

## A. Multilabel classification

The extension to the multilabel case is simple. Each label is treated independently conditional on the latent  $\mathbf{u}$ , the  $\arg \max$  is replaced with the  $\mathbb{1}$  indicator function which can be approximated with a temperature parameterized sigmoid.

$$\begin{aligned}
p_c &= P(y_c = 1|\mathbf{x}) \\
&= P(\mathbb{1}\{u(\mathbf{x})_c > 0\}) \\
&= \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \Sigma(\mathbf{x}))} [\mathbb{1}\{u(\mathbf{x})_c > 0\}] \\
&= \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \Sigma(\mathbf{x}))} \left[ \lim_{\tau \rightarrow 0} \text{sigmoid}_{\tau} u(\mathbf{x})_c \right] \\
&\approx \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \Sigma(\mathbf{x}))} \left[ \text{sigmoid}_{\tau} u(\mathbf{x})_c \right], \tau > 0 \\
&\approx \frac{1}{S} \sum_{i=1}^S \text{sigmoid}_{\tau} u^{(i)}(\mathbf{x})_c, \mathbf{u}^{(i)}(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \Sigma(\mathbf{x})).
\end{aligned} \tag{5}$$

## B. Experimental details

**Imagenet ILSVRC12.** We train a ResNet-152v2 [19] for 90 and 270 epochs. Stochastic gradient descent with momentum factor = 0.9 and initial learning rate of 0.1 is used as the optimizer. The learning rate is decayed by a factor of  $10\times$  at 30, 60 and 80 epochs for the 90 epoch training and 90, 180 and 240 epochs for the 270 epoch training. During the first 5 epochs the learning rate follows a linear warm-up schedule.  $L2$  regularization with weight  $10^{-3}$  is used. Standard data augmentation is used during training, an Inception crop followed by resizing to  $224 \times 224$  and left-right horizontal flipping. All inputs are scaled to the  $[-1, 1]$  range. Models are trained on  $4 \times 4$  Google Cloud TPUv3s with a batch size of 1,024. For all heteroscedastic models the optimal sigmoid temperature of 0.9 is tuned on the validation set. A rank 15 low-rank approximation to the covariance matrix is used for our method. 10,000 MC samples are used at train and eval time for the heteroscedastic methods.

**WebVision 1.0.** We train an Inception-ResNetv2 [40] for 95 epochs. A distributed asynchronous stochastic gradient descent optimizer with momentum factor = 0.9 and initial learning rate of 0.1 is used. The learning rate is decayed by a factor of  $10\times$  at 40, and 80 epochs. During the first 2 epochs the learning rate follows a linear warm-up schedule.  $L2$  regularization with weight  $4 \times 10^{-5}$  is used. Efficientnet [41] preprocessing is used during training. Models are trained on 32 NVIDIA v100 GPUs with a per GPU batch size of 64. For all heteroscedastic models the optimal softmax temperature of 0.9 is chosen based on the Imagenet ILSVRC12 optimal hyperparameters. A rank 15 low-rank approximation to the covariance matrix is used for our method. 1,000 MC sam-

ples are used at train and eval time for the heteroscedastic methods.

**Imagenet-21k.** We train a ResNet-152v2 [19] for 90 epochs. Stochastic gradient descent with momentum factor = 0.9 and initial learning rate of 0.1 is used as the optimizer. The learning rate is decayed by a factor of  $10\times$  at 30, 60 and 80 epochs. During the first 5000 steps the learning rate follows a linear warm-up schedule.  $L2$  regularization with weight  $1 \times 10^{-2}$  is used for the parameters mapping from the shared representation space to the low-rank covariance matrix,  $L2$  regularization with weight  $3 \times 10^{-3}$  is used for all other parameters. Standard data augmentation is used during training, an Inception crop followed by resizing to  $224 \times 224$  and left-right horizontal flipping. All inputs are scaled to the  $[-1, 1]$  range. Models are trained on  $8 \times 8$  Google Cloud TPUv3s with a batch size of 1,024. Gradients are clipped to a maximum  $L2$  of 1.0. For all heteroscedastic models the optimal sigmoid temperature of 0.15 is tuned on the validation set. A rank 50 low-rank approximation to the covariance matrix is used for our method. 1,000 MC samples are used at train and eval time for the heteroscedastic methods.

**JFT.** We train a ResNet-50v2 [19] for 30 epochs. Stochastic gradient descent with momentum factor = 0.9 and initial learning rate of 0.03 is used as the optimizer. The learning rate is decayed by a factor of  $10\times$  at 10, 20 and 25 epochs. During the first 5000 steps the learning rate follows a linear warm-up schedule.  $L2$  regularization with weight  $10^{-3}$  is used. Standard data augmentation is used during training, an Inception crop followed by resizing to  $224 \times 224$  and left-right horizontal flipping. All inputs are scaled to the  $[-1, 1]$  range. Models are trained on  $16 \times 16$  Google Cloud TPUv3s with a batch size of 4,096. Gradients are clipped to a maximum  $L2$  of 1.0. For all heteroscedastic models the optimal sigmoid temperature of 0.15 is chosen based on the Imagenet-21k optimal hyperparameters. A rank 50 low-rank approximation to the covariance matrix is used for our method. 1,000 MC samples are used at train and eval time for the heteroscedastic methods.

## C. Analysis of the effect of the covariance matrix on the log-likelihood

We examine the effect of the covariance on the log-likelihood of our method, particularly in contrast to the homoscedastic log-likelihood. We make a 2<sup>nd</sup> order Taylor series approximation to the log-likelihood of our method and show that the approximation decomposes into a term which corresponds to the homoscedastic log-likelihood and 2<sup>nd</sup> order term which depends on the covariance matrix. We

first examine the case when the covariance matrix is diagonal and then turn to the full covariance case.

For brevity denote the softmax( $\mathbf{u}$ ) $_k = \frac{\exp(u_k)}{\sum_j \exp(u_j)}$  as  $s_k(\mathbf{u})$  which will sometimes abbreviate to  $s_k$  when the softmax argument is clear. For simplicity we drop the dependence on the softmax temperature.

**2<sup>nd</sup> order Taylor series approximation.** The second order Taylor series approximation to a single sample of the heteroscedastic output layer is given in Eq. (6).

$$s_k(W^\top \mathbf{x} + V\epsilon) \approx s_k(W^\top \mathbf{x}) + \nabla s_k(W^\top \mathbf{x})^\top V\epsilon + \frac{1}{2}\epsilon^\top V^\top \nabla^2 s_k(W^\top \mathbf{x}) V\epsilon \quad (6)$$

where,  $\epsilon \sim \mathcal{N}(0_K, I_{K \times K})$ .

Marginalizing over  $\epsilon$ , the second term vanishes as  $\mathbb{E}[\epsilon] = 0$ . Hence the second order Taylor series approximation to the likelihood is:

$$\mathbb{E}_\epsilon [s_k(W^\top \mathbf{x} + V\epsilon)] \approx s_k(W^\top \mathbf{x}) + \frac{1}{2}\text{tr}(\nabla^2 s_k(W^\top \mathbf{x}) V V^\top) \quad (7)$$

**Lemma C.1** *The Hessian of  $s_k$ ,  $\mathbf{H}_{s_k}$ , has the following structure:*

$$\begin{bmatrix} \ddots & & & & \\ \cdots & s_k(1-s_k)(1-2s_k) & \cdots & -s_j s_k(1-2s_k) & \cdots \\ & \ddots & & & \\ & & \ddots & & \\ 2s_k s_i s_j & -s_j s_k(1-2s_k) & \cdots & -s_k s_j(1-2s_j) & \cdots \\ \cdots & & \cdots & & \ddots \end{bmatrix}$$

**Proof.** Note that the first derivative of the  $s_k$  falls into two cases:

$$\frac{\partial s_k(\mathbf{u})}{\partial u_k} = \begin{cases} -s_k s_j & k \neq j \\ s_k(1-s_k) & k = j. \end{cases} \quad (8)$$

The second derivatives breakdown into four cases:

**Case 1:**  $k \neq j$ ,  $\frac{\partial^2 s_k(\mathbf{u})}{\partial u_j \partial u_k} = -s_j s_k(1-2s_k)$

**Case 2:**  $\frac{\partial^2 s_k(\mathbf{u})}{\partial^2 u_k} = s_k(1-s_k)(1-2s_k)$

**Case 3:**  $i, j \neq k, i \neq j$ ,  $\frac{\partial^2 s_k(\mathbf{u})}{\partial u_i \partial u_j} = 2s_k s_i s_j$

**Case 4:**  $j \neq k$ ,  $\frac{\partial^2 s_k(\mathbf{u})}{\partial^2 u_j} = -s_k s_j(1-2s_j)$

### C.1. Diagonal covariance

Henceforth references to  $s_k$  correspond to  $s_k(W^\top \mathbf{x})$ .

First, assume that the covariance matrix  $V V^\top$  is diagonal with entries  $\sigma_1^2, \dots, \sigma_j^2, \dots, \sigma_K^2$ .

Substituting into Eq. (9) we get a special case of the approximate likelihood for the diagonal covariance case:

$$\mathbb{E}_\epsilon [s_k(W^\top \mathbf{x} + V\epsilon)] \approx s_k(1 - \frac{1}{2} \sum_{j \neq k} s_j(1-2s_j)\sigma_j^2 + \frac{1}{2}(1-s_k)(1-2s_k)\sigma_k^2) \quad (9)$$

With  $\log(1+t) \approx t$  the log-likelihood can be approximated as:

$$\log \mathbb{E}_\epsilon [s_k(W^\top \mathbf{x} + V\epsilon)] \approx \log(s_k) - \frac{1}{2} \sum_{j \neq k} s_j(1-2s_j)\sigma_j^2 + \frac{1}{2}(1-s_k)(1-2s_k)\sigma_k^2 \quad (10)$$

The  $\log s_k = \log s_k(W^\top \mathbf{x})$  term in Eq. (10) is precisely the standard log-likelihood term of a homoscedastic model so the remaining  $\frac{1}{2} \sum_{j \neq k} s_j(1-2s_j)\sigma_j^2 + \frac{1}{2}(1-s_k)(1-2s_k)\sigma_k^2$  term accounts for the approximate effect of the diagonal covariance on the heteroscedastic log-likelihood relative to the homoscedastic model.

Note that:

$$\begin{aligned} \frac{\partial}{\partial \sigma_j^2} \frac{1}{2} \sum_{j \neq k} s_j(1-2s_j)\sigma_j^2 + \frac{1}{2}(1-s_k)(1-2s_k)\sigma_k^2 \\ \propto -s_j(1-2s_j) \end{aligned} \quad (11)$$

$$\begin{aligned} \frac{\partial}{\partial \sigma_k^2} \frac{1}{2} \sum_{j \neq k} s_j(1-2s_j)\sigma_j^2 + \frac{1}{2}(1-s_k)(1-2s_k)\sigma_k^2 \\ \propto (1-s_k)(1-2s_k) \end{aligned} \quad (12)$$

Fig. 3 shows these derivatives. We see that when we observe a label  $y = k$ , then to maximize the log-likelihood, if  $s_j > 0.5$  i.e., we are incorrectly classifying the example then the gradient forces  $\sigma_j^2$  to increase and vice versa, if  $s_j < 0.5$  the gradient encourages  $\sigma_k^2$  to decrease. In this way a noisy label  $y = k$  which may be confused with class  $j$  can be explained away by a high  $\sigma_j^2$  term.

Likewise if  $s_k > 0.5$  i.e., we are correctly classifying the example then the gradient forces  $\sigma_k^2$  to reduce and vice versa, when  $s_k < 0.5$  the gradient encourages  $\sigma_k^2$  to increase. Again the  $\sigma_k^2$  allows the model to explain away a noisy label  $y = k$  if the model assigns low probability to that label.

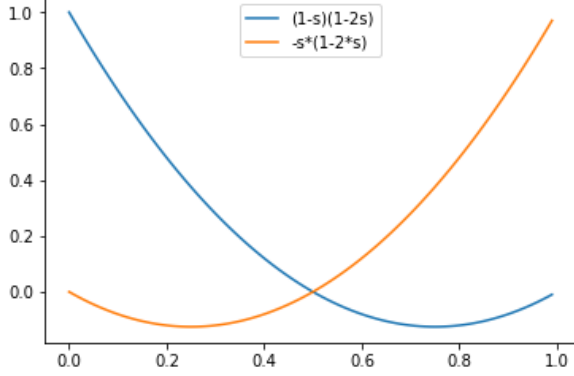


Figure 3: Behaviour of the coefficients that multiply the uncertainty factors in the second and third terms of the approximate likelihood, Eq. (10).

Interpreting the terms of the Taylor series relies on the 2nd order approximation being a reasonable approximation to the full method. We evaluate the efficacy of the approximation in the diagonal covariance case by training a ResNet-152 for 270 epochs using the 2nd order Taylor series approximation Eq. 9. All hyperparameters are equivalent to those in the main paper. See Table 8 for results.

The deterministic approximate method recovers 1.5% of the 2% gains in top-1 accuracy seen by the full stochastic method. This is evidence that the Taylor series approximation is a reasonable approximation to the full method.

Method	Top-1 Acc
Homoscedastic	76.7
Het. Diag	78.7
Het. Diag Taylor series approx	78.2

Table 8: Efficacy of 2nd order Taylor series approximation in diagonal covariance case for ResNet-152 trained on ILSVRC12 for 270 epochs.

## C.2. Full covariance

If we allow a full covariance matrix  $\Sigma = VV^\top$  then the approximate log-likelihood again breaks down into two terms, the first corresponding to the homoscedastic log-likelihood and the second corresponding to the effect of the heteroscedastic covariance matrix:

$$\begin{aligned} \log \mathbb{E}_\epsilon [s_k(W^\top \mathbf{x} + V\epsilon)] &\approx \log(s_k) \\ \frac{1}{2} \sum_{i \neq j, i, j \neq k}^K 2s_i s_j \Sigma_{ij} &+ \frac{1}{2} \sum_{j \neq k}^K -s_j(1-2s_k) \Sigma_{jk} \\ &+ \frac{1}{2} \sum_{j \neq k}^K -s_j(1-2s_j) \Sigma_{jj} + \frac{1}{2} (1-s_k)(1-2s_k) \Sigma_{kk} \end{aligned} \quad (13)$$

The diagonal covariance terms  $\Sigma_{kk}$  and  $\Sigma_{jj}$  have the same interpretation to the additional terms in the diagonal covariance log-likelihood Eq. (10). So we will focus on the off-diagonal terms  $\frac{1}{2} \sum_{i \neq j, i, j \neq k}^K 2s_i s_j \Sigma_{ij} + \frac{1}{2} \sum_{j \neq k}^K -s_j(1-2s_k) \Sigma_{jk}$ .

$\frac{\partial}{\partial \Sigma_{ij}} \frac{1}{2} \sum_{i \neq j, i, j \neq k}^K 2s_i s_j \Sigma_{ij} \propto s_i s_j$ , so covariance matrix entries are encouraged to be large and positive when the product  $s_i s_j$  is large i.e., when both classes  $i$  and  $j$  are assigned high probability by the homoscedastic term despite  $y = k$ . Classes  $i$  and  $j$  are encouraged to have a co-occurrence noise pattern.

Likewise, we note the derivative of the  $\frac{1}{2} \sum_{j \neq k}^K -s_j(1-2s_k) \Sigma_{jk}$  term w.r.t.  $\Sigma_{jk}$ ,  $\frac{\partial}{\partial \Sigma_{jk}} \frac{1}{2} \sum_{j \neq k}^K -s_j(1-2s_k) \Sigma_{jk} \propto -s_j(1-2s_k)$ . We see that to maximize the log-likelihood  $\Sigma_{jk}$  is encouraged to be large and positive when both  $s_j$  and  $s_k$  are large (top right corner of Figure 4) and highly negative when either  $s_j$  or  $s_k$  is large and the other is small. So when we observe a label  $y = k$  and classes  $j$  and  $k$  are assigned high probability then these classes are encouraged to have positively correlated noise i.e. to have a co-occurrence noise pattern. And when only one of the classes  $j$  or  $k$  has a high probability then the two classes are encouraged to have a substitution pattern in the noise.

## D. Sensitivity to number of MC samples

We test the sensitivity of our method to the number of MC samples. In Table 9 we vary the number of training and test MC samples on Imagenet ILSVRC12, using the same experimental setup as ResNet-152 270 epoch results in the main paper. As expected increasing the number of MC samples improves performance monotonically; however there are diminishing returns beyond 100 samples.

# MC Samples	1	10	100	1000	10000
Top-1 ACC	78.5	78.6	79.1	79.2	79.3

Table 9: Sensitivity of ResNet-152 trained for 270 epochs on ILSVRC12 to the number of MC samples.

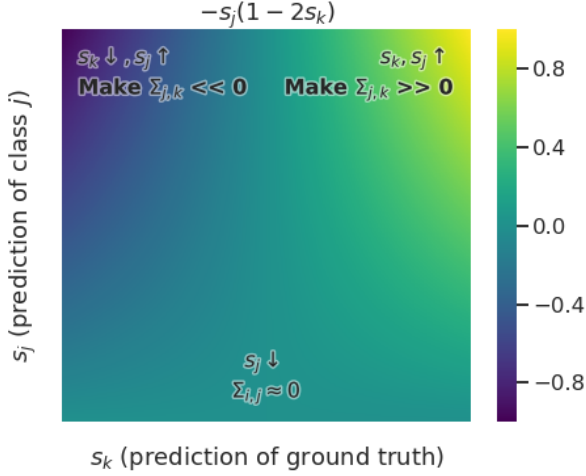


Figure 4: Heatmap of the derivatives of  $\sum_{j \neq k}^K -s_j(1 - 2s_k)\Sigma_{jk}$  w.r.t.  $\Sigma_{jk}$ . This is the effect of a small change in  $\Sigma_{jk}$  when  $j \neq k$  on the approximate log-likelihood, up to a constant multiplier, in the full covariance case. If the model is not predicting a high probability for (erroneous) class  $j$ , then there is no change to the noise term between  $j$  and the correct class  $k$ . Otherwise, the model learns anticorrelations or correlations between them.

## E. Equalizing the parameter count

Our method adds additional parameters to the network. We investigate in this section whether the quality gains of our method are due to these additional parameters. The homoscedastic ResNet-152 model for ILSVRC12 has 60,185,128 parameters as compared to the full heteroscedastic method with 15 factors: 92,969,128. By equalizing the number of parameters for the homoscedastic model, we demonstrate that the gains from the heteroscedastic method are not primarily due to the additional parameters.

In particular we add an additional Dense + ReLU layer with 11,500 output units before the logits layer in a ResNet-152. This brings the total parameter count of the homoscedastic model to 93,200,628. However these additional parameters only lead to a +0.4% increase in Top-1 accuracy, as opposed to +2.6% with the full heteroscedastic method, Table 10. Therefore the gains we observe from using our method on Imagenet ILSVRC are due primarily to the noise modelling and not the increase in parameter count.

## F. Training on ILSVRC12 with a sigmoid link function

Prior work argues that ILSVRC12 is in reality a multilabelled dataset i.e. contains multiple objects per images, but that we force it to be a multiclass classification by assigning one of these objects as the “primary” object [2]. The

Method	Top-1 Acc
Homoscedastic	76.7
Homoscedastic equal params	77.1
Het. Full (ours)	79.3

Table 10: Equalizing homoscedastic parameter count for ResNet-152 trained on ILSVRC12 for 270 epochs.

authors then show that training a homoscedastic model on ILSVRC12 with the sigmoid output activation i.e. as if the ILSVRC12 labels were multilabelled leads to performance improvements.

However from a probabilistic point of view this is an unsatisfying solution. Using a sigmoid link function may yield improved accuracy but the network does not yield a valid probability distribution over the ILSVRC12 labels.

We have seen that the off-diagonal covariance matrix entries can model substitution patterns between co-occurring objects in a single image. We argue that the gains observed by Beyer et al. [2] can be understood as arising from the misspecification of the homoscedastic model, due to the i.i.d. additive noise assumption.

In Table 11 we reproduce the gains from using the sigmoid link function for the homoscedastic model seeing the top-1 accuracy increase from 76.7% to 78.2%. However if we now train heteroscedastic model with the sigmoid link function, we see no significant gain compared to training the heteroscedastic model with a softmax link function. Thus the heteroscedastic noise model explains away the sigmoid effect. Removing the i.i.d. additive noise assumption has yielded a highly performant model which outputs a valid and well calibrated probability distribution over the observed labels.

Method	Top-1 Acc
Homoscedastic (softmax)	76.7 ( $\pm 0.13$ )
Homoscedastic (sigmoid)	78.2 ( $\pm 0.16$ )
Het. Full (ours - softmax)	79.3 ( $\pm 0.10$ )
Het. Full (sigmoid)	79.35 ( $\pm 0.05$ )

Table 11: Training a ResNet-152 on ILSVRC12 with softmax vs. sigmoid link function.