Correlated Input-Dependent Label Noise in Large-Scale Image Classification

Mark Collier  
Google AI  
markcollier@google.com

Basil Mustafa  
Google AI  
basilm@google.com

Efi Kokiopoulou  
Google AI  
kokiopou@google.com

Rodolphe Jenatton  
Google AI  
rjenatton@google.com

Jesse Berent  
Google AI  
jberent@google.com

Abstract

Large scale image classification datasets often contain noisy labels. We take a principled probabilistic approach to modelling input-dependent, also known as heteroscedastic, label noise in these datasets. We place a multivariate Normal distributed latent variable on the final hidden layer of a neural network classifier. The covariance matrix of this latent variable, models the aleatoric uncertainty due to label noise. We demonstrate that the learned covariance structure captures known sources of label noise between semantically similar and co-occurring classes. Compared to standard neural network training and other baselines, we show significantly improved accuracy on Imagenet ILSVRC 2012 79.3% (+ 2.6%), Imagenet-21k 47.0% (+ 1.1%) and JFT 64.7% (+ 1.6%). We set a new state-of-the-art result on WebVision 1.0 with 76.6% top-1 accuracy. These datasets range from over 1M to over 300M training examples and from $1^k$ classes to more than $21^k$ classes. Our method is simple to use, and we provide an implementation that is a drop-in replacement for the final fully-connected layer in a deep classifier.

1. Introduction

Image classification datasets with many classes and large training sets often have noisy labels [2, 30]. For example, Imagenet contains many visually similar classes that are hard for human annotators to distinguish [10, 2]. Datasets such as WebVision where labels are generated automatically by looking at co-occurring text to images on the Web, contain label noise as this automated process is not 100% reliable [30].

A wide range of techniques for classification under label noise already exist [29, 23, 16, 37, 24, 6, 9, 36, 18]. When an image is mis-labeled it is more likely that it gets confused with other related classes, rather than a random class [2]. Therefore it is important to take inter-class correlation into account when modelling label noise in image classification.

We take a principled probabilistic approach to modelling label noise. We assume a generative process for noisy labels with a multivariate Normal distributed latent variable at the final hidden layer of a neural network classifier. The mean and covariance parameters of this Normal distribution are input-dependent (aka heteroscedastic), being computed from a shared representation of the input image. By modelling the inter-class noise correlations our method can learn which class pairs are substitutes or commonly co-occur, resulting in noisy labels. See Fig. (1) for an example of two Imagenet classes which our model learns have correlated label noise.

Figure 1: Spot the difference? An Appenzeller (left) and EntleBucher (right). Two visually similar Imagenet classes our method learns have highly correlated label noise (average validation set covariance of -0.24) given only the standard Imagenet ILSVRC12 training labels.

We evaluate our method on four large-scale image classification datasets, Imagenet ILSVRC12 and Imagenet-21k [10], WebVision 1.0 [30] and JFT [21]. These datasets range from over 1M training examples (ILSVRC12) to 300M training examples (JFT) and from 1k classes to over 21k classes. We demonstrate improved accuracy and negative log-likelihood on all datasets relative to (a) standard neural network training, (b) methods which only model the diagonal of the covariance matrix and (c) methods from the noisy labels literature.

We evaluate the effect of our probabilistic label noise model on the representations learned by the network. We show that our method, when pre-trained on JFT, learns image
representations which transfer better to the 19 datasets from the Visual Task Adaptation Benchmark (VTAB) [47].

Contributions. In summary our contributions are:

1. A new method which models inter-class correlated label noise and scales to large-scale datasets.
2. We evaluate our method on four large-scale image classification datasets, showing significantly improved performance compared to standard neural network training and diagonal covariance methods.
3. We demonstrate that the learned covariance matrices model correlations between semantically similar or commonly co-occurring classes.
4. On VTAB our method learns more general representations which transfer better to 19 downstream tasks.

2. Method

In many datasets label noise is not uniform across the input space, some types of examples have more noise than others. We build upon prior work on probabilistic modelling of noisy labels [25, 9] by assuming a heteroscedastic latent variable generative process for our labels. This generative process leads to two main challenges while computing its resulting likelihood: (a) the intractable marginalization over the latent variables which we estimate via Monte Carlo integration and (b) an arg max in the generative process which we approximate with a temperature parameterized softmax.

Generative process. Suppose there is some latent vector of utility \( u(x) \in \mathbb{R}^K \), where \( K \) is the number of classes associated with each input \( x \). This utility is the sum of a deterministic reference utility \( \mu(x) \) and an unobserved stochastic component \( \epsilon \). A label is generated by sampling from the utility and taking the arg max, i.e. class \( c \) is the label if its associated utility is greater than the utility for all other classes \( \iff y = \arg\max_{j \in [K]} u_j(x) \):

\[
\begin{align*}
    u(x) &= \mu(x) + \epsilon \\
    p_c &= P(y = c | x) = P(\arg\max_{j \in [K]} u_j(x) = c) \\
    &= \int \mathbf{1}\{\arg\max_{j \in [K]} u_j(x) = c\} p(\epsilon) d\epsilon
\end{align*}
\] (1)

This generative process follows prior work in the econometrics, noisy labels and Gaussian Processes literature [42, 25, 9, 20, 45], discussed further in §3. First note that we choose each stochastic component to be distributed standard Gumbel independently, \( \epsilon_j \sim \text{i.i.d. } G(0,1) \forall j \), then the predictive probabilities \( p_c \) have a closed form solution that is precisely the popular softmax cross-entropy model used in training neural network classification models [42, 9]:

\[
p_c = P(\arg\max_{j \in [K]} u_j(x) = c) = \frac{\exp(\mu_c)}{\sum_{j=1}^{K} \exp(\mu_j)}
\] (2)

In other words, this generative process with Gumbel noise distribution is already an implicit standard assumption when training neural network classifiers. In (2), the independence and identical assumptions on the noise component is however too restrictive for applications with noisy labels:

1. **Identical**: for a particular input \( x \) some classes may have more noise than others, e.g., given an Imagenet image of a dog there may be high levels of noise on various different dog breeds but we may have high confidence that elephant classes are not present. Hence we need the level of noise to vary per class.

2. **Independence**: if one class has a high level of noise other related classes may have high/low levels of noise. In the above example there may be correlations in the noise levels between different dog breeds.

Our method breaks both the independence and identical assumptions by assuming that the noise term \( \epsilon(x) \) is distributed multivariate Normal, \( \epsilon(x) \sim \mathcal{N}(0, \Sigma(x)) \). Computing an input-dependent covariance matrix enables modelling of inter-class label noise correlations on a per image basis. We discuss more formally in Appendix C how going beyond an independent and identical noise model can lead to improved predictions in the presence of label noise. However it also raises a number of challenges:

First there is now no closed form solution for the predictive probabilities, Eq. (1). In order to address this, we transform the computation into an expectation and approximate using Monte Carlo estimation, Eq. (3).

Second, notice that the Monte Carlo estimate of Eq. (1) involves computing an arg max which makes gradient based optimization infeasible. We approximate the arg max with a temperature parameterized softmax \( \mathbb{S}_\tau \), Eq. (3):

\[
p_c = P(\arg\max_{j \in [K]} u_j(x) = c)
\]

\[
    = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \Sigma(x))} \left[ \mathbf{1}\{\arg\max_{j \in [K]} u_j(x) = c\} \right]
\]

\[
    = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \Sigma(x))} \left[ (\lim_{\tau \to 0} \text{softmax} u(x))_c \right]
\]

\[
    \approx \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \Sigma(x))} \left[ (\text{softmax} u(x))_c \right], \quad \tau > 0
\]

\[
    \approx \frac{1}{S} \sum_{i=1}^{S} (\text{softmax} u^{(i)}(x))_c, \quad u^{(i)}(x) \sim \mathcal{N}(\mu(x), \Sigma(x)).
\] (3)
Efficient parametrization of the covariance matrix. 

$\Sigma(x)$ is a $K \times K$ matrix which is a function of input $x$. The memory and computational resources required to compute the full $\Sigma(x)$ matrix are impractical for large-scale image classification datasets used in this paper (with $K$ up to $21k$ classes). We make a low-rank approximation to $\Sigma(x) = V(x) V(x)^T$ where $V(x)$ is a $K \times R$ matrix, $R << K$. To ensure the positive semi-definiteness of the covariance matrix, we compute a $K$ dimensional vector $d^2(x)$ which we add to the diagonal of $V(x) V(x)^T$. In order to sample from our noise distribution we first sample $\epsilon_K \sim {\mathcal N}(0_K, I_{K \times K})$, $\epsilon_R \sim {\mathcal N}(0_R, I_{R \times R})$, then $\epsilon = d(x) \odot \epsilon_K + V(x) \epsilon_R$, where $\odot$ denotes element-wise multiplication.

In practice we typically compute $V(x)$ as an affine transformation of a shared representation of $x$ computed by a deep neural network. Suppose that the dimension of this representation is $D$, then the number of parameters required to compute $V(x)$ is $O(DKR)$. For some datasets with many classes this is still impractically large. For example Imagenet-21k has 21,843 classes and in the below experiments we use $R = 50$ and a ResNet-152 which has a final layer representation with $D = 2048$. So computing $V(x)$ requires over $2.2B$ parameters, which dwarfs the total number of parameters in the rest of the network.

In order to further reduce the parameter and computational requirements of our method we introduce a parameter-efficient version which we use for datasets where the number of classes is too large (Imagenet-21k and JFT). We parameterize $V(x) = v(x) 1_R^T \odot V$ where $v(x)$ is a vector of dimension $R$, $1_R$ is a vector of ones of dimension $R$ and $V$ is a $K \times R$ matrix of learnable parameters which is not a function of $x$. Sampling the correlated noise component can be simplified, $V(x) \epsilon_R = (v(x) 1_R^T \odot V) \epsilon_R = v(x) \odot (V \epsilon_R)$. The total parameter count of this parameter-efficient version is $O(DK + KR)$ which typically reduces the memory and computational requirements dramatically for large-scale image classification datasets. For example, for Imagenet-21k the number of parameters required to compute $V(x)$ is $44.8M$, a $50 \times$ reduction. See Algorithm 1 for a full specification of our method.

### 3. Related work

#### 3.1. Heteroscedastic modelling

**Heteroscedastic regression.** Heteroscedastic regression is common in the Gaussian Processes [45] and econometrics literature [42]. Bishop and Quazaz [3] introduced a heteroscedastic regression model where a neural network outputs the mean and variance parameters of a Gaussian likelihood: $y \sim {\mathcal N}(\mu(x), \sigma(x)^2)$. The negative log-likelihood of the model is particularly amenable to interpretation, Eq. (4).

$$\frac{1}{N} \sum_{i=1}^{N} \frac{1}{2\sigma(x_i)^2} (y_i - \mu(x_i))^2 + \frac{1}{2} \log \sigma(x_i)^2. \quad (4)$$

We see that the squared error loss term for each example is weighted inversely to the predicted variance for that example, downweighting the importance of that example’s label and providing robustness to noisy labels. This heteroscedastic regression model has recently been applied to pixel-wise depth regression [25] and in deep ensembles [28].

**Heteroscedastic classification - diagonal covariance.** Kendall and Gal [25] extend the heteroscedastic regression model to classification by placing a multivariate Normal
distribution with diagonal covariance matrix over the softmax logits in a neural network classifier. They find that the method improves performance on image segmentation datasets which have noisy labels at object boundaries.

Closest to our methodology is the approach developed by Collier et al. [9] which we next describe. The authors reinterpret the method of Kendall and Gal [25] as an instance of the generative framework we follow in Eq. (1). They show that this connects the method to the discrete choice modelling econometrics literature where the temperature parameterized softmax smoothing function is known as the logit-smoothed accept-reject simulator [42, 32, 5]. The authors demonstrate that the softmax temperature does indeed control a bias-variance trade-off and that tuning the temperature results in different training dynamics, qualitatively improved predictions and significantly improved performance on image classification and image segmentation tasks. Both Kendall and Gal [25] and Collier et al. [9] always use a diagonal covariance matrix for the latent distribution.

The latent variable approach to heteroscedastic classification is also standard in the Gaussian Processes literature [20, 45]. A diagonal covariance matrix is used and a GP prior is placed on mean and log variance parameters. Again exact inference on the likelihood is intractable and different approximate inference methods are used [20].

**Heteroscedastic segmentation - full covariance.** Monteiro et al. [33] introduce Stochastic Segmentation Networks, a method for modelling spatially correlated label noise in image segmentation. Similar to Kendall and Gal [25] they place a multivariate Normal distribution over the softmax logits in an image segmentation network, but share a low-rank approximation to the full covariance matrix across all the pixels in the image, capturing spatially correlated noise. Unlike our method, Stochastic Segmentation Networks are a) only applied to medical image segmentation datasets, b) do not recognise the softmax as a temperature parameterized smoothing function w.r.t. an assumed generative process and therefore always implicitly use a softmax temperature of 1.0 and c) do not have a parameter-efficient version of the method to enable scaling to large output vocabularies.

**Discussion.** Our method combines the best of this prior work and enables scaling to large-scale image classification datasets. We follow Collier et al. [9] in assuming the generative process in Eq. (1). This gives the benefits of connecting the work to the existing discrete choice modelling econometrics literature and theory. However unlike Collier et al. [9] we use a low-rank approximation to a full (non-diagonal) covariance matrix in the latent distribution. In the below experiments, we demonstrate that our combination of (a) recognizing the importance of the softmax temperature in controlling a bias-variance trade-off and (b) modelling the inter-class noise correlations yields significantly improved performance compared with individually using (a) or (b). Our parameter-efficient method also enables scaling up correlated heteroscedastic noise models to a scale unprecedented by previous work e.g. Imagenet-21k and JFT.

### 3.2. Noisy labels

We provide a brief overview of some recent methods for training with noisy labels. Bootstrapping [36] sets the target label to be a linear combination of the ground truth label and the current model’s predictions. MentorNet [23] uses an auxiliary neural network, the MentorNet to output a scalar weighting for each potentially noisy example. MentorMix [24] adds mixup regularization [48] to the MentorNet approach. Co-teaching [18] jointly trains two neural networks. Each network makes predictions on a mini-batch and the small loss samples are then fed to the other network for learning. It is assumed that small loss examples are more likely to have clean labels. Cao et al. [6] propose a heteroscedastic adaptive regularization scheme which increases the regularization strength on high noise examples.

### 4. Experiments

Our main experiment is to evaluate our method on four large-scale image classification datasets and show significant performance improvements over baseline methods. We also provide qualitative analysis of the learned covariance matrices in §4.1. We analyse the effect of our method on the representations learned by the network in §4.2. Finally, in §4.3 we combine our method with Deep Ensembles [28] to yield a method with full predictive uncertainty.
As is standard, we report results on the official ILSVRC12 validation set. For our method we use a softmax temperature and covariance matrix rank as the above homoscedastic models, the additional parameters do not lead to more overfitting, see Appendix E.

Prior work has shown that neural networks fit cleanly labelled data points first and then fit examples with noisy labels [27]. The purpose of the 90 epoch ablation is to demonstrate the heteroscedastic models gain more from longer training schedules than the homoscedastic model. By overfitting less to noisy labels the heteroscedastic models can be trained for longer e.g., only our heteroscedastic sees improved top-5 accuracy from training for 270 epochs while the increase in top-1 accuracy from the longer training schedule increases from 0.2% from the homoscedastic model, to 0.4% for the diagonal covariance heteroscedastic model to 0.8% for our model. This demonstrates that despite the parameter count of the heteroscedastic models being higher than the homoscedastic models, the additional parameters do not lead to more overfitting, see Appendix E.

### Table 2: WebVision 1.0 results.

For het. models, \(\tau^* = 0.9\). Top-1 and top-5 accuracy and negative log-likelihood \(\pm 1\) standard deviation is reported for both the WebVision and ILSVRC12 validation sets. 5 runs from different random seeds are used for the homo/heteroscedastic methods, all other results are taken from the literature. † \(p\)-value < 0.05.

<table>
<thead>
<tr>
<th>Method</th>
<th>Webvision Top-1 Acc</th>
<th>Webvision Top-5 Acc</th>
<th>NLL</th>
<th>ILSVRC12 Top-1 Acc</th>
<th>ILSVRC12 Top-5 Acc</th>
<th>NLL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top-1 Acc</td>
<td>Top-5 Acc</td>
<td>NLL</td>
<td>Top-1 Acc</td>
<td>Top-5 Acc</td>
<td>NLL</td>
</tr>
<tr>
<td>Lee et al. [29]</td>
<td>69.1 ± (0.07)</td>
<td>86.7 ± (0.07)</td>
<td>-</td>
<td>61.0 ± (0.08)</td>
<td>82.0 ± (0.11)</td>
<td>1.49 ± (0.008)</td>
</tr>
<tr>
<td>Jiang et al. [23]</td>
<td>72.6 ± (0.07)</td>
<td>88.9 ± (0.07)</td>
<td>-</td>
<td>64.2 ± (0.08)</td>
<td>84.8 ± (0.07)</td>
<td>1.47 ± (0.004)</td>
</tr>
<tr>
<td>Guo et al. [16]</td>
<td>72.1 ± (0.07)</td>
<td>89.2 ± (0.07)</td>
<td>-</td>
<td>64.8 ± (0.08)</td>
<td>84.9 ± (0.07)</td>
<td>-</td>
</tr>
<tr>
<td>Saxena et al. [37]</td>
<td>67.5 ± (0.07)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Jiang et al. [24]</td>
<td>74.3 ± (0.07)</td>
<td>90.5 ± (0.07)</td>
<td>-</td>
<td>67.5 ± (0.08)</td>
<td>87.2 ± (0.07)</td>
<td>-</td>
</tr>
<tr>
<td>Cao et al. [6]</td>
<td>75.0 ± (0.07)</td>
<td>90.6 ± (0.07)</td>
<td>-</td>
<td>67.1 ± (0.07)</td>
<td>86.7 ± (0.07)</td>
<td>-</td>
</tr>
<tr>
<td>Homoscedastic</td>
<td>76.1† (±0.07)</td>
<td>91.2† (±0.07)</td>
<td>1.03† (±0.006)</td>
<td>67.0† (±0.08)</td>
<td>86.0† (±0.11)</td>
<td>1.49† (±0.008)</td>
</tr>
<tr>
<td>Het. Diag [9]</td>
<td>76.2† (±0.15)</td>
<td>91.4† (±0.08)</td>
<td>1.01† (±0.002)</td>
<td>67.3† (±0.10)</td>
<td>86.1† (±0.07)</td>
<td>1.47† (±0.004)</td>
</tr>
<tr>
<td>Het. Full (ours)</td>
<td>76.6 (±0.13)</td>
<td>92.1 (±0.09)</td>
<td>0.98 (±0.004)</td>
<td>68.6 (±0.17)</td>
<td>87.1 (±0.13)</td>
<td>1.41 (±0.010)</td>
</tr>
</tbody>
</table>

We provide code which implements our method as a TensorFlow Keras layer [1, 8], in the supplementary material. The layer is a drop-in replacement for the final layer of a classifier, requiring only a simple one line code change from:

```python
logits = tf.keras.layers.Dense(...) (x)
```

to

```python
logits = MCSofmaxDenseFA(...)(x).logits
```

No other changes, including to the loss function are required.

**Imagenet ILSVRC12.** Table 1 shows the results on Imagenet ILSVRC12, a dataset of over 1.2M training images with 1k classes. ILSVRC12 is known to have noisy labels [2]. Only one class can be present for each image. We train a ResNet-152 [19] for 90 and 270 epochs (further details in Appendix B). Hyperparameters are tuned on a validation set of 50,000 examples that we split off from the training set. As is standard, we report results on the official ILSVRC12 validation set. For our method we use a softmax temperature of 0.9 and covariance matrix rank of 15.

When trained for 270 epochs our method has a validation set top-1 accuracy of 79.3% statistically significantly better than the baseline models based on an unpaired two-tailed t-test. We compare to a homoscedastic baseline (standard neural network training), against which our method improves the top-1 accuracy by 2.6%. Compared to the diagonal covariance method we see a smaller improvement of 0.6% for top-1 accuracy, suggesting that much of the gain is from the diagonal covariance matrix entries, but that the off-diagonal terms give further improvements. We note that we are the first to evaluate the heteroscedastic diagonal model on ILSVRC12. A sensitivity analysis to the number of MC samples is provided in Appendix D. An ablation study equalizing the number of parameters in the heteroscedastic and homoscedastic models is provided in Appendix E.
shows the test set results for Imagenet-$21k$.

We use the parameter-efficient version of our method, §2, when training on WebVision. Our baseline homoscedastic method provides gains over standard neural network training and modelling the full covariance matrix provides further improvements in gAP for both datasets. On Imagenet-$21k$ and JFT, the additional improvement from modelling the full covariance matrix is more marginal than for JFT. We include an ablation to demonstrate the importance of the smoothing function temperature parameter. For both datasets, the performance of the diagonal method is significantly degraded at $\tau = 1.0$, which corresponds to the model of Kendall and Gal [25], compared to $\tau^* = 0.15$. We note that Collier et al. [9] have already empirically demonstrated the temperature controls a bias-variance trade-off in the training dynamics of the diagonal covariance heteroscedastic model.

<table>
<thead>
<tr>
<th>Method</th>
<th>Imagenet-$21k$ gAP</th>
<th>Imagenet-$21k$ NLL</th>
<th>JFT gAP</th>
<th>JFT NLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homoscedastic</td>
<td>45.91 (±0.14)</td>
<td>3.651 (±0.010)</td>
<td>63.11 (±0.22)</td>
<td>20.121 (±0.090)</td>
</tr>
<tr>
<td>Heteroscedastic Diag $\tau = 1.0$ [9]</td>
<td>45.91 (±0.06)</td>
<td>3.641 (±0.003)</td>
<td>63.11 (±0.03)</td>
<td>19.951 (±0.022)</td>
</tr>
<tr>
<td>Heteroscedastic Diag $\tau^* = 0.15$ [9]</td>
<td>46.81 (±0.04)</td>
<td>3.631 (±0.002)</td>
<td>64.11 (±0.07)</td>
<td>19.611 (±0.030)</td>
</tr>
<tr>
<td>Heteroscedastic PE $\tau^* = 0.15$ (ours)</td>
<td><strong>47.0</strong> (±0.08)</td>
<td><strong>3.62</strong> (±0.005)</td>
<td><strong>64.7</strong> (±0.06)</td>
<td><strong>19.34</strong> (±0.026)</td>
</tr>
</tbody>
</table>

Table 3: Imagenet-$21k$ and JFT results for heteroscedastic and homoscedastic models. Heteroscedastic PE, is the parameter-efficient version of our method. The test set global average precision (gAP) and negative log-likelihood ± 1 standard deviation is reported. 5 runs from different random seeds are used. † p-value < 0.05.

detailed results. Our method improves upon the best previously published WebVision top-1 accuracy by 1.6% and by 1.1% over the best published ILSVRC12 top-1 accuracy, when training on WebVision. Our baseline homoscedastic and diagonal heteroscedastic methods are strong relative to the previously published WebVision results, perhaps due to our longer than typical 95 epoch training schedule. We note however there is no standard training schedule for WebVision and our experiments showed that none of the models converge with shorter training schedules.

**Multilabel datasets.** We can also apply our method to multilabel datasets which may have more than one class in each image. The same latent variable formulation can be used but with temperature parameterized sigmoid smoothing function (see Appendix A). Imagenet-$21k$ and JFT are two large-scale multilabel image classification datasets.

Imagenet-$21k$ is a larger version of the standard ILSVRC-2012 Imagenet benchmark [10, 26, 2, 9]. It has over 12.8 million training images with 21,843 classes. No standard train/test split is provided, so we use 4% of the dataset as a validation set and a further 4% as the test set.

JFT-300M [21, 7, 39, 26] is a dataset introduced by Hinton et al. [21] with over 300M training images and validation and test sets with 50,000 images. JFT has over 17k classes and each image can have more than one class (average 1.89 per image). The labels were collected automatically, with 20% of them estimated to be noisy [39].

For Imagenet-$21k$ we train a Resnet-152 [19] for 90 epochs. Whereas for JFT we train a Resnet-50 [19] for 30 epochs. Further experimental setup details in Appendix B. Sigmoid temperature of 0.15 is used for the heteroscedastic methods. For our method the covariance matrix rank is set to 50 and as the number of classes is large for both datasets, we use the parameter-efficient version of our method, §2.

Table 3 shows the test set results for Imagenet-$21k$ and JFT. Global average precision (gAP), the average precision over all classes, is the metric of interest. The diagonal covariance heteroscedastic method with tuned sigmoid temperature provides gains over standard neural network training and modelling the full covariance matrix provides further improvements in gAP for both datasets. On Imagenet-$21k$

<table>
<thead>
<tr>
<th>Class A</th>
<th>Class B</th>
<th>Avg. Cov.</th>
</tr>
</thead>
<tbody>
<tr>
<td>partridge</td>
<td>ruffed grouse, partridge, Bonasa umbellus</td>
<td>-0.46</td>
</tr>
<tr>
<td>projectile, missile</td>
<td>missile</td>
<td>-0.44</td>
</tr>
<tr>
<td>mailot, tank suit</td>
<td>mailot</td>
<td>-0.42</td>
</tr>
<tr>
<td>screen, CRT screen</td>
<td>monitor</td>
<td>-0.37</td>
</tr>
<tr>
<td>tape player</td>
<td>cassette player</td>
<td>-0.34</td>
</tr>
<tr>
<td>Welsh springer spaniel</td>
<td>Blenheim spaniel</td>
<td>0.28</td>
</tr>
<tr>
<td>standard schnauzer</td>
<td>miniature schnauzer</td>
<td>0.27</td>
</tr>
<tr>
<td>French bulldog</td>
<td>Boston bull, Boston terrier</td>
<td>0.27</td>
</tr>
<tr>
<td>EntleBucher standard schnauzer</td>
<td>racket, racquet</td>
<td>-0.26</td>
</tr>
</tbody>
</table>

Table 4: Class pairs with the top-10 absolute covariance, averaged over the ILSVRC12 validation set.

4.1. Qualitative analysis of learned covariances

We can examine cases where our heteroscedastic model makes the correct prediction but the standard homoscedastic model is incorrect. Particularly we are interested in understanding whether there is structure in the covariance matrix of our method which helps explain the correct prediction. Note that we can reconstruct the full covariance matrix as $\Sigma(x) = diag(d(x)^2) + V(x)V(x)^T$.

In Table 5, we list 8 cases when the homoscedastic prediction has the highest absolute covariance with the correct
Table 5: 8 ILSVRC12 test set examples where the heteroscedastic model is correct, the homoscedastic model is incorrect and the absolute covariance between the heteroscedastic and homoscedastic prediction is the largest of all classes to the heteroscedastic prediction.

<table>
<thead>
<tr>
<th>Image</th>
<th>Predictions &amp; covariance</th>
</tr>
</thead>
</table>
| ![cowboy hat](image) | Het pred: cowboy hat   
Hom pred: ten-gallon hat, cowboy boot  
Covariance: -4.26 |
| ![castle](image) | Het pred: castle   
Hom pred: palace  
Covariance: -1.17 |
| ![elephant](image) | Het pred: African elephant, Loxodonta africana   
Hom pred: Indian elephant, Elephas maximus  
Covariance: 0.78 |
| ![tractor](image) | Het pred: tractor   
Hom pred: harvester, reaper  
Covariance: 1.37 |
| ![keyboard](image) | Het pred: computer keyboard, keypad   
Hom pred: space bar  
Covariance: 1.82 |
| ![clock](image) | Het pred: analog clock   
Hom pred: wall clock  
Covariance: 2.46 |
| ![monitor](image) | Het pred: monitor   
Hom pred: desktop computer  
Covariance: 3.07 |

We now conduct a simple analysis to show that the above examples are not anecdotal but that the learned covariance matrices are structured. For one homoscedastic and one heteroscedastic model selected from the previous Imagenet ILSVRC12 experiments there are 3,565 out of 50,000 validation set images where our model makes a correct prediction but the homoscedastic model is incorrect. Figure (2) shows a histogram of the rank of the sorted absolute covariance between the homoscedastic prediction and the correct class. Clearly the class the homoscedastic model has incorrectly predicted, is much more likely to have a high absolute covariance to the correct class in our learned covariance matrix. Assuming that the incorrect homoscedastic prediction is sometimes a plausible label then we see that the heteroscedastic model has learned to associate the noise on the correct class with the noise on plausible alternatives. This is consistent with our analysis of the 2nd order Taylor series approximation to our model’s log-likelihood, Appendix C. We expect to see covariances of commonly confused class pairs to be strengthened during training.

4.2. Transfer of learned image representations

Often large-scale image classification datasets are used to pre-train representations which are fine-tuned on smaller specialized datasets [47, 26, 39, 11, 35]. We wish to evaluate the transferability of the representations learned by our model.
We hypothesize that by overfitting less to noisy labels and learning a better model of the upstream pre-training distribution, our method should learn more general representations.

We test the JFT models on the VTAB [47]. VTAB consists of 19 unseen downstream classification datasets which cover a variety of visual domains and tasks. We evaluate using the VTAB1K protocol, where each model is fine-tuned on only 1000 datapoints for each downstream dataset. For all 19 datasets the fine-tuned model is homoscedastic, i.e., the heteroscedastic output layer is only ever used for upstream pre-training. The output layer of the network is removed and replaced with a untrained homoscedastic output layer for fine-tuning. For downstream fine-tuning we use the standard hyperparameters and data augmentation settings specified by Kolesnikov et al. [26]. The VTAB1K score is an average of the accuracy on all 19 datasets.

Table 6 shows VTAB1K scores. Our parameter-efficient heteroscedastic model, which captures correlations in the JFT label noise, improves the VTAB1K score by 0.88% over the homoscedastic baseline and by 0.22% over the heteroscedastic diagonal model. We stress that the downstream models are trained without a heteroscedastic output layer, so our experiment demonstrates that a model trained upstream on JFT with a heteroscedastic output layer learns representations which transfer better than a homoscedastic or a heteroscedastic diagonal model.

### 4.3. Deep Ensembles for Full Predictive Uncertainty

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-1 Acc</th>
<th>NLL</th>
<th>ECE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hom. Single Model</td>
<td>76.1</td>
<td>0.943</td>
<td>0.0392</td>
</tr>
<tr>
<td>Hom. Ensemble 4×</td>
<td>77.5</td>
<td>0.877</td>
<td>0.0305</td>
</tr>
<tr>
<td>Het. Single Model</td>
<td>77.5</td>
<td>0.898</td>
<td>0.033</td>
</tr>
<tr>
<td>Het. Ensemble 4×</td>
<td>79.5</td>
<td>0.79</td>
<td>0.015</td>
</tr>
</tbody>
</table>

Table 7: Deep Ensemble results ResNet-50 on Imagenet ILSVRC12. For heteroscedastic models, \( \tau^* = 1.5 \).

Our method estimates heteroscedastic aleatoric uncertainty i.e., input-dependent fundamental label noise in the data. Most approaches in the Bayesian neural networks literature focus on estimating epistemic uncertainty over the network’s parameters [13, 14, 4, 34, 46, 44, 28]. Our method can be easily combined with many of these methods, giving an estimate of full predictive uncertainty. We successfully combine our method with Deep Ensembles [28], a method inspired by Bayesian approaches, that compares favorably in uncertainty modelling benchmarks [38, 17]. To form a deep heteroscedastic ensemble we train each ensemble member with our heteroscedastic layer and average the predictions of the ensemble members, as in a standard (homoscedastic) Deep Ensemble.

A ResNet-50 [19] trained on Imagenet ILSVRC12 is a standard uncertainty quantification benchmark [12, 43]. We train our method and a standard homoscedastic method using 4 different random seeds. We then create an ensemble of each method. We use rank 15 covariance matrices and softmax temperature of 1.5 for the heteroscedastic method and train for 180 epochs. Homoscedastic models are trained for 90 epochs as this maximizes validation set log-likelihood. Table 7 shows the results. We also report expected calibration error [15], a standard metric for uncertainty benchmarks which measures how calibrated a network’s predictions are independent of its performance.

Going from a single homoscedastic model to an ensemble of homoscedastic models provides a substantial improvement in all metrics as does going from a single homoscedastic to our heteroscedastic model. However the best top-1 accuracy, negative log-likelihood and ECE are achieved by a Deep Ensemble of heteroscedastic models. The improvement in top-1 accuracy for the heteroscedastic ensemble (79.5%) compared to the homoscedastic model (76.1%) is greater than the combined gains from ensembling homoscedastic models and the heteroscedastic single model, perhaps due to additional diversity of the ensemble members. Code to reproduce these results and a leaderboard of methods is available publicly.

### 5. Conclusion

We have introduced a new probabilistic method for deep classification under input-dependent label noise. Our method models inter-class correlations in the label noise. We show that the learned correlations correspond to known sources of label noise such as two classes being visually similar or co-occurring. The proposed method scales to very large-scale datasets and we see significant gains on Imagenet ILSVRC12, Imagenet-21k and JFT. We set a new state-of-the-art top-1 accuracy on WebVision. The representations learned by our model on JFT transfer better when fine-tuned on 19 datasets from the VTAB. We combine our method with Deep Ensembles, giving a method for full predictive uncertainty estimation and see substantially improved accuracy, log-likelihood and expected calibration error on ILSVRC12.

---

1https://github.com/google/uncertainty-baselines/tree/master/baselines/imagenet
References


[8] François Chollet et al. Keras. https://keras.io, 2015.


[47] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Jostip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark, 2020. 2, 7, 8