of these terms factorize over the data, as previously. We first note that for the last term,

$$\log \mathbb{E}_{\boldsymbol{\rho}}[p(\boldsymbol{y}_t \mid \boldsymbol{\theta}, \boldsymbol{\rho})] = \log \boldsymbol{\alpha}[\boldsymbol{y}_t] - \log \sum_{\boldsymbol{y}} \boldsymbol{\alpha}[\boldsymbol{y}]. \tag{55}$$

via standard Dirichlet-Categorical marginalization, and where $\boldsymbol{\alpha}$ correspond to posterior Dirichlet concentration parameters. The first term in (54) is the embedding likelihood; we can compute this expectation of the log likelihood via Corollary 5.

The second term in (54) is less straight-forward. Note that

$$\mathbb{E}[\log p(\boldsymbol{x}_t \mid \boldsymbol{\theta}, \boldsymbol{\xi})] = \mathbb{E}[\log \sum_{\boldsymbol{y}} p(\boldsymbol{x}_t \mid \boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\xi}) p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\xi})] \tag{56}$$

which can be written as a log-sum-exp of log joint likelihood. We will again apply Jensen's to exchange the log and sum, and note

$$-\mathbb{E}[\log p(\boldsymbol{x}_t \mid \boldsymbol{\theta}, \boldsymbol{\xi})] = -\mathbb{E}[\log \sum_{\boldsymbol{y}} p(\boldsymbol{x}_t \mid \boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\xi}) p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\xi})] \tag{57}$$

$$\geq -\log \mathbb{E}[\sum_{\boldsymbol{y}} p(\boldsymbol{x}_t \mid \boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\xi}) p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\xi})] \tag{58}$$

$$= -\log \sum_{\boldsymbol{y}} \mathbb{E}_{\bar{\boldsymbol{\phi}}[\boldsymbol{y}]}[p(\boldsymbol{x}_t \mid \boldsymbol{y}, \boldsymbol{\theta}, \bar{\boldsymbol{\phi}}[\boldsymbol{y}])] \mathbb{E}_{\boldsymbol{\rho}}[p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\rho})] \tag{59}$$

$$= -\mathrm{LSE}_{\boldsymbol{y}}[\log \mathbb{E}_{\boldsymbol{\rho}}[p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\rho})] + \log \mathbb{E}_{\bar{\boldsymbol{\phi}}[\boldsymbol{y}]}[p(\boldsymbol{x}_t \mid \boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\xi})]] \tag{60}$$

where the second line follows from Jensen's, and the third line follows from the structure of the variational posterior. The first term in (60) corresponds to (55). The second term in (55) (the sum over concentration parameters) is equivalent for all classes $\boldsymbol{y}$, and thus can be pulled out of the log-sum-exp (due to the equivariance of this function under shifts) where it cancels the same third term in (54).

To compute the second expectation in (60), we apply Lemma 7. Combining all terms completes the proof. ■

Appendix C. Algorithmic Details

In this section we present concrete details on training VBLL models. We first describe the procedure for MAP estimation and variational learning of features, as described in the paper body. We then discuss prior choice, describe the resultant regularization terms, and describe prediction within these models.

C.1. Feature Point Estimation

We propose to train our models via joint variational inference for the last layer and MAP estimation of network weights (and noise covariance), yielding optimization problem

$$\boldsymbol{\theta}^*, \boldsymbol{\eta}^*, \Sigma^* = \arg\max_{\boldsymbol{\theta}, \boldsymbol{\eta}, \Sigma} \left\{ \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}, \Sigma) + \frac{1}{T}(\log p(\boldsymbol{\theta}) + \log p(\Sigma) - \mathrm{KL}(q(\boldsymbol{\xi} \mid \boldsymbol{\eta}) || p(\boldsymbol{\xi}))) \right\}. \quad (61)$$

We will write the three terms on the RHS (scaled by $1/T$) as $\mathcal{R}(\boldsymbol{\theta}, \boldsymbol{\eta}, \Sigma)$. Reasonable priors for neural network weights have been discussed in several papers (Blundell et al., 2015; Pearce et al., 2020; Fortuin, 2022; Farquhar et al., 2020; Watson et al., 2020; Dusenberry et al., 2020; Nalisnick, 2018). In this work, we use simple isotropic Gaussian priors which yields a weight decay regularizer. We use an inverse-Wishart prior on the noise covariance (or a product of inverse Gamma priors for diagonal covariances). While variational inference for the noise covariance is possible, we choose point estimation to simplify the model. This optimization problem is solved via stochastic gradient descent on mini-batch objectives.

For prediction with VBLL models, we predict directly using the variational posterior, exploiting the conjugate prediction results described in Section 2. For all three VBLL models, training objective computation and prediction can be reduced from cubic to quadratic complexity (in the last layer input width) by careful parameterization and computation. The assumptions required to achieve quadratic complexity for the first two models are minor. However, for the generative classification model, diagonal covariances must be assumed. We discuss complexity in the next section.

C.2. Collapsed Variational Inference for Bayesian Neural Networks

Stochastic variational approximations to the posterior over the network weights have previously been used for Bayesian learning (Blundell et al., 2015). In this section, we aim to compute a variational posterior $q(\boldsymbol{\theta})$, following the same SVI methodology as discussed previously. Whereas our approaches developed in the previous section were deterministic, SVI for all network weights is not possible via deterministic marginalization. Thus, computing

$$\mathbb{E}_{q(\boldsymbol{\theta})}[\log p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})] \quad (62)$$

is typically approximated using Monte Carlo methods. In (Blundell et al., 2015), the authors turn to the reparameterization gradient estimator (Kingma and Welling, 2014; Mohamed et al., 2020) to enable the computation of the (Monte Carlo estimator of the) gradient with respect to the parameters of the variational posterior. We could take a similar strategy for both $\boldsymbol{\xi}$ and $\boldsymbol{\theta}$, turning to sampling-based approximation. However, this sampling scheme yields both noisy gradient estimates and is expensive, as each sample corresponds to a full network evaluation. Our approach will instead marginalize the last layer and sample (some of) the other layers. This corresponds to Rao-Blackwellization (Rao, 1992; Blackwell, 1947) of the variational lower bound estimator, yielding lower variance gradient estimates.

We will choose a mean field posterior that factorizes over the parameters and weights, $q(\boldsymbol{\xi}, \boldsymbol{\theta} \mid \boldsymbol{\eta}) = q(\boldsymbol{\xi} \mid \boldsymbol{\eta}_{\boldsymbol{\xi}}) q(\boldsymbol{\theta} \mid \boldsymbol{\eta}_{\boldsymbol{\theta}})$. We also, in the discussion below, suppress dependence on Σ; in practice, we will turn to point estimation for this term. Note that further mean field approximations for $q(\boldsymbol{\theta} \mid \boldsymbol{\eta})$ are typically employed. For example, (Blundell et al., 2015) factorize the posterior over all weights in the neural network. Given this posterior approximation, we have

$$\log p(Y \mid X) \geq \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\eta}_{\boldsymbol{\theta}}) q(\boldsymbol{\xi}|\boldsymbol{\eta}_{\boldsymbol{\xi}})}[\log p(Y \mid X, \boldsymbol{\xi}, \boldsymbol{\theta})] - \mathrm{KL}(q(\boldsymbol{\xi} \mid \boldsymbol{\eta}_{\boldsymbol{\xi}}) || p(\boldsymbol{\xi})) - \mathrm{KL}(q(\boldsymbol{\theta} \mid \boldsymbol{\eta}_{\boldsymbol{\theta}}) || p(\boldsymbol{\theta})) \tag{63}$$

under the assumption that the prior $p(\boldsymbol{\xi}, \boldsymbol{\theta}) = p(\boldsymbol{\xi}) p(\boldsymbol{\theta})$ and thus

$$\frac{1}{T} \log p(Y \mid X) \geq \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\eta}_{\boldsymbol{\theta}})}[\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}_{\boldsymbol{\xi}})] - \frac{1}{T} \mathrm{KL}(q(\boldsymbol{\xi} \mid \boldsymbol{\eta}_{\boldsymbol{\xi}}) || p(\boldsymbol{\xi})) - \frac{1}{T} \mathrm{KL}(q(\boldsymbol{\theta} \mid \boldsymbol{\eta}_{\boldsymbol{\theta}}) || p(\boldsymbol{\theta})) \tag{64}$$

for the lower bounds $\mathcal{L}$ developed in Section 3. Thus, algorithmically, we first compute the inner expectation and then approximate the outer expectation with a sampling-based estimator.

C.3. Weight Priors

We place a prior on all trainable terms: the (neural network) feature weights as well as the noise covariance Σ. In the regression and generative classification case, it is possible to perform conjugate inference for the noise covariance in addition to means. Due to the algorithmic complexity of this approach, we instead propose to perform MAP estimation of Σ for these models. We use a standard inverse-Wishart prior for the noise covariance. Ignoring terms that vanish in gradient computation, we have likelihood

$$\log p(\Sigma) = \frac{\nu + N + 1}{2} \mathrm{logdet} \Sigma^{-1} - \frac{1}{2} \mathrm{tr}(M \Sigma^{-1}) \tag{65}$$

where Σ is $N \times N$, $\nu > N - 1$ are the degrees of freedom and M is the scale matrix. The terms ν, M are hyperparameters that are fixed.

For the feature weights $\boldsymbol{\theta}$ we will typically use a simple isotropic Gaussian prior, which corresponds to l_2 regularization. Numerous regularization strategies have been proposed for controlling the kernel generated by neural network features (Watson et al., 2021; Fortuin, 2022), and we view this as necessary follow-up work but beyond the scope of this paper.

C.4. Training

We now present our full training approach for the regression and classification settings. Generally, we will minimize the lower bounds we developed for each model. We note that $\mathcal{L}$ is a sum over data; following (Blundell et al., 2015), we compute an (unbiased) estimator $\hat{\mathcal{L}}$ for this term with mini-batches.

The factorization of the ELBO over the data implies a mini-batch estimator for the gradient. Note that

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta})}[\log p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{\xi}, \boldsymbol{\theta})] = \mathbb{E}_t \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta})}[\log p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{\xi}, \boldsymbol{\theta})] \tag{66}$$

where the outer expectation is with respect to a uniform distribution over $t = 1, \ldots, T$. Note that this also holds for lower bound on the data likelihood, in the case of classification. We can construct a randomized estimator for this expectation based on sub-sampling the data, in our case in mini-batches. For a mini-batch of B datapoints, this yields an estimator for the ELBO of the form

$$\hat{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\eta}, \Sigma) = \frac{1}{B} \sum_{t=1}^{B} \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta})}[\log p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{\xi}, \boldsymbol{\theta}, \Sigma)]. \tag{67}$$

In the classification case, this may be an inequality. This re-scaling of the KL divergence was also discussed by (Blundell et al., 2015). Note that in the limit of infinite training data ($T \to \infty$) the weight on the KL term goes to zero.

We have trained VBLL models with both momentum SGD and AdamW (Loshchilov and Hutter, 2017). While both work effectively, they result in different uncertainty representations far from the data. The interaction of VBLLs with the stochastic regularization associated with different optimizers is an important direction of future work. Practically, gradient clipping was necessary to stabilize late training, especially in the regression case. As the noise variance concentrates, gradient magnitude is highly sensitive to small perturbations to features, which can be rapidly destabilizing; gradient clipping was necessary and sufficient to prevent this destabilization. Beyond these details, training VBLL models did not differ from training normal models.

C.5. Hyperparameters

VBLL models introduce a small number of hyperparameters over standard network training. First, standard hyperparameters may need to be modified for VBLL models. For example, we found longer training runs resulted in slightly improved calibration, but we believe further investigation of learning rate schedules is necessary. For MAP features estimation, we use standard weight decay regularization values.

The main novel hyperparameters introduced by the VBLL model are those associated with priors. In particular, the last layer mean prior (defined by a mean and variance; in the regression case, these are written $\bar{\boldsymbol{w}}_0, S_0$) must be chosen. Practically, it is common to normalize outputs to have isotropic Gaussian distributions for regression, and thus we have found $\bar{\boldsymbol{w}}_0 = 0$ and $S_0 = I$ yield a reasonable if diffuse prior. For the classification case, we found these values similarly induce reasonable epistemic uncertainty over the predictive categorical distribution.

The other novel hyperparameters are those associated with the noise covariance inverse-Wishart prior, the degrees of freedom ν and the scale matrix M. For all experiments, we fix the scale matrix as a scalar multiple of the identity matrix, $M = mI$. In our regression experiments we fix these to be $(1, 1)$, and find good resulting performance, but further investigation is possible. In the classification case—and in particular the generative classification case—these parameters control the degree of concentration in the feature space, and thus must be more carefully selected (and often co-selected with the weight decay strength). While we have not established effective rules of thumb for selecting these hyperparameters, we believe such rules are possible to avoid expensive hyperparameter search. Moreover, the performance dependence on these parameters was observed to be fairly weak.

C.6. Prediction and Monitoring

After training, we have learned network weights $\boldsymbol{\theta}^*$ (or a variational posterior over these weights), noise covariance Σ^*, and variational posterior parameters $\boldsymbol{\eta}^*$. To make predictions, there are two options. In the case of the regression and generative classification model, we may discard the variational posterior and leverage exact conjugacy. Under (Gaussian) distributional assumptions, exact posteriors may be computed with fixed features.

Exact posteriors may be badly calibrated due to violation of distributional assumptions. Instead, we may make predictions under the variational posterior directly, under the assumption that $q(\boldsymbol{\xi} \mid \boldsymbol{\eta}^*) \approx p(\boldsymbol{\xi} \mid X, Y)$, yielding

$$p(\boldsymbol{y} \mid \boldsymbol{x}, X, Y) \approx \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta}^*)}[p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{\theta}^*, \Sigma^*)] \tag{68}$$

where $(\boldsymbol{x}, \boldsymbol{y})$ denote a test point. For the discriminative classification model, only prediction under the variational posterior is possible, and in this model, sampling or an approximation (e.g. Laplace) may be used. In the regression and generative classification case, conjugacy with respect to the variational posteriors may be exploited to yield a closed-form predictive density (via (1) and (8)).

The generative classification case provides predicted class probability biases (the predicted probability of seeing a particular class before observing a label) through the Dirichlet posterior. In cases where a system designer believes there is likely to exist distributional shift between the training data and the evaluation conditions, predictions may be directly controlled by modifying this Dirichlet posterior.

For the variational feature approach, prediction can be done by sampling features and computing mixture distributions, yielding

$$p(\boldsymbol{y} \mid \boldsymbol{x}, X, Y) \approx \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\eta}^*)}\mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta}^*)}[p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{\theta}, \Sigma^*)] \tag{69}$$

$$\approx \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta}^*)}[p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{\theta}_k, \Sigma^*)] \tag{70}$$

for $\boldsymbol{\theta}_k$ sampled i.i.d. from the variational posterior. In the regression case, this averaging is straightforward. For the classification cases, we can average pre-softmax or post-softmax. For example, in the case of generative classification, both

$$p(\boldsymbol{y} \mid \boldsymbol{x}, X, Y) \approx \frac{1}{K}\sum_{k=1}^{K} \mathrm{softmax}_{\boldsymbol{y}}(\log p(\boldsymbol{y} \mid X, Y) + \log \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta}^*)}[p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{\xi}, \boldsymbol{\theta}_k)]) \tag{71}$$

and

$$p(\boldsymbol{y} \mid \boldsymbol{x}, X, Y) \approx \mathrm{softmax}_{\boldsymbol{y}}(\log p(\boldsymbol{y} \mid X, Y) + \log \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}_{q(\boldsymbol{\xi}|\boldsymbol{\eta}^*)}[p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{\xi}, \boldsymbol{\theta}_k)]) \tag{72}$$

are valid Monte Carlo estimators for the predictive density, and the same holds for the discriminative classifier. In practice, we typically use the former (in which we directly average the post-softmax samples) due to the relative implementation simplicity, although the latter is necessary for some forms of out of distribution detection. Note that in the latter estimator,

$$\log \frac{1}{K}\sum_{k} \boldsymbol{x}_k = \mathrm{LSE}_k(\log \boldsymbol{x}_k) - \log K \tag{73}$$

for generic $\boldsymbol{x}_k$ and $\log K$ vanishes in the softmax and my therefore be ignored, and where the use of log-sum-exp improves numerical stability.

C.7. Out of Distribution Detection

A desirable feature of robust deep learning models is the ability to distinguish between in-distribution and out of distribution (OOD) data. We use several metrics for OOD detection with VBLL models. For the discriminative VBLL, we follow previous work (Liu et al., 2022) and use the maximum softmax probability (Hendrycks and Gimpel, 2016) for an OOD measure. This is computed by sampling from the distribution over logits and passing these samples through the softmax, where they are averaged.

For the generative classification model, we can use the feature density

$$p(\boldsymbol{x} \mid X, Y) \approx \sum_{\boldsymbol{y}} \mathbb{E}_{\boldsymbol{\theta}, \boldsymbol{\xi}}[p(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{\xi}, \boldsymbol{\theta})] \tag{74}$$

as an OOD measure. In the above, the expectation are with respect to the variational posteriors; for the MAP estimation case, this corresponds to direct evaluation.

In practice, we found post-training noise covariance calibration improved OOD detection performance for the G-VBLL model. More precisely, we aim to replace a shared diagonal Σ across all classes with a $\Sigma[\boldsymbol{y}]$ for each class. Our intuition is that while the Σ that is used in training is *prescriptive*—in the sense that it provides a model within which learning occurs—the estimated per-class $\Sigma[y]$ are *descriptive* of the accuracy of modelling during training. Indeed, the training objective for the G-VBLL model is label (marginal) predictive likelihood, and so the training signal to model class feature densities highly accurately is weak.

Our calibration procedure is as follows. First, we assume a (MAP) point estimate for feature means $\bar{\boldsymbol{\phi}}[\boldsymbol{y}]$. For sufficiently large datasets $S[\boldsymbol{y}]$ rapidly concentrates, so the impact of this assumption is relatively minor. For each class, we then compute the MAP noise covaraince $\Sigma[\boldsymbol{y}]$ under the inverse-Wishart prior. Concretely, the mean under Gaussian prior $\mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0)$ and known noise covariance Σ is

$$\boldsymbol{\mu}[\boldsymbol{y}] = (\Sigma_0^{-1}[\boldsymbol{y}] + T_y \Sigma^{-1})(\Sigma^{-1} \sum \boldsymbol{\phi}_t + \Sigma_0^{-1}[\boldsymbol{y}] \boldsymbol{\mu}_0[\boldsymbol{y}]) \tag{75}$$

$$= (\frac{1}{T_y} \Sigma_0^{-1}[\boldsymbol{y}] \Sigma + I)(\frac{1}{T_y} \sum \boldsymbol{\phi}_t + \Sigma \Sigma_0^{-1}[\boldsymbol{y}] \boldsymbol{\mu}_0[\boldsymbol{y}]) \tag{76}$$

where recall T_y is the number of class occurances for class $\boldsymbol{y}$ and where the sum is over all inputs in class $\boldsymbol{y}$. For sufficiently large T and zero mean prior, this mean is approximately equal to the empirical average $\frac{1}{T} \sum \boldsymbol{x}_t$. Thus, taking $\hat{\boldsymbol{\mu}} = \frac{1}{T} \sum \boldsymbol{x}_t$, the noise covariance can be estimated as

$$\hat{\Sigma}[\boldsymbol{y}] = \frac{1}{T_y + \nu + N + 1} (M + \sum (\boldsymbol{\phi}_t - \hat{\boldsymbol{\mu}}[\boldsymbol{y}])(\boldsymbol{\phi}_t - \hat{\boldsymbol{\mu}}[\boldsymbol{y}])^\top) \tag{77}$$

which corresponds to the MAP posterior with a known mean, and where the sum is again over all inputs in class $\boldsymbol{y}$.

We note that while our strategy of sequentially estimating two MAP estimates is relatively unsophisticated, it is straightforward and yields good results, and is consistent for large datasets (under straightforward distributional assumptions). In the above, N corresponds to the dimension of the covariance matrix (as in (65)) and ν and M corresponds to the prior degrees of freedom and scale matrix, respectively. We found that this MAP covariances estimation outperformed the max likelihood covariance estimation as performed in

(*Liuet al.*, 2022). Moreover, we note that both the empirical mean of the features for each class and the covariance can be recursively estimated in one pass over the data, and so the complexity of this step is $\mathcal{O}(T)$. Inspired by (Ren et al., 2019, 2021), we subtract the log density under the feature prior as a normalization strategy, which also slightly improves performance.

While this post-training last layer posterior improves OOD performance, it is substantially over-concentrated for label prediction, yielding to dramatically over-confident predictions. It is an open question how to best estimate the last layer posterior to achieve both effective and calibrated label and OOD prediction.

Appendix D. Parameterization and Complexity

In this section, we discuss how to parameterize each of the terms appearing in each type of VBLL. In each model, we use a "mixed" parameterization—in contrast to the standard parameterization or natural parameterization. More precisely, we will parameterize the inverse noise covariance Σ^{-1} and the covariance of the variational posterior S via Cholesky factorizations, and directly parameterize means $\bar{W}, \boldsymbol{\mu}$. In our (limited) comparisons of the performance of different parameterizations, we found that our mixed parameterization performed equivalently (if slightly better) to the standard parameterization, and both performed better than natural parameterization. Interestingly, this stands in contrast to standard practice in variational Gaussian process learning (Hensman et al., 2013), in which authors frequently aim to derive natural gradient optimization algorithms.

We will show that for each VBLL model, under a set of reasonable assumptions, complexity is at worst quadratic in the last layer width and linear in the output dimension. These complexity results enable use of VBLL models on problems with high input dimensionality and high output dimensionality. Moreover, our mini-batch gradient estimation training objective results in (standard) linear complexity of gradient estimation in batch size, enabling training on much larger datasets than is possible with standard marginal likelihood objectives.

D.1. Regression

Our analysis will focus on the multivariate case, for which the univariate outputs are a special case. We directly parameterize the mean $\bar{W} \in \mathbb{R}^{N_y \times N_\phi}$. The covariances are parameterized via Cholesky decomposition to guarantee positive semi-definiteness; in particular we parameterize $\Sigma^{-1} = LL^\top$ and $S = PP^\top$. Each of these Cholesky matrices must be lower triangular with positive diagonals, and so they are parameterized in automatic differentiation packages as

$$L = \texttt{tril}(L_d) + \texttt{diag}(\exp(\boldsymbol{m})) \tag{78}$$

for matrix L_d and vector $\boldsymbol{m}$, and where `tril` denotes the triangular-lower function.

Given these parameterization, we show the complexity of each operation required for training is at most quadratic in N_ϕ. The training objective has two terms: the log Gaussian density and the trace term. For the log density, we have

$$\boldsymbol{e}_t^\top \Sigma^{-1} \boldsymbol{e}_t = \boldsymbol{e}_t^\top L L^\top \boldsymbol{e}_t \tag{79}$$

for $\boldsymbol{e}_t = \boldsymbol{y} - \bar{W}\boldsymbol{\phi}_t$. The term $L^\top \boldsymbol{e}_t$ can be computed in $\mathcal{O}(N_y^2)$ time. The second term is $\boldsymbol{\phi}_t^\top S \boldsymbol{\phi}_t \mathrm{tr}(\Sigma^{-1})$, for which $\boldsymbol{\phi}_t^\top S \boldsymbol{\phi}_t$ can be computed in $\mathcal{O}(N_\phi^2)$ time, and the trace term

$$\mathrm{tr}(\Sigma^{-1}) = \mathrm{tr}(LL^\top) = \|L\|_F^2 \tag{80}$$

which can be computed in $\mathcal{O}(N_y^2)$ time via squaring and summing the elements of L.

The remaining terms are the KL penalty on the variational posterior, and the inverse-Wishart prior on the noise covariance. Fixing a prior $\mathcal{MN}(\bar{W}_0, I, S_0)$, the KL penalty for the multivariate regression case is (ignoring constants)

$$KL(q(\boldsymbol{\xi} \mid \boldsymbol{\eta})||p(\boldsymbol{\xi})) = \frac{1}{2}(\text{tr}((\bar{W} - \bar{W}_0)^\top (\bar{W} - \bar{W}_0)S_0^{-1}) + N_y \text{tr}(S_0^{-1}S) + N_y \log \frac{\det S_0}{\det S}) \quad (81)$$

We will fix an isotropic prior, $S_0 = sI$ for $s > 0$. Thus, the first term is

$$\text{tr}((\bar{W} - \bar{W}_0)^\top (\bar{W} - \bar{W}_0)S_0^{-1}) = \frac{1}{s}\|\bar{W} - \bar{W}_0\|_F^2 \quad (82)$$

yielding complexity $\mathcal{O}(N_y N_\phi)$, and the second term is

$$N_y \text{tr}(S_0^{-1}S) = \frac{N_y}{s}\text{tr}(S) \quad (83)$$

where the trace can again be computed as the squared Frobenius norm of the Cholesky factor of P, for complexity $\mathcal{O}(N_\phi^2)$. The last term is

$$N_y \log \frac{\det S_0}{\det S} = N_y N_\phi \log s - N_y \text{logdet} S \quad (84)$$

where $\text{logdet}S = 2\text{logdet}(P)$ which is equal to the sum of the log diagonal elements, which can be computed in $\mathcal{O}(N_\phi)$.

Finally, we have the inverse-Wishart noise covariance prior, which has terms $\text{logdet}\Sigma^{-1}$ and $\text{tr}(M\Sigma^{-1})$ for scale matrix M. The log determinant term may be computed as previously, with complexity $\mathcal{O}(N_y)$. Choosing scale matrix $M = mI$, we have $\text{tr}(M\Sigma^{-1}) = m\text{tr}(\Sigma^{-1})$ which again is $\mathcal{O}(N_y^2)$. Summing all of this up, we have the total complexity of VBLL computations as $\mathcal{O}(N_y^2 + N_\phi^2)$, which is equivalent to the complexity of standard matrix multiplication; thus, there is effectively zero added computational expense from the VBLL model compared to a standard network. The reader may easily verify that complexity of prediction is no greater than the training complexity in the regression model.

D.2. Classification

The complexity for the discriminative classification model follows from the regression model. We use the same parameterization, although we turn to a diagonal noise covariance Σ. The computation of the KL penalty is identical to the regression case. The only difference is that $\boldsymbol{\phi}_t^\top S[\boldsymbol{y}]\boldsymbol{\phi}_t$ must be computed for all classes $\boldsymbol{y}$, yielding complexity $\mathcal{O}(N_\phi^2 N_y)$. This term dominates the complexity of this model; however, further factorization of the covariance is straightforward and can reduce the practical complexity. To predict in these models, sampling realizations of the last layer must be done to sample logits. This sampling is straightforward to do using the Cholesky factorization of the covariance, and has quadratic complexity.

For the generative classification model, we are limited by the $\Sigma + S[\boldsymbol{y}]$ term in the log-sum-exp. As far as we are aware, there is no (practical) way to compute this term with quadratic complexity, or otherwise inexpensively compute this log density. Thus, in this paper we restrict Σ and S to diagonal matrices, which results in linear complexity in N_ϕ for all operations in loss computation. Thus, under this approximate posterior, the complexity of the full training loss computation is $\mathcal{O}(N_\phi N_y)$, which is equivalent to standard neural network models. This covariance structure is relatively restrictive, and improvements may results from sparse covariance structures.

Appendix E. Related Work and Discussion

Bayesian methods capable of flexible nonlinear learning have been a topic of active study for the last several decades. Historically, early interest in Bayesian neural networks (MacKay, 1992; Neal, 1995) diminished as Gaussian processes rose to prominence (Rasmussen, 2004). In recent years, however, there has been growing interest in methods capable of learning expressive features, effectively quantifying uncertainty, and training efficiently on large datasets. Variational methods have seen particular attention in both neural networks (Blundell et al., 2015; Ovadia et al., 2019) and GPs (Hensman et al., 2013; Titsias, 2009; Liu et al., 2020) due to their flexibility and their ability to produce mini-batch gradient estimation training schemes.

While a wide range of work has aimed to produce more performant approximate Bayesian methods (including more expressive prior and posterior representations (Fortuin et al., 2021; Izmailov et al., 2021; Sun et al., 2019; Wilson and Izmailov, 2020)), they have still seen limited application, often due to the increased computational expense of these methods (Lakshminarayanan et al., 2017; Dusenberry et al., 2020). While some approaches to Bayesian neural networks have focused on improving the quality of the posterior uncertainty through e.g. better priors (Farquhar et al., 2020; Fortuin, 2022) or inference methods (Izmailov et al., 2021), other lines of work have focused on designing comparatively inexpensive approximate Bayesian methods. Indeed, simple strategies such as Bayesian dropout (Gal and Ghahramani, 2016) and stochastic weight averaging (Maddox et al., 2019) have seen much wider use than more expressive methods due to their simplicity.

One of the simplest Bayesian models is the BLL model that is the focus of this paper, which enables single-pass, often deterministic uncertainty prediction. This model has gained prominence through the lens of deep kernel learning (Wilson et al., 2016b,a; Watson et al., 2020; Liu et al., 2022) and within few-shot learning (Harrison et al., 2018, 2020; Harrison, 2021; Watson et al., 2021; Zhang et al., 2021). Deep kernel learning aims to augment standard neural network kernels with neural network inputs. This approach allows control of the behavior of uncertainty, particularly as a function of Euclidean distance (Liu et al., 2022). While stochastic variational inference has been applied to these models (Wilson et al., 2016a), efficient and deterministic mini-batch methods have not been a major focus. Moreover, classification in these models typically relies on sampling logits applying softmax functions, which increases variance (Ovadia et al., 2019; Kristiadi et al., 2020, 2021), or on Laplace approximation (Liu et al., 2022).

Within few-shot learning, exact conjugacy of the Bayesian linear regression model (Harrison et al., 2018) and Bayesian GDA (Harrison et al., 2020; Zhang et al., 2021; Snell et al., 2017) has been exploited for efficient few-shot adaptation. These models have (in addition to (Van Amersfoort et al., 2020) among others) shown the strong performance of GDA-based/radial basis function networks, especially on problems such as out of distribution detection, which we further highlight in this work. However, training these models (as well as the DKL methods discussed previously) relies on direct computation of the marginal likelihood. In contrast to prior work on DKL and few-shot learning, our approach achieves efficient and deterministic training and prediction through our variational objectives and through similarly exploiting conjugacy, and thus the added complexity compared to standard neural network models is minimal.

	Boston		Concrete		Energy	
	NLL (↓)	RMSE (↓)	NLL (↓)	RMSE (↓)	NLL (↓)	RMSE (↓)
VBLL	$\mathbf{1.80 \pm 0.04}$	$\mathbf{1.42 \pm 0.05}$	$\mathbf{2.49 \pm 0.07}$	$\mathbf{2.76 \pm 0.12}$	$\mathbf{0.69 \pm 0.07}$	$\mathbf{0.45 \pm 0.03}$
GBLL	2.90 ± 0.05	4.19 ± 0.17	3.09 ± 0.03	5.01 ± 0.18	$\mathbf{0.69 \pm 0.03}$	$\mathbf{0.46 \pm 0.02}$
LDGBLL	2.60 ± 0.04	3.38 ± 0.18	2.97 ± 0.03	4.80 ± 0.18	4.80 ± 0.18	0.50 ± 0.02
MAP	2.60 ± 0.07	3.02 ± 0.17	3.04 ± 0.04	4.75 ± 0.12	1.44 ± 0.09	0.53 ± 0.01
RBF GP	2.41 ± 0.06	2.83 ± 0.16	3.08 ± 0.02	5.62 ± 0.13	$\mathbf{0.66 \pm 0.04}$	$\mathbf{0.47 \pm 0.01}$
VBLL BNN	$\mathbf{1.78 \pm 0.09}$	$\mathbf{1.37 \pm 0.11}$	2.61 ± 0.12	2.92 ± 0.25	$\mathbf{0.57 \pm 0.09}$	$\mathbf{0.39 \pm 0.03}$
BNN	2.39 ± 0.04	2.74 ± 0.16	2.97 ± 0.03	4.80 ± 0.13	0.63 ± 0.05	0.43 ± 0.01
Ensemble	2.48 ± 0.09	2.79 ± 0.17	3.04 ± 0.08	4.55 ± 0.12	$\mathbf{0.58 \pm 0.07}$	$\mathbf{0.41 \pm 0.02}$
Dropout	2.36 ± 0.04	2.78 ± 0.16	2.90 ± 0.02	4.45 ± 0.11	1.33 ± 0.00	0.53 ± 0.01
SWAG	2.64 ± 0.16	3.08 ± 0.35	3.19 ± 0.05	5.50 ± 0.16	1.23 ± 0.08	0.93 ± 0.09

Table 2: Negative log likelihood (NLL) and root mean squared error (RMSE) for UCI regression tasks. Baseline models are reproduced from (Watson et al., 2021), and our experimental procedure carefully matched theirs. The upper set of methods (VBLL through RBF GP) are single-pass methods, whereas the lower group are multiple-pass.

	Power		Wine		Yacht	
	NLL (↓)	RMSE (↓)	NLL (↓)	RMSE (↓)	NLL (↓)	RMSE (↓)
VBLL	$\mathbf{2.67 \pm 0.01}$	$\mathbf{3.48 \pm 0.01}$	$\mathbf{0.45 \pm 0.02}$	$\mathbf{0.37 \pm 0.01}$	1.71 ± 0.29	0.83 ± 0.07
GBLL	2.77 ± 0.01	3.85 ± 0.03	1.02 ± 0.01	0.64 ± 0.01	1.67 ± 0.11	1.09 ± 0.09
LDGBLL	2.77 ± 0.01	3.85 ± 0.04	1.02 ± 0.01	0.64 ± 0.01	1.13 ± 0.06	0.75 ± 0.10
MAP	2.77 ± 0.01	3.81 ± 0.04	0.96 ± 0.01	0.63 ± 0.01	5.14 ± 1.62	0.94 ± 0.09
RBF GP	2.76 ± 0.01	3.72 ± 0.04	$\mathbf{0.45 \pm 0.01}$	0.56 ± 0.05	$\mathbf{0.17 \pm 0.03}$	$\mathbf{0.40 \pm 0.03}$
VBLL BNN	2.73 ± 0.02	3.70 ± 0.07	0.67 ± 0.02	0.46 ± 0.01	1.96 ± 0.22	1.03 ± 0.13
BNN	2.77 ± 0.01	3.86 ± 0.04	0.95 ± 0.01	0.63 ± 0.01	1.43 ± 0.17	1.10 ± 0.11
Ensemble	2.70 ± 0.01	3.59 ± 0.04	0.95 ± 0.01	0.63 ± 0.01	0.35 ± 0.07	0.83 ± 0.08
Dropout	2.80 ± 0.01	3.90 ± 0.04	0.93 ± 0.01	0.61 ± 0.01	1.82 ± 0.01	1.21 ± 0.13
SWAG	2.77 ± 0.02	3.85 ± 0.05	0.96 ± 0.03	0.63 ± 0.01	1.11 ± 0.05	1.13 ± 0.20

Table 3: Further results for UCI regression tasks.

Appendix F. Experiments

We investigate the three VBLL models, with both MAP and variational feature learning, in regression and classification tasks. To illustrate VBLL models, we show predictions on simple datasets in Figure 1. The left figure shows a regression VBLL model with variational features trained on the function $f(x) = cx^3$, with training data shown in red. This figure shows the behavior on so-called *gap* datasets—so named because of the interval between subsets of the data. The VBLL model shows desirable increasing uncertainty between the intervals (Foong et al., 2019). The right figure shows the generative classification model (G-VBLL) on the half-moon dataset. In particular, we visualize the feature density for each class. Importantly, the density has high Euclidean distance sensitivity, which has been advocated by (Liu et al., 2022) as a desirable feature for robustness and out of distribution detection.

Metrics. For regression experiments, we report the predictive negative log likelihood (NLL) of test data, which can be computed in closed form for point feature estimates. We also report the root mean squared error (RMSE), a standard metric for regression. For classification, in addition to the negative log likelihood, we also report predictive accuracy (based on standard argmax of the predictive distribution), and expected calibration error (ECE), which measures

how the model's subjective predictive uncertainty agrees with predictive error. Finally, we also investigate out of distribution detection performance, a standard evaluation scheme for robust and probabilistic machine learning (Liu et al., 2022). We compute the area under the ROC curve (AUC) for near-OOD and far-OOD datasets, which is discussed in more detail later in this section.

Baselines. We distinguish baselines between single-pass and multi-pass models, which we show in upper and lower segments of each table, respectively. Single-pass methods require only a single network evaluation, and we compare VBLLs with MAP feature estimation to these models. Within regression, we compare to models which exploit exact conjugacy, including Bayesian last layer models (GBLL and LDGBLL (Watson et al., 2021)) and RBF kernel Gaussian processes. We note that these methods require computing full marginal likelihood and are thus difficult to scale to large training sets. We also compare to MAP learning, in which a full network is trained via MAP estimation, and a Bayesian last layer is fit to these fixed features (Snoek et al., 2015). Within classification, we primarily compare to standard networks (DNN), as these output a distribution over labels and thus can be directly compared to our model. We also compare to SNGP (Liu et al., 2022), which is a similar last layer model wihch aims to approximate deep kernel GPs (Wilson et al., 2016b). We note that in contrast to SNGP (Liu et al., 2022), we do not modify a standard neural network backbone, such as including aggressive weight decay, adding residual connections, or using sinusoidal nonlinearities. Multi-pass methods require several network evaluations, and includes variational methods like Bayes-by-backprop (which we refer to as BNN) (Blundell et al., 2015), ensembles (Lakshminarayanan et al., 2017), Bayesian dropout (Gal and Ghahramani, 2016) and stochastic weight averaging-Gaussian (SWAG) (Maddox et al., 2019).

F.1. Regression

We investigate the performance of the regression VBLL models on UCI regression datasets (Dua and Graff, 2017), which are standard benchmarks for Bayesian neural network regression (Moberg et al., 2019; Ober and Rasmussen, 2019; Daxberger et al., 2021b; Watson et al., 2021; Kristiadi et al., 2021). Results are shown in Tables 2, 3. We include baseline models run in (Watson et al., 2021), and we replicate their experimental procedure as closely as possible (details in the appendix). We use a fixed inverse-Wishart prior with $\nu = 1$ and identity scale matrix for all datasets.

Our experiments show strong results for VBLL models across datasets. Of particular interest is the performance relative to the GBLL model, which is trained directly on the exact marginal likelihood within the Bayesian last layer model. There are several contributing factors: the prior parameters were jointly optimized with the feature weights in the GBLL model, whereas prior terms were fixed in our VBLL model, resulting in a stronger regularization effect. Moreover, exact Bayesian inference can perform poorly under model misspecification (Grünwald and Van Ommen, 2017), whereas variational Bayes has comparatively favorable robustness properties and asymptotics (Giordano et al., 2018; Wang and Blei, 2019), although the Gaussian process (GP) model generally also has strong performance across datasets. Finally, directly targeting the marginal likelihood (computed exactly within conjugate models such as BLL models) has been shown to induce substantial overfitting (Ober et al., 2021; Thakur et al., 2020; Harrison, 2021), which the variational approach may avoid due to worse inferential efficiency.

	Accuracy (↑)	ECE (↓)	NLL (↓)	SVHN AUC (↑)	CIFAR-10 AUC (↑)
DNN	80.4 ± 0.29	0.107 ± 0.004	0.941 ± 0.016	0.799 ± 0.020	0.795 ± 0.001
SNGP	80.3 ± 0.23	$\mathbf{0.030 \pm 0.004}$	$\mathbf{0.761 \pm 0.007}$	$\mathbf{0.846 \pm 0.019}$	$\mathbf{0.798 \pm 0.001}$
D-VBLL	$\mathbf{80.7 \pm 0.03}$	0.040 ± 0.002	0.913 ± 0.011	$\mathbf{0.849 \pm 0.006}$	0.791 ± 0.003
G-VBLL	80.4 ± 0.10	0.051 ± 0.003	0.945 ± 0.009	0.767 ± 0.055	0.752 ± 0.015
G-VBLL-MAP	–	–	–	0.793 ± 0.032	0.765 ± 0.008
BNN	79.6 ± 0.04	0.127 ± 0.002	1.611 ± 0.006	0.809 ± 0.060	0.777 ± 0.008
D-VBLL BNN	77.6 ± 0.17	0.041 ± 0.003	1.169 ± 0.018	0.785 ± 0.022	0.756 ± 0.002
G-VBLL BNN	78.1 ± 0.18	0.046 ± 0.002	1.156 ± 0.008	0.832 ± 0.023	0.742 ± 0.004

Table 4: Results for Wide ResNet-28-10 on CIFAR-100.

F.2. Classification

To quantitatively evaluate performance of VBLL models in classification, we train the discriminative (D-VBLL) and generative (G-VBLL) classification models on the CIFAR-10 and CIFAR-100 image classification task. Following (Liu et al., 2022), all experiments utilize a Wide ResNet-28-10 backbone architecture. We evaluate out of distribution (OOD) detection performance using the Street View House Numbers (SVHN) (Netzer et al., 2011) as a far-OOD dataset for both datasets, and CIFAR-100 for CIFAR-10 (and vice-versa) as near-OOD datasets. In-distribution data normalization is used in both cases. The DNN, BNN, D-VBLL and D-VBLL BNN models use maximum softmax probability (Hendrycks and Gimpel, 2016) as an OOD measure. The G-VBLL and G-VBLL BNN models use a normalized feature density. Two methods for this exist: G-VBLL and G-VBLL BNN both use the learned variational posteriors to compute feature likelihoods. However, the performance of this is relatively weak, as there is no guarantee that learned feature likelihoods correspond effectively to true embedding densities. Thus, we also investigate an approach in which we estimate distributions for fixed features after training. This method estimates noise covariances for each class using the trained features, similar to the approach used in (Liu et al., 2022). We refer to this model is G-VBLL-MAP, as the approach corresponds to MAP noise covariance estimation. These estimated covariances often result in overly-confident predictions, and so we do not advocate for label prediction under these fit covariances, and do not include results for them. Appendix C.7 discusses OOD methods, and further experimental details are in Appendix G.

Tables 1, 4 summarize the CIFAR-10 and CIFAR-100 results. D-VBLL and G-VBLL report strong accuracy performance and competitive metrics for both ECE and NLL. D-VBLL in particular demonstrates extremely strong accuracy results, as well as competitive (with SNGP) NLL and OOD detection ability. Despite its comparative simplicity, it outperforms SNGP on accuracy and OOD on CIFAR-10 and accuracy on CIFAR-100. It matches SNGP on OOD for CIFAR-100, and is competitive (although slightly worse) on ECE and NLL. Overall, D-VBLL models stand out as exceptionally strong performance relative to their complexity.

While models with MAP feature estimation show very strong performance versus baseline models, the performance of variational feature learning models (BNN) is more mixed. In regression tasks, these models are competitive, while in classification the performance is worse than deterministic models. In both settings, we use default KL term weighting (one over the dataset size). This contrasts with the tempered/cold posterior effect (Kapoor et al., 2022; Wenzel et al., 2020; Izmailov et al., 2021; Aitchison, 2020), in which it has been observed that alternative weightings of the likelihood and the KL may outperform this one. This is attributable (in part) to two factors: data augmentation and stochastic regularization. In regression there is no data augmentation and the model is trained for substantially longer

than deterministic models; in classification we use standard augmentation and our training is more limited. Thus, it is possible that classification BNN models are over-regularized. We investigate this question in more detail in the appendix.

Appendix G. Experimental Details

This section contains details about the experiments in the body of the paper. We note that for highlighting in the tables in the paper body, if a single-pass method (in the upper half of each table) is the best performing in a metric, that result is highlighted. If the best performing is multi-pass, we highlight both the best multi-pass and single-pass method in the column. We believe that this is important, as many applications required single-pass methods and thus multi-pass results are irrelevant.

G.1. Toy Experiments

Figure 1 contains simple visualizations for the regression model and the generative VBLL model. In particular, the regression model shows predictions with variational feature learning (with KL weight of 1.0) on a cubic function with a gap in the data. This dataset consisted of 100 points sampled in $[-4, -2] \cup [2, 4]$, with a noise standard deviation of 0.1. The model was consisted of a two hidden-layer MLP of width 128, trained for 1000 epochs with a batch size of 32, with stochastic gradient descent with momentum, with a learning rate of $3 \cdot 10^{-4}$, zero weight decay, and momentum beta parameters of 0.9. These values were arbitrarily chosen, although the choice of SGDM versus Adam (Kingma and Ba, 2015) does make a difference on prediction far from the data. Gradient clipping with a maximum magnitude of 2.0 was used. The DOF and scale parameters were both set to 1.0

For the classification problem, we used the scikit-learn implementation of the half moon dataset, with 1000 data points and a noise standard deviation of 0.2. We trained a G-VBLL model with residual-structured MLP of width 128 (each hidden layer is added to the layer input). This model was trained with SGDM with learning rate $3 \cdot 10^{-2}$, momentum beta 0.9, and weight decay 10^{-4}, for 100 epochs and with a batch size of 32. The DOF parameter was 128, and the scale parameter was 1.0.

G.2. Regression

Our UCI experiments closely follow (Watson et al., 2021), and we compare directly to their baselines. For VBLLs, we used a $\mathcal{N}(0, I)$ last layer mean prior and a $\mathcal{W}^{-1}(1, 1)$ noise covariance prior. For all experiments, we use the same MLP used in (Watson et al., 2021) consisting of two layers of 50 hidden units each (not counting the last layer). For all datasets we matched (Watson et al., 2021) and used a batch size of 32, other than the Power dataset for which we used a batch size of 256 to accelerate training. For all datasets we normalize inputs (using the training set statistics) and subtract the training set means for the outputs. We did not re-scale the output magnitudes, to retain comparability of NLLs. We note that the extent to which outputs were normalized in (Watson et al., 2021) is unclear. However, they make the parameters of their prior learnable, which can have a similar effect to centering the outputs, and so we believe our output centering is reasonable. All results shown in the body of the paper are for leaky ReLU activations. For all experiments, a fixed learning rate of 0.001 was used with the AdamW optimizer (Loshchilov and Hutter, 2017). A default weight decay of 0.01 was used for all experiments. We clipped gradients with a max magnitude of 1.0.

Features	Boston	Concrete	Energy	Power	Wine	Yacht
MAP	3000	3000	2000	3000	1000	2000
Variational	10000	10000	10000	10000	10000	10000

Table 5: Maximum number of epochs for each set of features and each UCI dataset.

		Feature KL Weight		
LL KL Weight	MAP	50	5	0.5
1.0	0.160	0.266	0.281	0.282
0.1	0.162	0.266	0.286	0.272
0.01	0.168	0.268	0.268	0.280
0.001	0.160	0.267	0.272	0.276

Table 6: CIFAR-10 NLL for varying values of KL weights, for both the last layer and the feature weighting in variational feature learning.

For all deterministic feature experiments, we ran 20 seeds. For each seed, we split the data in to train/val/test sets (0.72/0.18/0.1 of the data respectively). We train on the train set and monitor performance on the validation set to choose a total number of epochs. In contrast to (Watson et al., 2021) who compute validation performance for every epoch, we compute validation performance (predictive NLL) every 10 epochs (note that the datasets are small and typically train for hundred of epochs). After choosing a number of epochs, we train on the combined training and validation set and evaluate performance on the test set. We use a max number of epochs shown in Table 5, which were large enough to not be reached but often lower than those used in (Watson et al., 2021).

For our BNN feature models, we ran 10 seeds with a similar procedure to the above. We follow (Watson et al., 2021) and use a $\mathcal{N}(0, 4/\sqrt{n_{\text{in}}})$ for each weight (where n_{in} denotes the layer input width). Validation performance was monitored every 100 epochs, and 10 weight samples were used to compute the validation predictive likelihood and choose a full training number of epochs. For test set evaluation, 100 weight samples were used.

G.3. Classification

All classification experiments utilize the Wide ResNet-28-10 (WRN-28-10) backbone network architecture. Hyperparameters are similar to those proposed by (Zagoruyko and Komodakis, 2016). Unlike the original implementation of WRN, we do not employ Nesterov momentum and we fully decay an initial learning rate of 0.1 according to a Cosine Annealing schedule instead of a stepped decay schedule. Gradients are clipped with a maximum magnitude of 2.0 and we impose a last layer KL weight of 1.0. We All classification results are reported across 3 seeds and use the standard WRN data-augmentations proposed by (Zagoruyko and Komodakis, 2016).. For the deterministic feature experiments, we train each model for 300 epochs.

The BNN backbone-based models utilize the same WRN architecture and are primarily deterministic. The BNN models implement a single final Bayesian linear layer with a prior distribution of $\mathcal{N}(0, 0.01)$. Each BNN-based model used 10 weight samples for test set evaluation. This operation is relatively cheap when compared to a fully stochastic network because the intermediate features are cached prior to the final Bayesian linear layer weight sampling and computation. All BNN are trained for 400 epochs and we impose a last layer KL weight of 1.0 and a feature KL weight of 0.5 the VBLL-BNN and DBLL-BNN models. The BNN baseline model utilized a feature KL weight of 50. We additionally explore the

NLL sensitivity of the DBLL and DBLL-BNN models to various KL weighting configurations. In Table 6, we sweep across orders of magnitude for both the last layer and feature KL weighting parameters. The exploration results suggest that higher feature KL weighting improves the BNN-based models.