Workshop at the 7th Symposium on Advances in Approximate Bayesian Inference (non-archival), 2025 1–22

Heteroscedastic Variational Last Layers

James Harrison JAMESHARRISON@GOOGLE.COM
Google DeepMind

John Willes JOHN.WILLES@VECTORINSTITUTE.AI
Vector Institute

Paul Brunzema PAUL.BRUNZEMA@DSME.RWTH-AACHEN.DE
RWTH Aachen

Jasper Snoek JSNOEK@GOOGLE.COM
Google DeepMind

Abstract

We improve the performance of Variational Bayesian Last Layer (VBLL) networks by better modeling aleatoric noise. In particular, we (1) Introduce t-VBLL layers, which perform variational inference for the noise covariance, and (2) Introduce Het-VBLL, a Bayesian last layer scheme to model heteroscedastic noise. These methods are based on novel, analytically tractable evidence lower bounds. We show that these novel design elements extend the capabilities of VBLLs at minimal additional cost, and substantially improve performance.

1. Introduction

Variational Bayesian Last Layers (VBLLs) are a lightweight and effective method for uncertainty quantification within neural networks (Harrison et al., 2024; Brunzema et al., 2025; Watson et al., 2021). VBLLs replace point estimation of neural network last layers with a variational inference, and exploit analytically tractable evidence lower bounds (ELBOs) to yield an easy-to-train, inexpensive approach to Bayesian deep learning.

The uncertainty quantification in these models relies on the likelihood, which assumes additive Gaussian noise. For standard VBLLs, a point estimate of the noise variance is computed via MAP estimation, due to the simplicity of this procedure in combination with neural network training. Although the noise term may appear unimportant, it is essential for weighing the predictive accuracy and uncertainty terms, thus playing a vital role in accurate uncertainty quantification. Moreover, the standard VBLL model assumes homoscedastic noise (Hayashi, 2011), i.e., the noise in the likelihood has the same magnitude for all inputs.

In this work, we investigate this noise term in VBLLs. Specifically, we present two methods for more expressive noise modeling. First, we develop t-VBLLs—a variational approach to noise covariance inference that maintains a full variational posterior over the noise covariance. In the regression case, this leads to a familiar Student t-distributed predictive distribution. For classification, we introduce a novel sampling-free lower bound on the standard ELBO, enabling effective training with low variance. Second, we introduce Het-VBLLs—a method for variational inference of heteroscedastic noise (Le et al., 2005). This

© J. Harrison, J. Willes, P. Brunzema & J. Snoek.

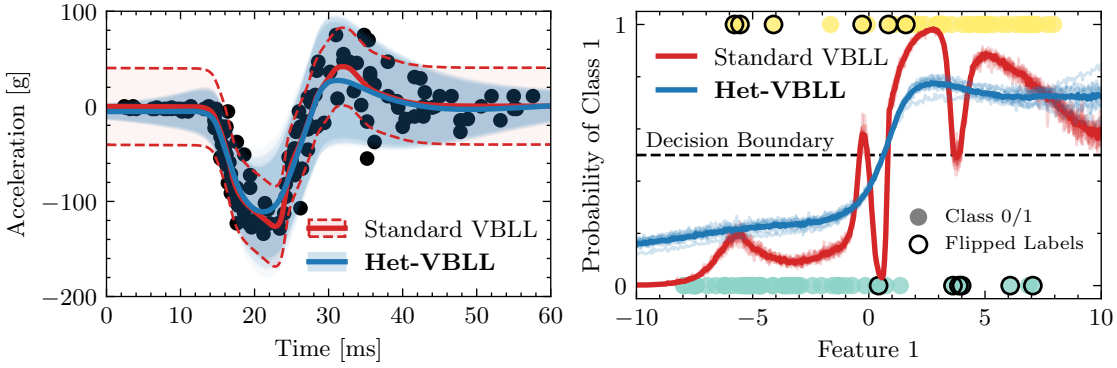

Figure 1: *Left:* Het-VBLLs vs. VBLLs on motorcycle-impact data from Silverman (1985). Het-VBLLs can capture the heteroscedasticity for expressive uncertainty estimates. *Right:* Classification on a toy sigmoid dataset with 100 points and 10% flipped labels. The Het-VBLL models demonstrate greater robustness to outliers.

builds upon a second VBLL model to characterize noise and leverages the novel training ELBOs developed in the first part of the paper.

These approaches introduces only a handful of new parameters, making them highly scalable while substantially improving the calibration and uncertainty quantification of neural networks. We demonstrate the effectiveness of our models across various settings, including supervised regression, classification, and Bayesian optimization. In the following, we explain the key ideas behind the methods in the body of the paper but refer to the Appendix for the technical details.

2. General Setup and Preliminaries

We consider models of the form $\boldsymbol{z} = W\boldsymbol{\phi}(\boldsymbol{x}) + \boldsymbol{\varepsilon}$ where in the regression case $\boldsymbol{y} = \boldsymbol{z}$, and in the classification case $p(\boldsymbol{y} \mid \boldsymbol{x}) = \text{softmax}(\boldsymbol{z})$. VBLLs (Harrison et al., 2024) fix a prior and variational posterior over W, which we denote as $p(W)$ and $q(W)$, respectively. We refer to the dimensionality of inputs, outputs, and features as N_x, N_y, and N_ϕ respectively. Harrison et al. (2024) develop a lower bound on the standard variational lower bound (of the log marginal likelihood),

$$\log p(Y \mid X, \Sigma) \geq \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \mathbb{E}_{q(W)}[\log p(\boldsymbol{y} \mid \boldsymbol{x}, W, \Sigma)] - \text{KL}(q(W)\|p(W)) \tag{1}$$

that exploits exact computation of the expected log likelihood (or a lower bound on this term, in the case of classification). This lower bound yields an inexpensive, sampling-free objective. To train these models, the weights of the neural network features $\boldsymbol{\phi}$ and the variational posterior $q(\cdot)$ are jointly trained. It is assumed throughout Harrison et al. (2024) that $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \Sigma)$, and Σ is either fixed a priori, or a point estimate is learned by either MAP estimation or maximum likelihood.

In this paper, we will consider variational inference of Σ, the noise covariance. Throughout, we will consider a factorized variational posterior over diagonal elements of the noise covariance, $q(\Sigma) = \prod_i^{N_y} q(\Sigma_i)$. We consider both homoscedastic variational posteriors of the form $q(\Sigma)$ and heteroscedastic variational posteriors $q(\Sigma \mid \boldsymbol{x})$.

3. Training Objectives for Homoscedastic and Heteroscedastic Modeling

In this section we introduce two models: t-VBLLs and Het-VBLLs (see toy experimental results in Fig. 1). We first discuss the noise parameterization that defines each model. We introduce a set of variational bounds used as training objectives, and these general results apply to both homoscedastic and heteroscedastic models. A full discussion of the technical details is provided in the Appendix.

3.1. Parameterizing the Noise Distribution

We consider independent priors (and variational posteriors) over rows of W and fix a diagonal Σ, and so it is sufficient to consider each row independently, of the form

$$\boldsymbol{z}_i = \boldsymbol{w}_i^\top \boldsymbol{\phi}(\boldsymbol{x}) + \boldsymbol{\varepsilon}_i \tag{2}$$

with $\boldsymbol{\varepsilon}_i \sim \mathcal{N}(0, \Sigma_i)$. We fix a prior over the last layer parameters for each row $\boldsymbol{w}_i$ and the covariance element Σ_i

$$p(W, \Sigma \mid \boldsymbol{x}) = \prod_{i=1}^{N_y} p(\boldsymbol{w}_i \mid \Sigma_i, \boldsymbol{x}) p(\Sigma_i \mid \boldsymbol{x}) \tag{3}$$

with $p(\boldsymbol{w}_i \mid \Sigma_i) = \mathcal{N}(\underline{\bar{\boldsymbol{w}}}_i, \Sigma_i(\boldsymbol{x})\underline{S}_i)$[1]. This corresponds to the same choice of prior as Harrison et al. (2024), although with the covariance parameterization being scaled by Σ_i. This scaling is standard in Bayesian linear regression to enable recursive updating of sufficient statistics of the posterior. Here, it yields easy-to-evaluate variational objectives.

$$q(\boldsymbol{w}_i \mid \Sigma_i) = \mathcal{N}(\boldsymbol{w}_i, \Sigma_i(\boldsymbol{x})S_i) \tag{4}$$

We similarly structure our variational posterior as

$$q(W, \Sigma \mid \boldsymbol{x}) = \prod_{i=1}^{N_y} q(\boldsymbol{w}_i \mid \Sigma_i, \boldsymbol{x}) q(\Sigma_i \mid \boldsymbol{x}) \tag{5}$$

following standard results in Bayesian regression. We will consider two variational families for the noise covariance. In the homoscedastic setting we will fix an inverse Gamma variational posterior. In this homoscedastic case, Σ_i and the variational posterior do not depend on $\boldsymbol{x}$. This is the canonical (conjugate) prior for the noise covariance in Bayesian linear regression. This choice of prior results in a Student t-distributed posterior predictive distribution (Box and Tiao, 2011), which we exploit in our prediction, giving rise to t-VBLLs.

In the heteroscedastic setting, we parameterize the noise covariance with a VBLL as

$$\log \Sigma_i = \boldsymbol{m}_i^\top \boldsymbol{\phi}(\boldsymbol{x}) \tag{6}$$

with $\boldsymbol{m}_i \sim q(\boldsymbol{m}_i) = \mathcal{N}(\bar{\boldsymbol{m}}_i, Z_i)$ (and similarly choose as prior $\boldsymbol{m}_i \sim \mathcal{N}(\underline{\bar{\boldsymbol{m}}}_i, \underline{Z}_i)$). This yields a log-Normal variational posterior for Σ. We refer to the approach of using a second VBLL for the noise covariance as Het-VBLL. While this does not result in a closed-form predictive distributions, it yields convenient evaluation of the training evaluation. More precisely, for both the inverse Gamma and log-Normal variational posteriors, $\mathbb{E}[\Sigma_i], \mathbb{E}[\Sigma_i^{-1}]$, and $\mathbb{E}[\log \Sigma_i]$ (where the expectation is taken with respect to the variational posterior) are analytically tractable, which we exploit in the development of our variational lower bounds.

1. Throughout, we use overbars to denote mean parameters and underbars to denote prior parameters

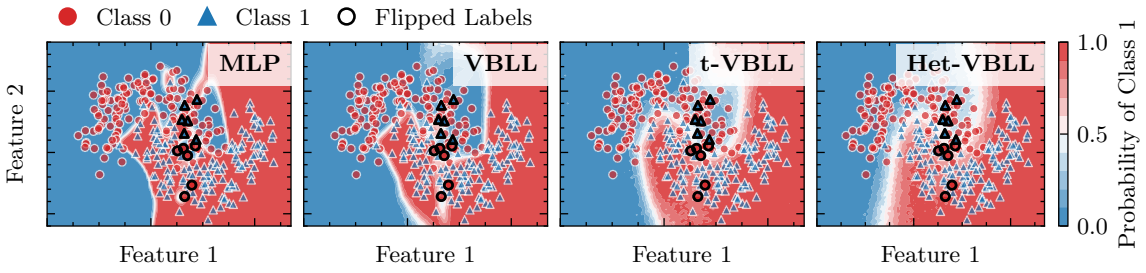

Figure 2: Classification on a noisy two-moon dataset under label noise. A standard MLP struggles on this complex data set. VBLLs show an improved decision boundary but still display slight overfitting. Our models perform best, with Het-VBLL providing the best qualitative results.

3.2. Variational Lower Bounds and Prediction

The variational lower bounds is then

$$\log p(Y \mid X) \geq \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \mathbb{E}_{q(W,\Sigma|\boldsymbol{x})}[\log p(\boldsymbol{y} \mid \boldsymbol{x}, W, \Sigma)] - \mathrm{KL}(q(W,\Sigma \mid \boldsymbol{x}) \| p(W,\Sigma \mid \boldsymbol{x})). \tag{7}$$

With the chosen variational posterior in (5), we now discuss the bounds that we use as training objectives in both the homoscedastic and heteroscedastic case. We will focus on the likelihood term in (7) and discuss both the regression case and the classification case. Full development of results is presented in Appendix *C* for homoscedastic and Appendix *D* for heteroscedastic. For the likelihood term in the variational lower bound we have

$$\mathbb{E}_{q(W,\Sigma|\boldsymbol{x})}[\log p(\boldsymbol{y} \mid \boldsymbol{x}, W, \Sigma)] = \sum_{i=1}^{N_y} \mathbb{E}_{q(\Sigma_i|\boldsymbol{x})}[\mathbb{E}_{q(\boldsymbol{w}_i|\Sigma_i,\boldsymbol{x})}[\log p(\boldsymbol{y}_i \mid \boldsymbol{x}, \boldsymbol{w}_i, \Sigma_i)]]. \tag{8}$$

Regression. In the regression case, the inner expectation over $\boldsymbol{w}_i$ evaluates to

$$\mathbb{E}_{q(W,\Sigma|\boldsymbol{x})}[\log p(\boldsymbol{y} \mid \boldsymbol{x}, W, \Sigma)] = -\frac{1}{2}\sum_{i=1}^{N_y} \mathbb{E}_{q(\Sigma|\boldsymbol{x})}[\Sigma_i^{-1}(\boldsymbol{y}_i - \bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi})^2 + \log \Sigma_i] + \boldsymbol{\phi}^\top S_i \boldsymbol{\phi} \tag{9}$$

following results from Harrison et al. (2024). Due to linearity with respect to Σ_i^{-1} and $\log \Sigma_i$, this expectation is analytically tractable for both the homoscedastic and the heteroscedastic case, yielding a sampling-free training objective. Prediction for regression in the homoscedastic case is analytically tractable via a Student t-distributed posterior predictive. For the heteroscedastic case, we instead turn to sampling Σ.

Classification. For classification, we develop the following bound on the likelihood term

$$\mathbb{E}_{q(W,\Sigma|\boldsymbol{x})}[\log p(\boldsymbol{y} \mid \boldsymbol{x}, W, \Sigma)] \geq \boldsymbol{y}^\top \bar{W}\boldsymbol{\phi} - \mathrm{LSE}_i(\bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi} + \alpha_i(\boldsymbol{\phi}^\top S_i \boldsymbol{\phi} + 1)) \tag{10}$$

$$- \frac{1}{2}\sum_i \mathbb{E}_{q(\Sigma|\boldsymbol{x})}[4\Sigma_i - \alpha_i + \alpha_i^2 \Sigma_i^{-1}](\boldsymbol{\phi}^\top S_i \boldsymbol{\phi} + 1) \tag{11}$$

where α_i are variational parameters. This result was derived to yield linearity in Σ_i, Σ_i^{-1} which in turn yields analytical tractability of this objective. This result builds upon the variational multivariate logistic regression ELBO developed by Knowles and Minka (2011). For the full derivation, see Appendix C. For both the homoscedastic and heteroscedastic case, sampling Σ is required for prediction.

Table 1: Results on CIFAR-10

Method	Accuracy (↑)	ECE (↓)	NLL (↓)	SVHN OOD (↑)	CIFAR-100 OOD (↑)
DNN	95.8 ± 0.19	0.028 ± 0.028	0.183 ± 0.007	0.946 ± 0.005	0.893 ± 0.001
VBLL	$\mathbf{96.4 \pm 0.01}$	0.024 ± 0.000	0.176 ± 0.000	0.943 ± 0.002	0.895 ± 0.000
t-VBLL	96.3 ± 0.03	$\mathbf{0.006 \pm 0.000}$	$\mathbf{0.133 \pm 0.001}$	$\mathbf{0.975 \pm 0.000}$	0.892 ± 0.001
Het-VBLL	96.2 ± 0.02	0.009 ± 0.000	0.135 ± 0.000	$\mathbf{0.975 \pm 0.001}$	0.894 ± 0.001
LLLA	96.3 ± 0.03	0.010 ± 0.001	$\mathbf{0.133 \pm 0.003}$	0.965 ± 0.010	$\mathbf{0.898} \pm 0.001$

Table 2: Results on CIFAR-100

Method	Accuracy (↑)	ECE (↓)	NLL (↓)	SVHN OOD (↑)	CIFAR-10 OOD (↑)
DNN	80.4 ± 0.29	0.107 ± 0.004	0.941 ± 0.016	0.799 ± 0.020	0.795 ± 0.001
VBLL	80.7 ± 0.02	0.063 ± 0.000	0.831 ± 0.005	0.843 ± 0.001	0.804 ± 0.001
t-VBLL	$\mathbf{81.3 \pm 0.10}$	$\mathbf{0.039 \pm 0.001}$	0.782 ± 0.004	0.681 ± 0.017	$\mathbf{0.811 \pm 0.003}$
Het-VBLL	$\mathbf{81.2 \pm 0.07}$	$\mathbf{0.039 \pm 0.001}$	$\mathbf{0.777 \pm 0.005}$	0.685 ± 0.001	$\mathbf{0.811 \pm 0.001}$
LLLA	80.4 ± 0.29	0.210 ± 0.018	1.048 ± 0.014	$\mathbf{0.834} \pm 0.014$	$\mathbf{0.811 \pm 0.002}$

3.3. Training

Both the regression and classification models are trained in the same way as standard neural networks: both the weights of the variational posteriors and the neural network features $\boldsymbol{\phi}$ are trained via minibatch gradient descent methods. These models introduce minimal additional complexity; t-VBLL adds N_y parameters and is thus essentially free, whereas Het-VBLL adds a number of parameters approximately equal to adding an additional last layer. For regression models, we use the data-reweighting approach of Seitzer et al. (2022) to improve mean estimation, although the impact is relatively minor. The models presented here can both be used in full model training, or can be used in a multi-step training procedure. For example, these models are effective as heads trained on frozen features (which we do for our classification experiments) and show large performance gains in this setting.

4. Experiments

Supervised Regression. To demonstrate the Het-VBLLs ability to capture heteroscedatic noise, we compare them against standard VBLLs in Figure 1 on motorcycle-impact data (Silverman, 1985; Kersting et al., 2007). Het-VBLLs are able to capture the heteroscedasticity in the data whereas standard VBLLs with an MAP estimate do not. We also benchmark the t-VBLLs and Het-VBLLs on the popular UCI data sets (Dua and Graff, 2017) in Appendix F.1.2. Both models exhibit strong performance, matching or surpassing standard VBLLs (and other baselines) across all tasks.

Supervised Classification. We test our models on a noisy version of the two-moon data set in Figure 2. We further introduce label noise by flipping 20% of the labels for points within a specific range of the first feature (circled). A standard MLP overfits significantly, and even standard VBLLs struggle with these outliers. In contrast, our proposed t-VBLL and Het-VBLL models demonstrate robustness, achieving a significantly improved decision boundary.

We also evaluate the proposed models on CIFAR-10 and CIFAR-100, comparing t-VBLL and Het-VBLL against standard DNNs, VBLLs (Harrison et al., 2024), and LLLA (Daxberger et al., 2021) baselines. On CIFAR-10 (Table 1), both t-VBLL and Het-VBLL achieved competitive accuracy while significantly reducing Expected Calibration Error and

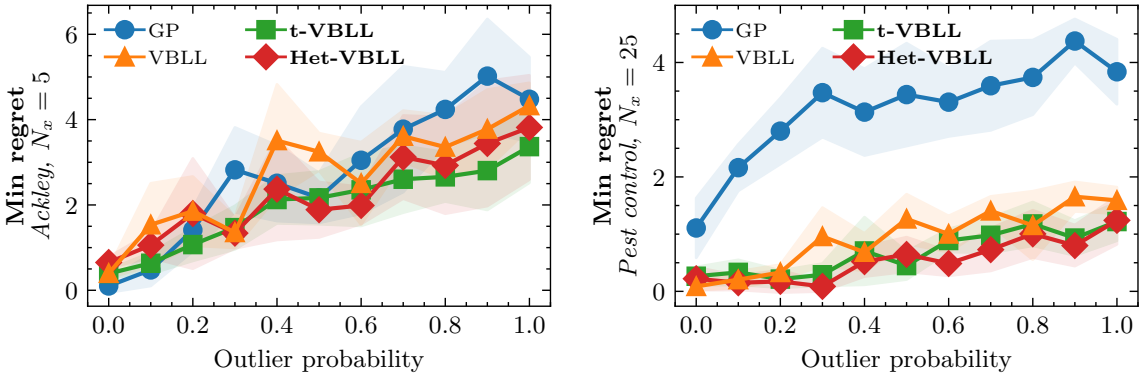

Figure 3: Minimum regret obtained under heavy-tail distributed outliers on Ackley (*left*) and Pest Control (*right*). Our models demonstrate more robustness to outliers.

Negative Log-Likelihood, with improved out-of-distribution detection performance. Similarly, on CIFAR-100 (Table 2), our models maintained accuracy with a substantial reduction in ECE and NLL. The results indicate that incorporating heavy-tailed and heteroscedastic modeling improves uncertainty quantification.

Bayesian Optimization. We evaluate t-VBLL and Het-VBLL networks as surrogates in Bayesian optimization. As baselines, we use standard VBLLs (Harrison et al., 2024; Brunzema et al., 2025) and Gaussian processes (GPs) with a Matérn kernel ($\nu = 2.5$). All experimental details and surrogate configurations are listed in Appendix F.3. We compare all models using an upper confidence bound (UCB) acquisition function as $\alpha_{UCB}(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \beta^{1/2}\sigma_{UCB}(\boldsymbol{x})$ (Srinivas et al., 2010). In our experiments, we use the same hyperparameter for UCB ($\beta = 2$) such that the difference in performance is only due to differences in the model.

As a benchmark experiments, we focus on the classic Ackley objective ($N_x = 5$) and on pest control ($N_x = 25$) (Oh et al., 2019), where automatic feature learning in Bayesian neural networks has been shown to outperform standard GPs (Li et al., 2024; Brunzema et al., 2025). To test our models, we add heavy-tailed noise from a zero-mean Laplace distribution to the outcome of an experiment, controlled by an outlier probability. In Fig. 3, we show the performance of all surrogates over this outlier probability for 10 random seeds each. VBLL-based models clearly outperform GPs. We can further see, that as the outlier probability increases also the minimum regret obtained by a surrogate increases. Our proposed models outperform the standard VBLLs as they are, by design, more robust to such outliers.

5. Discussion and Conclusion

In this paper we have introduced two novel methods for scalable Bayesian deep learning, each applicable to both regression and classification. Development of these methods relied on the design of novel training objectives and model architectures. These models show extremely strong performance relatively to both similar last layer methods and considerably more expensive methods.

References

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Christopher Bishop and Cazhaow Quazaz. Regression with input-dependent noise: A bayesian treatment. *Neural Information Processing Systems (NeurIPS)*, 9, 1996.

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.

Paul Brunzema, Mikkel Jordahn, John Willes, Sebastian Trimpe, Jasper Snoek, and James Harrison. Bayesian optimization via continual variational last layer training. *International Conference on Learning Representations (ICLR)*, 2025.

Mark Collier, Basil Mustafa, Efi Kokiopoulou, Rodolphe Jenatton, and Jesse Berent. A simple probabilistic method for deep classification under input-dependent label noise. *arXiv:2003.06778*, 2020.

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)*, 2021.

Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 2009.

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

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local Bayesian optimization. In *Neural Information Processing Systems (NeurIPS)*, 2019.

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

Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix Gaussian process inference with gpu acceleration. In *Neural Information Processing Systems (NeurIPS)*, 2018.

James Harrison, John Willes, and Jasper Snoek. Variational Bayesian last layers. *International Conference on Learning Representations (ICLR)*, 2024.

Fumio Hayashi. *Econometrics*. Princeton University Press, 2011.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Neural Information Processing Systems (NeurIPS)*, 30, 2017.

Kristian Kersting, Christian Plagemann, Patrick Pfaff, and Wolfram Burgard. Most likely heteroscedastic gaussian process regression. In *International Conference on Machine Learning (ICML)*, 2007.

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

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

Miguel Lazaro-Gredilla and Michalis K Titsias. Variational heteroscedastic gaussian process regression. In *International Conference on Machine Learning (ICML)*, 2011.

Quoc V Le, Alex J Smola, and Stephane Canu. Heteroscedastic gaussian process regression. In *International Conference on Machine Learning (ICML)*, pages 489–496, 2005.

Yucen Lily Li, Tim G. J. Rudner, and Andrew Gordon Wilson. A study of Bayesian neural network surrogates for Bayesian optimization. In *International Conference on Learning Representations (ICLR)*, 2024.

Jeremiah 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. *Journal of Machine Learning Research*, 2022.

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

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.

David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *International conference on neural networks (ICNN)*, 1994.

Changyong Oh, Jakub Tomczak, Efstratios Gavves, and Max Welling. Combinatorial Bayesian optimization using the graph cartesian product. *Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. *Neural Information Processing Systems (NeurIPS)*, 2018.

Ian Osband, Zheng Wen, Seyed Mohammad Asghari, Vikranth Dwaracherla, Morteza Ibrahimi, Xiuyuan Lu, and Benjamin Van Roy. Epistemic neural networks. In *Neural Information Processing Systems (NeurIPS)*, 2023.

Theodore Papamarkou, Maria Skoularidou, Konstantina Palla, Laurence Aitchison, Julyan Arbel, David Dunson, Maurizio Filippone, Vincent Fortuin, Philipp Hennig, José Miguel Hernández-Lobato, et al. Position: Bayesian deep learning is needed in the age of large-scale ai. In *International Conference on Machine Learning (ICML)*, 2024.

Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. In *International Conference on Learning Representations (ICLR)*, 2022.

Bernhard W Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society: Series B (Methodological)*, 47, 1985.

Nicki Skafte, Martin Jorgensen, and Soren Hauberg. Reliable training and estimation of variance networks. *Neural Information Processing Systems (NeurIPS)*, 2019.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, 2010.

Andrew Stirn, Harm Wessels, Megan Schertzer, Laura Pereira, Neville Sanjana, and David Knowles. Faithful heteroscedastic regression with neural networks. In *Artificial Intelligence and Statistics (AISTATS)*, 2023.

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.

A Related Work **11**
A.1 Variance Prediction . . . 11
A.2 Epistemic Uncertainty and Bayesian Deep Learning . . . 11

B Variational Posteriors **12**
B.1 Inverse Gamma . . . 12
B.2 Log-Normal . . . 13

C Variational Inference for Homoscedastic Noise **13**
C.1 Variational Lower Bound and Approach . . . 13
C.2 Regression . . . 14
C.2.1 Training Objective . . . 14
C.2.2 Prediction . . . 14
C.3 Classification . . . 15
C.3.1 Training Objective . . . 15
C.3.2 Prediction . . . 16
C.4 Complexity and Parameterization . . . 16

D Heteroscedastic Noise Modeling **17**
D.1 Bayesian Last Layer Methods for Heteroscedastic Noise . . . 17
D.2 Variational Lower Bound for VBLL-Variance Networks . . . 17
D.2.1 Regression . . . 18
D.2.2 Classification . . . 18
D.3 Prediction . . . 18
D.4 Complexity and Parameterization . . . 18

E Training and Algorithmic Details **19**

F Experiment Details and Further Experiments **19**
F.1 Supervised Regression . . . 19
F.1.1 Motorcycle Dataset . . . 19
F.1.2 UCI Datasets . . . 19
F.2 Supervised Classification . . . 20
F.2.1 Half Moon Dataset . . . 20
F.2.2 CIFAR 10 and CIFAR 100 . . . 20
F.3 Bayesian Optimization . . . 21

Appendix A. Related Work

Uncertainty quantification has been an active topic of research since the reemergence of neural networks in the early 2010s (Gal and Ghahramani, 2016; Blundell et al., 2015; Lakshminarayanan et al., 2017; Papamarkou et al., 2024). Two notable lines of work have developed: variance prediction, in which the model directly outputs predictive uncertainty, and forms of uncertainty quantification in the network parameters (including Bayesian deep learning). While the former is conceptually simple and is capable of more expressive representation of aleatoric uncertainty, it is limited in its ability to quantify epistemic uncertainty (Der Kiureghian and Ditlevsen, 2009; Kendall and Gal, 2017). Similarly, many approaches toward epistemic uncertainty quantification rely on assumed likelihoods that do not sufficiently characterize aleatoric uncertainty.

A.1. Variance Prediction

Variance prediction networks aim to, beyond standard point predictions, predict variance terms (Nix and Weigend, 1994; Bishop and Quazaz, 1996). In regression, this typically corresponds to predicting the mean and variance of a Gaussian likelihood. In classification, this typically corresponds to modeling logit noise (Collier et al., 2020, 2021). Concretely, these networks typically parameterize the variance via

$$\log \Sigma_i = \boldsymbol{m}_i^\top \boldsymbol{\phi}(\boldsymbol{x}) \tag{12}$$

where $\boldsymbol{m}_i$ is a (point estimate) last layer, and are trained via maximum likelihood. These models broadly fall into the set of heteroscedastic models, although there is a conflation of epistemic and aleatoric uncertainty in the predicted variance (Kendall and Gal, 2017).

While heteroscedastic models based on variance prediction are conceptually simple and easy to implement, they have several limitations. First, training these models is often difficult as high predictive uncertainty on data effectively down-samples them, resulting in poor performance in mean estimation (Skafte et al., 2019; Seitzer et al., 2022; Stirn et al., 2023). Concretely, in regression, the loss term associated with the mean is weighted by the inverse variance, and thus data points with high predictive noise are weighted less in gradient computation than those with less predictive noise. Seitzer et al. (2022) propose re-weighting the data by the covariance (with a stop-grad operation applied) to mitigate the impacts of this effective down-sampling. We also take this approach in regression models.

A second important improvement was proposed by Skafte et al. (2019), who (instead of predicting the log variance) output the parameters of an inverse Gamma distribution, and marginalizing these parameters to yield a Student-t predictive distribution. This parameterization has strong similarities to the approaches taken in this work, and similarly aims to address the shortcomings of aleatoric uncertainty prediction via epistemic noise modeling.

A.2. Epistemic Uncertainty and Bayesian Deep Learning

Variance prediction networks are limited in their ability to quantify epistemic uncertainty. Like standard networks, the predictive behavior far from data is hard to anticipate, and thus predictive uncertainty may be smaller (or larger) than desired. Thus, epistemic uncertainty quantification—even for models that characterize aleatoric uncertainty such as variance prediction models—are necessary. Moreover, the posterior inferred by approximate Bayesian

methods relies on the chosen likelihood. Thus, model misspecification in the form of assumed homoscedasticity can substantially harm posterior inference and the predictive performance of the model (Le et al., 2005; Lazaro-Gredilla and Titsias, 2011; Kersting et al., 2007).

A wide variety of epistemic uncertainty quantification methods for neural networks have been developed including variational methods (Blundell et al., 2015), ensemble-based methods (Gal and Ghahramani, 2016), ensembling (Lakshminarayanan et al., 2017), randomized priors (Osband et al., 2018, 2023), and many others. However, these methods are typically expensive and scale poorly to larger models.

In this work, we build on variational Bayesian last layers (VBLLs) as a scalable and effective Bayesian approach (Harrison et al., 2024). These models fit into a class of similar last layer uncertainty quantification approaches, including SNGP (Liu et al., 2022) and last layer Laplace approximation methods (Daxberger et al., 2021). For these methods, the last layer inference strategy relies on computing a last layer via approximate Bayesian linear regression after each epoch, and thus incorporating variance prediction (and epistemic uncertainty for this term) is challenging.

Appendix B. Variational Posteriors

In this section we discuss the two variational posteriors used in this paper for modeling noise covariance.

B.1. Inverse Gamma

Our first approach is an inverse Gamma distribution,

$$q(\Sigma_i) = \mathcal{IG}(\nu_i, \Psi_i) \tag{13}$$

where $\nu_i > 1, \Psi_i > 0$ are shape and scale parameters. This distribution is equivalent to a Gamma distribution for the inverse covariance. This distribution has several desirable properties. First, it is conjugate: in the Bayesian linear regression (BLR) model (with appropriately chosen priors), an inverse Gamma prior yields an inverse Gamma posterior. Additionally, within the BLR model, an inverse-Gamma posterior yields a t-distributed posterior predictive. While this approach is desirable for several reasons in the classical BLR setting, it is less suited to the heteroscedastic setting as we discuss in Section D. The (inverse) Gamma posterior also has a straightforward generalization for non-diagonal covariances in the (inverse) Wishart distribution. We note a few useful identities, which will be useful in developing our main training objectives:

$$\mathbb{E}[\Sigma_i^{-1}] = \nu_i \Psi_i^{-1} \tag{14}$$

$$\mathbb{E}[\log \Sigma_i] = \log(\Psi_i) - \psi(\nu_i) \tag{15}$$

where $\psi(\cdot)$ denotes the digamma function.

For this parameterization, we write our prior as $p(\Sigma_i) = \mathcal{IG}(\underline{\nu}_i, \underline{\Psi}_i)$. The KL divergence between inverse Gammas is a standard result, with

$$\mathrm{KL}(q(\Sigma_i) \| p(\Sigma_i)) = \underline{\nu}_i \log \frac{\Psi_i}{\underline{\Psi}_i} - \log \frac{\Gamma(\nu_i)}{\Gamma(\underline{\nu}_i)} + (\nu_i - \underline{\nu}_i)\psi(\nu_i) - (\Psi_i - \underline{\Psi}_i)\frac{\nu_i}{\Psi_i} \tag{16}$$

where $\Gamma(\cdot)$ is the gamma function.

B.2. Log-Normal

A second approach to the variational posterior is a log-Normal[2]

$$q(\Sigma_i) = \log \mathcal{N}(\mu_i, C_i) \tag{17}$$

which is defined by a mean μ_i and variance $C_i > 0$. This choice of variational posterior may initially seem like an odd one: we lose the favorable conjugacy properties of the inverse Gamma posterior, and gain little in return. However, as we see in Section D, the log-Normal approach has substantial benefits for heteroscedastic modeling. We have the same identities,

$$\mathbb{E}[\Sigma_i^{-1}] = \exp(-\mu_i + \frac{1}{2}C_i) \tag{18}$$

$$\mathbb{E}[\log \Sigma_i] = \mu_i. \tag{19}$$

For this posterior specification, the prior is also log-Normal and written as $p(\Sigma_i) = \log \mathcal{N}(\underline{\mu}_i, \underline{C}_i)$. The KL divergence between log-Normals is equivalent to the KL divergence between their corresponding Normal distributions.

Appendix C. Variational Inference for Homoscedastic Noise

In this section, we derive a variational last layer objective with noise inference in the homoscedastic case. First, we will lay out the structure of the variational lower bounds we develop throughout the paper. We then describe two variational posterior design options, and prior choices. We will then define the training objectives and the resulting posterior predictive distribution.

C.1. Variational Lower Bound and Approach

We begin by writing the lower bound on the marginal likelihood for arbitrarily specified noise covariance Σ, and then discuss outcomes for inverse Gamma priors/variational posteriors. We exclude dependence on the features $\boldsymbol{\phi}$, which we assume fixed in the derivation. Point estimates for the feature weights are learned via stochastic gradient descent on the ELBO. Generally, we will choose noise priors with distributions that match the variational posterior for tractability, although this is not strictly necessary.

We structure our variational posterior as

$$q(W, \Sigma) = \prod_{i=1}^{N_y} q(\boldsymbol{w}_i \mid \Sigma_i) q(\Sigma_i) \tag{20}$$

with $q(\boldsymbol{w}_i \mid \Sigma_i) = \mathcal{N}(\bar{\boldsymbol{w}}_i, S_i \Sigma_i)$, following standard results in Bayesian regression. The variational lower bound is

$$\log p(Y \mid X) \geq \sum_{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}} \mathbb{E}_{q(W,\Sigma)}[\log p(\boldsymbol{y} \mid \boldsymbol{x}, W, \Sigma)] - \mathrm{KL}(q(W, \Sigma) \| p(W, \Sigma)) \tag{21}$$

By independence assumptions in the prior and variational posterior, we have

$$\mathrm{KL}(q(W, \Sigma) \| p(W, \Sigma)) = \sum_{i=1}^{N_y} (\mathrm{KL}(q(\Sigma_i) \| p(\Sigma_i)) + \mathbb{E}_{q(\Sigma_i)}[\mathrm{KL}(q(\boldsymbol{w}_i \mid \Sigma_i) \| p(\boldsymbol{w}_i \mid \Sigma_i))]). \tag{22}$$

2. Note that if $z \sim \mathcal{N}(\mu, \sigma^2)$ then $\exp(z) \sim \log \mathcal{N}(\mu, \sigma^2)$.

The expectation of the first KL term is straightforward, and is

$$\mathbb{E}_{q(\Sigma_i)}[\mathrm{KL}(q(\boldsymbol{w}_i \mid \Sigma_i) \| p(\boldsymbol{w}_i \mid \Sigma_i))] = \frac{1}{2}\left(\mathrm{logdet}\underline{S}_i - \mathrm{logdet}S_i - N_\phi + \mathrm{tr}(\underline{S}_i^{-1}S_i)\right) \tag{23}$$

$$+ \frac{1}{2}\left(\mathbb{E}_{q(\Sigma_i)}[\Sigma_i^{-1}](\boldsymbol{w}_i - \bar{\boldsymbol{w}}_i)\underline{S}_i^{-1}(\boldsymbol{w}_i - \bar{\boldsymbol{w}}_i)\right) \tag{24}$$

The expectation of Σ_i^{-1} can be computed via the identities in Section 2. The KL divergence between prior and variational posteriors for Σ_i can also be computed via the identities in Section 2. Thus, the unresolved aspect in evaluating the variational lower bound is computing the expected likelihood $\mathbb{E}_{q(W,\Sigma)}[\log p(\boldsymbol{y} \mid \boldsymbol{x}, W, \Sigma)]$.

C.2. Regression

We first discuss the likelihood term and prediction in the regression setting.

C.2.1. Training Objective

The predictive likelihood term in the ELBO is

$$\mathbb{E}_{q(W,\Sigma)}[\log p(\boldsymbol{y} \mid \boldsymbol{x}, W, \Sigma)] = \sum_{i=1}^{N_y} \mathbb{E}_{q(\Sigma_i)}[\mathbb{E}_{q(\boldsymbol{w}_i|\Sigma_i)}[\log p(\boldsymbol{y}_i \mid \boldsymbol{x}, \boldsymbol{w}_i, \Sigma_i)]] \tag{25}$$

where the inner expectation is computed as in Harrison et al. (2024) as

$$\mathbb{E}_{q(\boldsymbol{w}_i|\Sigma_i)}[\log p(\boldsymbol{y}_i \mid \boldsymbol{x}, \boldsymbol{w}_i, \Sigma_i)] = \log \mathcal{N}(\boldsymbol{y}_i \mid \bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi}, \Sigma_i) - \frac{1}{2}\boldsymbol{\phi}^\top S_i \boldsymbol{\phi} \tag{26}$$

$$= -\frac{1}{2}(\Sigma_i^{-1}(\boldsymbol{y}_i - \bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi})^2 + \log \Sigma_i + \boldsymbol{\phi}^\top S_i \boldsymbol{\phi}). \tag{27}$$

To evaluate the outer expectation,

$$\mathbb{E}_{q(W,\Sigma)}[\log p(\boldsymbol{y} \mid \boldsymbol{x}, W, \Sigma)] = -\frac{1}{2}\sum_{i=1}^{N_y} \mathbb{E}_{q(\Sigma)}[\Sigma_i^{-1}(\boldsymbol{y}_i - \bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi})^2 + \log \Sigma_i + \boldsymbol{\phi}^\top S_i \boldsymbol{\phi}] \tag{28}$$

we leverage the identities for $\mathbb{E}[\Sigma_i^{-1}]$ and $\mathbb{E}[\log \Sigma_i]$ described in the previous section.

C.2.2. Prediction

We now discuss computing the predictive distribution

$$p(\boldsymbol{y}_i \mid \boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{w}_i, \Sigma_i)}[p(\boldsymbol{y}_i \mid \boldsymbol{x}, \boldsymbol{w}_i, \Sigma_i)]. \tag{29}$$

Inverse Gamma. For the inverse Gamma variational posterior, we can exploit standard conjugacy results. In particular, the posterior predictive for each row i is multivariate t-distributed,

$$p(\boldsymbol{y}_i \mid \boldsymbol{x}) = t_{2\nu_i}(\bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi}, \frac{\Psi}{\nu_i}(1 + \boldsymbol{\phi}^\top S \boldsymbol{\phi})) \tag{30}$$

Log-Normal. For the log-Normal variational posterior, we must turn instead to a Monte Carlo approximation. We will sample

$$\hat{\Sigma}_i \sim q(\Sigma_i) \tag{31}$$

for all i, and for which sampling is straight-forward by simply sampling from a normal and exponentiating. Given this realized sample, we can marginalize over the last layer yielding predictive

$$p(\boldsymbol{y}_i \mid \boldsymbol{x}) = \mathcal{N}(\bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi}, \hat{\Sigma}_i(\boldsymbol{\phi}^\top S_i \boldsymbol{\phi} + 1)) \tag{32}$$

C.3. Classification

We consider a classification model of the form

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \text{softmax}(W\boldsymbol{\phi}(\boldsymbol{x}) + \boldsymbol{\varepsilon}) \tag{33}$$

where the addition of the aleatoric noise $\boldsymbol{\varepsilon}$ is optional. Our model exactly matches the regression model, with the only difference being the softmax. Critically, we again assume Σ is diagonal, with inverse-Gamma distributed diagonal entries.

C.3.1. Training Objective

We write the likelihood

$$\mathbb{E}_{q(W,\Sigma)}[\log p(\boldsymbol{y} \mid \boldsymbol{x}, W, \Sigma)] = \boldsymbol{y}^\top \bar{W}\boldsymbol{\phi} - \mathbb{E}_{q(\Sigma)}\mathbb{E}_{q(W|\Sigma)}[\text{LSE}_i(\boldsymbol{w}_i^\top \boldsymbol{\phi} + \boldsymbol{\varepsilon}_i)] \tag{34}$$

where we assume $\boldsymbol{y}$ is a one-hot encoding of the class labels, and lower bound the $\text{LSE}(\cdot)$ term. It may be tempting to exploit the relatively standard (Blei and Lafferty, 2007) approach for variational multinominal logistic regression to construct a lower bound on the log-sum-exp term for the inner expectation (as is used in Harrison et al. (2024)), yielding

$$-\mathbb{E}_{q(W|\Sigma)}[\text{LSE}_i(\boldsymbol{w}_i^\top \boldsymbol{\phi} + \boldsymbol{\varepsilon}_i)] \geq -\mathbb{E}_{q(\Sigma)}[\log \sum_i \exp(\bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi} + \frac{\Sigma_i}{2}(1 + \boldsymbol{\phi}^\top S_i \boldsymbol{\phi})]. \tag{35}$$

It is possible to further exchange the expectation and the negative log/sum terms, yielding

$$-\mathbb{E}_{q(\Sigma)}[\text{LSE}_i(\bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi} + \frac{\Sigma_i}{2}(1 + \boldsymbol{\phi}^\top S_i \boldsymbol{\phi})] \geq -\log \sum_i \mathbb{E}_{q(\Sigma)}[\exp(\bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi} + \frac{\Sigma_i}{2}(1 + \boldsymbol{\phi}^\top S_i \boldsymbol{\phi})]. \tag{36}$$

However, the expectation on the RHS generally does not exist[3]. Thus, we propose two possible approaches, each of which we discuss below.

Semi-Monte Carlo. First, we may turn to sampling. We sample (via reparameterization trick, available in standard automatic differentiation packages) gamma random variables to compute a Monte Carlo approximation to the expectation in (35).

Reduced Knowles-Minka. Instead of directly using the bound on the log-sum-exp used in Harrison et al. (2024), we can instead use the main result from Knowles and Minka (2011), where (applying their result to our chosen parameterization)

$$-\mathbb{E}_{q(W|\Sigma)}[\text{LSE}_i(\boldsymbol{w}_i^\top \boldsymbol{\phi} + \boldsymbol{\varepsilon})] \geq -\frac{1}{2}\sum_i a_i^2(\boldsymbol{\phi}^\top S_i \boldsymbol{\phi} + 1)\Sigma_i - \text{LSE}_i(\bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi} + (\frac{1}{2} - a_i)(\boldsymbol{\phi}^\top S_i \boldsymbol{\phi} + 1)\Sigma_i) \tag{37}$$

where a_i are variational parameters that are typically optimized to be maximally tight. Note that choosing $a_i = 0$ for all i exactly recovers (35). To yield a tractable bound for the outer expectation with respect to Σ, it is necessary to remove it from inside the log-sum-exp term. So, we choose

$$a_i = \frac{1}{2} - \frac{\alpha_i}{\Sigma_i} \tag{38}$$

3. This can be seen by the non-existence of the moment generating function of the inverse gamma distribution.

and plugging in yields

$$-\mathbb{E}[\text{LSE}_i(\boldsymbol{w}_i^\top \boldsymbol{\phi} + \boldsymbol{\varepsilon})] \geq -\frac{1}{2}\sum_i (\frac{1}{2} - \frac{\alpha_i}{\Sigma_i})^2(\boldsymbol{\phi}^\top S_i \boldsymbol{\phi} + 1)\Sigma_i - \text{LSE}_i(\bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi} + \alpha_i(\boldsymbol{\phi}^\top S_i \boldsymbol{\phi} + 1)) \tag{39}$$

$$= -\frac{1}{2}\sum_i (\frac{\Sigma_i}{4} - \alpha_i + \frac{\alpha_i^2}{\Sigma_i})(\boldsymbol{\phi}^\top S_i \boldsymbol{\phi} + 1) - \text{LSE}_i(\bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi} + \alpha_i(\boldsymbol{\phi}^\top S_i \boldsymbol{\phi} + 1)) \tag{40}$$

which is analytically tractable for the outer expectation over Σ, yielding (for the inverse Gamma variational posterior)

$$\mathbb{E}_{q(W,\Sigma)}[\log p(\boldsymbol{y} \mid \boldsymbol{x}, W, \Sigma)] \geq \boldsymbol{y}^\top \bar{W}\boldsymbol{\phi} - \text{LSE}_i(\bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi} + \alpha_i(\boldsymbol{\phi}^\top S_i \boldsymbol{\phi} + 1)) \tag{41}$$

$$- \frac{1}{2}\sum_i \left(\frac{\Psi_i}{4(\nu_i - 1)} - \alpha_i + \frac{\alpha_i^2 \nu_i}{\Psi_i}\right)(\boldsymbol{\phi}^\top S_i \boldsymbol{\phi} + 1). \tag{42}$$

We have several options with α_i; setting $\alpha_i = \frac{1}{2}$ results in the first two terms exactly matching the standard VBLL objective (Harrison et al., 2024). Choosing $\alpha_i = 0$ yields the remarkably simple overall bound

$$-\mathbb{E}_{q(W|\Sigma)}[\text{LSE}_i(\boldsymbol{w}_i^\top \boldsymbol{\phi} + \boldsymbol{\varepsilon})] \geq -\frac{1}{8}\sum_i (\boldsymbol{\phi}^\top S_i \boldsymbol{\phi} + 1)\Sigma_i - \text{LSE}_i(\bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi}). \tag{43}$$

Other options can be chosen for learning α_i's. We can treat them as hyperparameters, in which it is convenient to set $\alpha_i = \alpha$ to reduced the number of parameters, and it can be swept over. Alternatively, because the bound holds for any α_i, they can be learned as model parameters together with the other model parameters. Knowles and Minka (2011) propose an iterative update that exploits convexity with respect to α_i's—while better optimization schemes exploiting convexity are possible, we will not investigate them in this paper.

C.3.2. Prediction

For prediction, we again turn to Monte Carlo approximation within the hierarchical model, combined with local reparameterization, and sample $\hat{\Sigma}_i$ from the variational posterior, and compute

$$\hat{\boldsymbol{z}}_i \sim \mathcal{N}(\bar{\boldsymbol{w}}_i^\top \boldsymbol{\phi}, \hat{\Sigma}_i(1 + \boldsymbol{\phi}^\top S_i \boldsymbol{\phi})) \tag{44}$$

for each logit element. Variance reduction schemes are possible for prediction, but they are beyond the scope of this paper.

C.4. Complexity and Parameterization

We follow the parameterization presented in Harrison et al. (2024), which proposes to parameterize the variational posterior $q(\bar{\boldsymbol{w}}_i, \Sigma_i S_i)$ via a simple unconstrained tensor for $\bar{\boldsymbol{w}}_i$ and with either a strictly positive diagonal or Cholesky-decomposed representation for S_i. Recall the inverse Gamma variational posterior is $q(\Sigma_i) = \mathcal{IG}(\nu_i, \Psi_i)$ with $\nu_i > 1, \Psi_i > 0$. Enforcing strict positivity is done via exponentiating tensors (that are unconstrained). Lower bounds on parameter values are similarly accomplished by adding the offset.

The addition of variational inference in the homoscedastic case adds minimal additional complexity. The only additional tensors added are those of the variational posterior. Thus,

we add $2N_y$ parameters but do not estimate a point estimate for Σ_i, and thus add only N_y parameters.

Appendix D. Heteroscedastic Noise Modeling

In this section we discuss approaches to modeling heteroscedastic noise that also quantifies the epistemic uncertainty associated with aleatoric noise prediction.

D.1. Bayesian Last Layer Methods for Heteroscedastic Noise

Our approach to aleatoric noise modeling in this work builds on the standard VBLL model. In particular, we use a VBLL variational posterior for the last layer in a covariance predictive model of the form

$$\log \Sigma_i = \boldsymbol{m}_i^\top \boldsymbol{\phi}(\boldsymbol{x}) \tag{45}$$

with $\boldsymbol{m}_i \sim q(\boldsymbol{m}_i) = \mathcal{N}(\bar{\boldsymbol{m}}, Z)$ (and similarly choose a Normal prior $\boldsymbol{m}_i \sim \mathcal{N}(\underline{\boldsymbol{m}}, \underline{Z})$).

Given this structure, $\log \Sigma_i$ is Normally distributed and Σ_i is log-Normally distributed, as

$$\Sigma_i \sim \log \mathcal{N}(\bar{\boldsymbol{m}}_i^\top \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x})^\top Z \boldsymbol{\phi}(\boldsymbol{x})) \tag{46}$$

We choose a prior of the same structure over $\boldsymbol{m}$, which we write $p(\boldsymbol{m})$. We established in Section 2 that the log-Normal covariance distribution is a reasonable one. While it allows for tractable variational objectives, it does not allow analytical marginalization for the predictive distribution. We will note identities which are critical in our development, which build upon those presented in Section 2:

$$\mathbb{E}_{q(\boldsymbol{m}_i)}[\Sigma_i] = \exp(\bar{\boldsymbol{m}}_i^\top \boldsymbol{\phi}(x) + \frac{1}{2}\boldsymbol{\phi}(x)^\top Z_i \boldsymbol{\phi}(x)) \tag{47}$$

$$\mathbb{E}_{q(\boldsymbol{m}_i)}[\Sigma_i^{-1}] = \exp(-\bar{\boldsymbol{m}}_i^\top \boldsymbol{\phi}(x) + \frac{1}{2}\boldsymbol{\phi}(x)^\top Z_i \boldsymbol{\phi}(x)) \tag{48}$$

$$\mathbb{E}_{q(\boldsymbol{m}_i)}[\log \Sigma_i] = \bar{\boldsymbol{m}}_i \tag{49}$$

We can compare this modeling approach to learning point estimates of Σ in the standard VBLL model, or to learning a standard variance prediction network. If we set $N_\phi = 1$ and set $\boldsymbol{\phi} = 1$, we exactly recover the VBLL noise estimation scheme under a log-Normal prior. Thus, this heteroscedastic noise scheme represents a strict generalization of the standard VBLL model. If we replace the variational posterior over $\boldsymbol{m}_i$ with a point estimate, we recover a standard variance prediction model.

D.2. Variational Lower Bound for VBLL-Variance Networks

We can obtain tractable variational objectives by combining our variance parameterization with the variational objectives obtained in the last section. Note that each input $\boldsymbol{x}$ induces a Σ and W in our generative model. For T training examples, our variational posterior is

$$q(W_{1:T}, M \mid X) = q(M) \prod_{t=1}^{T} q(W_t \mid M, \boldsymbol{x}_t) \tag{50}$$

where t indexes training data. Following the previous section and indexing rows with i (and dropping data indexing), the terms in this factorized variational posterior can be written

$$q(W \mid M, \boldsymbol{x}) = \prod_{i=1}^{N_y} q(\boldsymbol{w}_i \mid \boldsymbol{m}_i, \boldsymbol{x}) \tag{51}$$

with
$$q(\boldsymbol{w}_i \mid \boldsymbol{m}_i, \boldsymbol{x}) = \mathcal{N}(\bar{\boldsymbol{w}}_i, \Sigma_i(\boldsymbol{m}_i, \boldsymbol{x})S_i) \tag{52}$$
and
$$q(M) = \prod_{i=1}^{N_y} q(\boldsymbol{m}_i). \tag{53}$$
With this variational posterior, we have
$$\log p(Y \mid X) \geq \mathbb{E}_{q(M)}[\log p(Y \mid X, M)] - \mathrm{KL}(q(M) \| p(M)) \tag{54}$$
$$\geq \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \mathbb{E}_{q(W|M,\boldsymbol{x})q(M)}[\log p(\boldsymbol{y} \mid \boldsymbol{x}, W, M)] \tag{55}$$
$$- \mathrm{KL}(q(M) \| p(M)) - \sum_{\boldsymbol{x}\in\mathcal{D}} \mathbb{E}_{q(M)}[\mathrm{KL}(q(W \mid M, \boldsymbol{x}) \| p(W \mid M, \boldsymbol{x}))].$$
There are two KL terms in (55). The term for $q(M)$ is a straightforward KL between Gaussians, and factorizes over the dimensionality of $\boldsymbol{y}$. The second KL term is
$$\sum_{\boldsymbol{x}\in\mathcal{D}} \sum_{i=1}^{N_y} \frac{1}{2}(\mathbb{E}_{q(\boldsymbol{m}_i)}[\Sigma_i^{-1}](\bar{\boldsymbol{w}} - \underline{\bar{\boldsymbol{w}}})^\top \underline{S}_i^{-1}(\bar{\boldsymbol{w}} - \underline{\bar{\boldsymbol{w}}}) + \mathrm{tr}(\underline{S}_i^{-1} S_i) - \log \frac{\det S_i}{\det \underline{S}_i} - N_\phi) \tag{56}$$
which is tractable via (48).

Practically, the relative weight of the second KL term is much larger as it is the sum of T terms. In practice, we will scale this second KL term by $1/T$ (to match the relative weight of the first KL term), which improves performance. This larger weighting factor results from sharing the parameters of the variational posterior for each W (specifically $\bar{W}, S$).

D.2.1. Regression

To compute the likelihood term in (55) in the regression case, we build upon the objective as written in (28), and apply the developed identities for $\mathbb{E}[\Sigma_i^{-1}]$ and $\mathbb{E}[\log \Sigma_i]$.

D.2.2. Classification

To compute the likelihood term for the classification case, we use the previously developed objective in (40) and apply the expressions for $\mathbb{E}[\Sigma_i^{-1}]$ and $\mathbb{E}[\Sigma_i]$, yielding an analytically tractable heteroscedastic classification model.

D.3. Prediction

Because the variational posterior for the noise covariance is log-Normal, we lose conjugacy for prediction in the regression case. Thus, for both regression and classification, we turn to sampling. For the regression case, we sample Σ from the variational posterior and then marginalize $\boldsymbol{w}$ as in standard VBLL regression models. In classification, we are forced to sample the noise covariance Σ, and then sample from the conditional distribution over logits under the variational posterior.

D.4. Complexity and Parameterization

The approach to heteroscedastic modeling fundamentally relies on parameterizing two VBLL heads: one for the mean, and one for the noise covariance. Both are parameterized as with standard VBLLs, which can have a dense, diagonal, or low-rank covariance

structure, and thus the complexity of this layer is twice the complexity of a standard VBLL layer. However, we note that VBLLs (with appropriately chosen covariance parameterizations) have comparable complexity to standard neural network layers, and thus the added computational cost of adding heteroscedasticity is comparable to adding an additional layer to a neural network.

Appendix E. Training and Algorithmic Details

We train our t-VBLL and Het-VBLL models with standard neural network optimization strategies. Concretely, we jointly train the parameters of the variational posterior together with the the neural network features ϕ. In this work we only train feature point estimates, but it is also possible to train variational posteriors for features via Bayes-by-backprop (as discussed in Harrison et al. (2024)), or train ensembles of models with t-VBLL or Het-VBLL heads.

We train via minibatch optimization, with the sums over the data replaced with (re-weighted) expectations over minibatches as in Blundell et al. (2015); Harrison et al. (2024). Note that this implies that, in the heteroscedastic model, the KL term for $q(M)$ should be weighted by one over the dataset size, whereas the other KL term and the likelihood term are averaged over the dataset.

The models presented in this paper can be used for full model training, or for a phase of training in more complex network training pipelines. For example, a t- or Het-VBLL head can be trained as a linear probe on pre-trained features, or may be used as a head for a fine-tuned model. Practically, VBLL-based models train slightly more slowly than standard neural networks due to (typically) being more heavily regularized, and thus training VBLL-based heads in a second phase of training often accelerates training versus training from scratch with VBLL heads.

Appendix F. Experiment Details and Further Experiments

F.1. Supervised Regression

F.1.1. Motorcycle Dataset

For the experiment in Figure 1 (left) on the data set provided by Silverman (1985), we use for both the VBLL and three Het-VBLL three hidden layers with a feature dimension of $N_\phi = 64$, ELU activation functions, and a prior scale of one. As optimizer, we use AdamW (Loshchilov and Hutter, 2017). For the Het-VBLL, we choose a learning rate of $3e{-}4$ and a weight decay of $1e{-}5$. For the standard VBLLs, we choose a learning rate of $3e{-}3$ and a weight decay of 0. For both, we use gradient clipping at 1. We train both surrogates for 2000 epochs with a batch size of 32.

F.1.2. UCI Datasets

Results on UCI datasets (Dua and Graff, 2017) are shown in Tables 3 and 4. These experiments match the setting used in Watson et al. (2021) and Harrison et al. (2024), and we compare against their baselines. In particular, we use two hidden layer MLPs of width 50, and use a batch size of 32 (except for Power in which we use batch size 256). We use

Table 3: Results for UCI regression tasks.

	Boston		Concrete		Energy	
	NLL (↓)	RMSE (↓)	NLL (↓)	RMSE (↓)	NLL (↓)	RMSE (↓)
t-VBLL	2.51 ± 0.10	$\mathbf{2.87 \pm 0.28}$	$\mathbf{2.95 \pm 0.08}$	$\mathbf{4.70 \pm 0.36}$	1.19 ± 0.13	0.85 ± 0.11
Het-VBLL	$\mathbf{2.35 \pm 0.14}$	3.14 ± 0.53	3.05 ± 0.13	5.20 ± 0.35	1.17 ± 0.11	1.53 ± 0.23
VBLL	2.55 ± 0.06	$\mathbf{2.92 \pm 0.12}$	3.22 ± 0.07	5.09 ± 0.13	1.37 ± 0.08	0.87 ± 0.04
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	$\mathbf{2.97 \pm 0.03}$	$\mathbf{4.80 \pm 0.18}$	4.80 ± 0.18	0.50 ± 0.02
MAP	2.60 ± 0.07	3.02 ± 0.17	3.04 ± 0.04	$\mathbf{4.75 \pm 0.12}$	1.44 ± 0.09	0.53 ± 0.01
RBF GP	$\mathbf{2.41 \pm 0.06}$	$\mathbf{2.83 \pm 0.16}$	3.08 ± 0.02	5.62 ± 0.13	$\mathbf{0.66 \pm 0.04}$	$\mathbf{0.47 \pm 0.01}$
Dropout	$\mathbf{2.36 \pm 0.04}$	$\mathbf{2.78 \pm 0.16}$	$\mathbf{2.90 \pm 0.02}$	$\mathbf{4.45 \pm 0.11}$	1.33 ± 0.00	0.53 ± 0.01
Ensemble	2.48 ± 0.09	$\mathbf{2.79 \pm 0.17}$	3.04 ± 0.08	4.55 ± 0.12	$\mathbf{0.58 \pm 0.07}$	$\mathbf{0.41 \pm 0.02}$
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
BBB	$\mathbf{2.39 \pm 0.04}$	$\mathbf{2.74 \pm 0.16}$	2.97 ± 0.03	4.80 ± 0.13	$\mathbf{0.63 \pm 0.05}$	0.43 ± 0.01

Table 4: Further results for UCI regression tasks.

	Power		Wine		Yacht	
	NLL (↓)	RMSE (↓)	NLL (↓)	RMSE (↓)	NLL (↓)	RMSE (↓)
t-VBLL	2.75 ± 0.02	3.83 ± 0.07	0.91 ± 0.05	0.62 ± 0.03	0.99 ± 0.34	0.87 ± 0.21
Het-VBLL	$\mathbf{2.73 \pm 0.03}$	3.75 ± 0.09	0.92 ± 0.07	$\mathbf{0.61 \pm 0.03}$	0.74 ± 0.50	1.94 ± 1.03
VBLL	$\mathbf{2.73 \pm 0.01}$	$\mathbf{3.68 \pm 0.03}$	1.02 ± 0.03	0.65 ± 0.01	1.29 ± 0.17	0.86 ± 0.17
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	$\mathbf{0.45 \pm 0.01}$	$\mathbf{0.56 \pm 0.05}$	$\mathbf{0.17 \pm 0.03}$	$\mathbf{0.40 \pm 0.03}$
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
Ensemble	$\mathbf{2.70 \pm 0.01}$	$\mathbf{3.59 \pm 0.04}$	0.95 ± 0.01	0.63 ± 0.01	0.35 ± 0.07	0.83 ± 0.08
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
BBB	2.77 ± 0.01	3.86 ± 0.04	0.95 ± 0.01	0.63 ± 0.01	1.43 ± 0.17	1.10 ± 0.11

AdamW (Loshchilov and Hutter, 2017) with a learning rate of $1e-3$ and weight decay of $1e-2$ on the hidden layers. We ran 10 seeds for each dataset. For more details on the experimental setting, we refer the reader to Harrison et al. (2024).

F.2. Supervised Classification

F.2.1. Half Moon Dataset

For the two-moon data set in Figure 2, we use `sklearn` to generate the data and set the noise level to 0.25. We introduce input-dependent label noise by flipping 20% of the labels between $[0.5, 1]$ for feature 1. For all baselines, we use the same backbone configuration consisting of two hidden layers with 128 neurons ($N_\phi = 128$) and ELU activations. Further, we choose a learning rate of $1e-3$ and a weight decay of $1e-4$ and train all models for 1000 epochs. For the standard VBLL and the t-VBLL, we use a prior scale of 1 and a Wishart scale of 1. For the Het-VBLL, we choose a noise prior scale of 0.1 and a prior scale of 1. For better visibility, we present the results of Figure 2 again in Figure 4.

F.2.2. CIFAR 10 and CIFAR 100

For the CIFAR 10 and CIFAR 100 datasers, we compare our models against other last-layer methods, swapping out the classification head of a pretrained and frozen Wide ResNet-28-10 network following Liu et al. (2022). These experiments, again, match the settings used

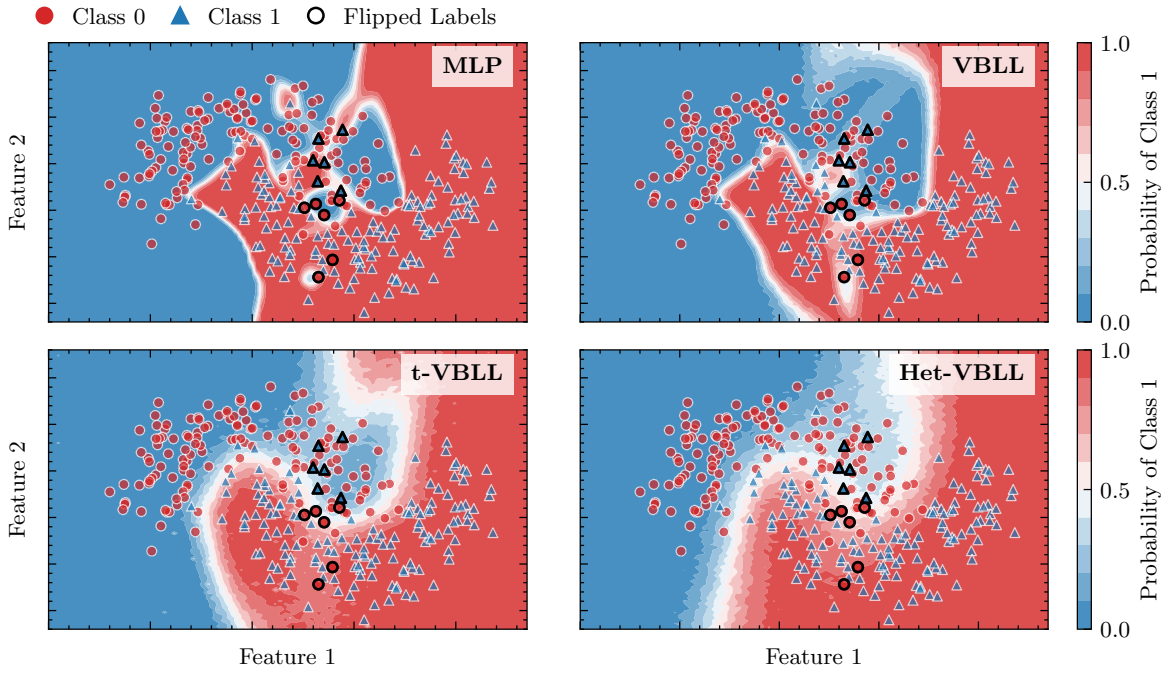

Figure 4: Same results on the noisy two-moon data set under label noise as in Figure 2. in Harrison et al. (2024), and we compare against their reported baselines. In particular we compare against standard VBLL (Harrison et al., 2024) and last-layer laplace (Daxberger et al., 2021) methods, as well as the results for the standard (vanilla) network. Performance is evaluated through accuracy, calibration error (ECE), negative log-likelihood (NLL) and out-of-distribution AUROC for both near and far OOD detection capabilities. For CIFAR-10, we assess near OOD performance using CIFAR-100 as the out-of-distribution dataset and vice versa for the CIFAR-100 OOD evaluation. In both cases, we utilize Street View House Numbers (SVHN) (Netzer et al., 2011) as a far OOD dataset. For t-VBLL we set a prior scale of 1 and an inverse Gamma scale parameter of 10. For Het-VBLL we select a noise prior scale of 0.01 and prior scale of 10. All models are trained using the AdamW optimizer and a learning rate of 1e-3 with linear warmup.

F.3. Bayesian Optimization

All models are implemented using GPyTorch (Gardner et al., 2018) and BoTorch (Balandat et al., 2020). For the Student-t predictive of the t-VBLLs, we directly use the standard deviation of the predictive in UCB. For the Het-VBLL model, we construct this standard deviation by sampling ten Σ from the variational posterior as discussed in Sec. D.3 and then use their mean in the acquisition function. For the outliers, we add heavy-tailed noise from a zero mean Laplace distribution with a standard deviation of 0.5 (pest control) or 1 (Ackley) to the output of an experiment.

For all VBLL baselines, we use 3 hidden layers with 64 neurons and ELU activations and set the maximum training epochs to 10000. For all models but the Het-VBLL, we use a patience of 100 (Brunzema et al., 2025). For the GP, we use constraints on the lengthscales

as $\ell_i \in [0.005, 4]$ (Eriksson et al., 2019) and optimize the lengthscales at iteration through maximum likelihood estimation.