A APPENDIX

A.1 IMPLEMENTATION DETAILS

To assess our proposed model's predictive performance and uncertainty estimation capabilities, we conducted experiments on synthetic two moons data (Pedregosa et al., 2011), a mixture of Gaussians, the CIFAR-100 dataset (Krizhevsky et al., 2009), and the Imagenet dataset (Deng et al., 2009). We compare against a standard deterministic ResNet model as a baseline (He et al., 2016), against the heteroscedastic method (Collier et al., 2020; 2021) and the SNGP (Liu et al., 2020) (which form the basis for our combined model) and against the recently proposed Posterior Network model (Charpentier et al., 2020), which also offers distance-aware uncertainties, similarly to the SNGP. We used the same backbone neural network architecture for all models, which was a fully-connected ResNet for the synthetic data, a WideResNet18 on CIFAR and a ResNet50 in Imagenet.

For most baselines, we used the hyperparameters from the `uncertainty_baselines` library (Nado et al., 2021). On CIFAR, we trained our HetSNGP with a learning rate of 0.1 for 300 epochs and used $R = 6$ factors for the heteroscedastic covariance, a softmax temperature of $\tau = 0.5$ and $S = 5000$ Monte Carlo samples. On Imagenet, we trained with a learning rate of 0.07 for 270 epochs and used $R = 15$ factors, a softmax temperature of $\tau = 1.25$ and $S = 5000$ Monte Carlo samples. We implemented all models in TensorFlow in Python and trained on Tensor Processing Units (TPUs) in the Google Cloud.

We train all Imagnet-21k models for 90 epochs with batch size 1024 on 8×8 TPU slices. We train using the Adam optimizer with initial learning rate of 0.001 using a linear learning rate decay schedule with termination point 0.00001 and a warm-up period of 10,000 steps. We train using the sigmoid cross-entropy loss function and L2 weight decay with multiplier 0.03. The heteroscedastic method uses a temperature of 0.4, 1,000 Monte Carlo samples and $R = 50$ for the low rank approximation. HetSNGP has the same heteroscedastic hyperparameters except the optimal temperature is 1.5. For SNGP and HetSNGP the GP covariance is approximated using the momentum scheme presented in Liu et al. (2020) with momentum parameter 0.999.

A.2 LAPLACE APPROXIMATION

In this section, we will derive the Laplace posterior in Eq. (5). The derivation follows mostly from the sections 3.4 and 3.5 in Rasmussen & Williams (2006).

First note that the log posterior of $\boldsymbol{\beta}_c$ given the data is

$$\log p(\boldsymbol{\beta}_c \,|\, \boldsymbol{x}, y) = \log p(y \,|\, \boldsymbol{\beta}_c) + \log p(\boldsymbol{\beta}_c) - Z \tag{7}$$

where Z is a normalization constant that does not depend on $\boldsymbol{\beta}_c$. Following Rasmussen & Williams (2006), we will denote the unnormalized log posterior as

$$\Psi(\boldsymbol{\beta}_c) = \log p(y \,|\, \boldsymbol{\beta}_c) + \log p(\boldsymbol{\beta}_c) \tag{8}$$

Recall that the first term is the likelihood and the second term is our prior from Eq. (4).

The Laplace approximation now approximates the posterior with a local second-order expansion around the MAP solution, that is

$$p(\boldsymbol{\beta}_c \,|\, \boldsymbol{x}, y) \approx \mathcal{N}(\hat{\boldsymbol{\beta}}_c, \boldsymbol{\Lambda}^{-1}) \tag{9}$$

with the MAP solution $\hat{\boldsymbol{\beta}}_c = \arg\max_{\boldsymbol{\beta}_c} \Psi(\boldsymbol{\beta}_c)$ and the Hessian $\boldsymbol{\Lambda} = -\nabla^2 \Psi(\boldsymbol{\beta}_c)|_{\boldsymbol{\beta}_c = \hat{\boldsymbol{\beta}}_c}$.

The MAP solution can be found using standard (stochastic) gradient descent, while the Hessian is given by

$$\begin{aligned}
\nabla^2 \Psi(\boldsymbol{\beta}_c) &= \nabla^2 \log p(y \,|\, \boldsymbol{\beta}_c) + \nabla^2 \log p(\beta_c) \\
&= \nabla_{\boldsymbol{\beta}} (\nabla_{\boldsymbol{u}} \log p(y \,|\, \boldsymbol{u}) \nabla_{\boldsymbol{\beta}} \boldsymbol{u}) - \boldsymbol{I}_m \\
&= \nabla_{\boldsymbol{\beta}} (\nabla_{\boldsymbol{u}} \log p(y \,|\, \boldsymbol{u}) \boldsymbol{\Phi}) - \boldsymbol{I}_m \\
&= \boldsymbol{\Phi}^\top \nabla^2_{\boldsymbol{u}} \log p(y \,|\, \boldsymbol{u}) \boldsymbol{\Phi} - \boldsymbol{I}_m \\
&= -W \boldsymbol{\Phi}^\top \boldsymbol{\Phi} - \boldsymbol{I}_m
\end{aligned}$$

where we used the chain rule and the fact that $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{\beta}$ and W is a diagonal matrix of point-wise second derivatives of the likelihood, that is, $W_{ii} = -\nabla^2 \log p(y_i \mid \boldsymbol{u}_i)$ (Rasmussen & Williams, 2006). For instance, in the case of the logistic likelihood, $W_{ii} = \boldsymbol{p}_i (1 - \boldsymbol{p}_i)$, where $\boldsymbol{p}_i$ is a vector of output probabilities for logits $\boldsymbol{u}_i$. To get the Hessian at the MAP, we then just need to compute this quantity for $\hat{\boldsymbol{u}} = \boldsymbol{\Phi}\hat{\boldsymbol{\beta}}$.

The approximate posterior is therefore

$$p(\boldsymbol{\beta}_c \mid \boldsymbol{x}, y) \approx \mathcal{N}(\hat{\boldsymbol{\beta}}_c, (W\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \boldsymbol{I}_m)^{-1}) \tag{10}$$

where the precision matrix can be computed over data points (recovering Eq. (5)) as

$$\boldsymbol{\Lambda} = \boldsymbol{I}_m + \sum_{i=1}^{N} \boldsymbol{p}_i(1 - \boldsymbol{p}_i)\boldsymbol{\Phi}_i\boldsymbol{\Phi}_i^\top \tag{11}$$