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 1k classes to more than 21k 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-occuring 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.

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 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.

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 (ILSVRC12 & WebVision) to over 21k classes (Imagenet-21k). 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 $\mathrm{softmax}$.

Generative process. Suppose there is some latent vector of utility $\mathbf{u}(\mathbf{x}) \in \mathbf{R}^K$, where K is the number of classes associated with each input $\mathbf{x}$. This utility is the sum of a deterministic reference utility $\boldsymbol{\mu}(\mathbf{x})$ and an unobserved stochastic component $\boldsymbol{\epsilon}$. 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 $\iff y = \arg\max_{j\in[K]} \mathbf{u}_j(\mathbf{x})$:

$$\begin{aligned}
\mathbf{u}(\mathbf{x}) &= \boldsymbol{\mu}(\mathbf{x}) + \boldsymbol{\epsilon} \\
p_c &= P(y=c|\mathbf{x}) = P(\arg\max_{j\in[K]} \mathbf{u}_j(\mathbf{x}) = c) \\
&= \int \mathbb{1}\Big\{\arg\max_{j\in[K]} \mathbf{u}_j(\mathbf{x}) = c\Big\} p(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon}
\end{aligned} \tag{1}$$

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 if we choose each stochastic component to be distributed standard Gumbel independently, $\epsilon_j \sim$ 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]:

$$\begin{aligned}
p_c &= P(\arg\max_{j\in[K]} \mathbf{u}_j(\mathbf{x}) = c) \\
&= \frac{\exp(\mu_c)}{\sum_{j=1}^{K} \exp(\mu_j)} \\
&\iff \epsilon_j \sim \text{ i.i.d. } G(0,1)\ \ \forall j
\end{aligned} \tag{2}$$

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 $\mathbf{x}$ 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 $\boldsymbol{\epsilon}(\mathbf{x})$ is distributed multivariate Normal, $\boldsymbol{\epsilon}(\mathbf{x}) \sim \mathcal{N}(\mathbf{0}, \Sigma(\mathbf{x}))$. 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 $\mathrm{softmax}_\tau$, Eq. (3).

$$\begin{aligned}
p_c &= P(\arg\max_{j\in[K]} \mathbf{u}_j(\mathbf{x}) = c) \\
&= \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\Sigma(\mathbf{x}))}\left[\mathbb{1}\left\{\arg\max_{j\in[K]} \mathbf{u}_j(\mathbf{x}) = c\right\}\right] \\
&= \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\Sigma(\mathbf{x}))}\left[(\lim_{\tau\to 0} \underset{\tau}{\mathrm{softmax}}\, \mathbf{u}(\mathbf{x}))_c\right] \\
&\approx \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\Sigma(\mathbf{x}))}\left[(\underset{\tau}{\mathrm{softmax}}\, \mathbf{u}(\mathbf{x}))_c\right],\ \tau > 0 \\
&\approx \frac{1}{S}\sum_{i=1}^{S} (\underset{\tau}{\mathrm{softmax}}\, \mathbf{u}^{(i)}(\mathbf{x}))_c,\ \ \mathbf{u}^{(i)}(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \Sigma(\mathbf{x})).
\end{aligned} \tag{3}$$

The notation $(\mathbf{u}(\mathbf{x}))_c$ denotes the c^{th} entry of $\mathbf{u}(\mathbf{x})$ and S is the number of MC samples.

In the zero temperature limit this approximation is exact, but for non-zero temperatures τ controls a bias-variance trade-off. At lower temperatures the approximation is closer to the assumed generative process but the gradient variance is higher and vice versa. In practice τ is a hyperparameter that must be selected on a validation set. This approximation is similar to the Gumbel-softmax/Concrete distribution [22, 31]. A similar derivation can be given in the multilabel classification case in which a temperature parameterized sigmoid is used as a smooth approximation to the hard $\mathbb{1}$ function for each class, see Appendix A. We analyze the effect of modelling inter-class correlations by taking a Taylor series approximation to Eq. 3 in Appendix C.

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

In practice we typically compute $V(\mathbf{x})$ as an affine transformation of a shared representation of $\mathbf{x}$ computed by a deep neural network. Suppose that the dimension of this representation is D, then the number of parameters required to compute $V(\mathbf{x})$ is $\mathcal{O}(DKR)$. 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 $V(\mathbf{x})$ 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 $V(\mathbf{x}) = v(\mathbf{x})\mathbf{1}_R^\intercal \odot V$ where $v(\mathbf{x})$ is a vector of dimension R, $\mathbf{1}_R$ 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 $\mathbf{x}$. Sampling the correlated noise component can be simplified, $V(\mathbf{x})\boldsymbol{\epsilon}_R = (v(\mathbf{x})\mathbf{1}_K^\intercal \odot V)\boldsymbol{\epsilon}_R = v(\mathbf{x}) \odot (V\boldsymbol{\epsilon}_R)$. The total parameter count of this parameter-efficient version is $\mathcal{O}(DK + KR)$ 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 $V(\mathbf{x})$ is $44.8M$, a $50\times$ reduction. See Algorithm 1 for a full specification of our method.

Algorithm 1: Computing p_c

Input: Boolean *is-parameter-efficient* = { true/false};
compute shared representation $\mathbf{r}(\mathbf{x}) := f^\theta(\mathbf{x})$;
compute mean parameter $\boldsymbol{\mu}(\mathbf{x}) := W_\mu \mathbf{r}(\mathbf{x}) + \mathbf{b}_\mu$;
compute diagonal correction $\mathbf{d}(\mathbf{x}) := W_d \mathbf{r}(\mathbf{x}) + \mathbf{b}_d$;

generate S standard normal samples
  $\boldsymbol{\epsilon}_K \sim \mathcal{N}(0_K, I_{K\times K})$, $\boldsymbol{\epsilon}_R \sim \mathcal{N}(0_R, I_{R\times R})$;
if *is-parameter-efficient* **then**
  compute heteroscedastic low-rank component
    $v(\mathbf{x}) := W_v \mathbf{r}(\mathbf{x}) + b_v$;
  load homoscedastic low-rank component V;
  $U(\mathbf{x}) := \boldsymbol{\mu}(\mathbf{x}) + \mathbf{d}(\mathbf{x}) \odot \boldsymbol{\epsilon}_K + v(\mathbf{x}) \odot V\boldsymbol{\epsilon}_R$;
else
  compute low-rank parameters
    $V(\mathbf{x}) = W_V \mathbf{r}(\mathbf{x}) + b_V$;
  $V(\mathbf{x}) := \mathtt{reshape}(V(\mathbf{x}), [K, R])$;
  $U(\mathbf{x}) := \boldsymbol{\mu}(\mathbf{x}) + \mathbf{d}(\mathbf{x}) \odot \boldsymbol{\epsilon}_K + V(\mathbf{x})\boldsymbol{\epsilon}_R$;
end
$p_c = \mathtt{mean}(\text{softmax}_\tau\, U(\mathbf{x}), axis = 1)[c]$

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: $y \sim \mathcal{N}(\mu(\mathbf{x}), \sigma(\mathbf{x})^2)$. The negative log-likelihood of the model is particularly amenable to interpretation, Eq. (4).

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

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

Method	Top-1 Acc		Top-5 Acc		NLL	
	90 epochs	270 epochs	90 epochs	270 epochs	90 epochs	270 epochs
Homoscedastic	$76.5^\dagger$ ($\pm$0.22)	$76.7^\dagger$ ($\pm$0.13)	$93.0^\dagger$ ($\pm$0.16)	$92.9^\dagger$ ($\pm$0.12)	$0.98^\dagger$ ($\pm$0.010)	$1.08^\dagger$ ($\pm$0.007)
Het. Diag [9]	$78.3^\dagger$ ($\pm$0.06)	$78.7^\dagger$ ($\pm$0.09)	$94.0^\dagger$ ($\pm$0.06)	$94.0^\dagger$ ($\pm$0.08)	$0.88^\dagger$ ($\pm$0.003)	$0.95^\dagger$ ($\pm$0.008)
Het. Full (ours)	**78.5** ($\pm$0.06)	**79.3** ($\pm$0.10)	**94.3** ($\pm$0.03)	**94.5** ($\pm$0.11)	**0.86** ($\pm$0.002)	**0.92** ($\pm$0.006)

Table 1: Results of ResNet-152 trained on ILSVRC12. For heteroscedastic models $\tau^* = 0.9$. Top-1 and top-5 accuracy and negative log-likelihood $\pm$ 1 standard deviation is reported. 5 runs from different random seeds are used. † $p < 0.05$.

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.

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

Table 2: WebVision 1.0 results. For het. models, $\tau^* = 0.9$. Top-1 and top-5 accuracy and negative log-likelihood ± 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.

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:

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

to

```
logits = MCSoftmaxDenseFA(...)(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.

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.

WebVision. WebVision 1.0 [30] is a popular benchmark for noisy label techniques with the same 1,000 classes as Imagenet ILSVRC12. Labels are gathered through a noisy automated process based on co-occuring text with the images which are scraped from the Web. The training set consists of 2.4M examples and we report results on the validation set, as is standard, which has 50,000 examples.

Following other approaches in the literature [23, 16, 24, 6] we use a InceptionResNet-v2 architecture [40] with the same softmax temperature and covariance matrix rank as the above ILSVRC12 experiments. Other hyperparameters are taken from Jiang et al. [24], however we adopt a longer training schedule of 95 epochs. Following previous methods we report top-1 and top-5 accuracy on both the WebVision validation set and on the ILSVRC12 validation set. For the methods we implement we also report negative log-likelihood.

Our method achieves a new state-of-the-art on WebVision 1.0 with top-1 accuracy of 76.6% and top-5 accuracy of 92.1% on the WebVision validation set. See Table 2 for

Method	Imagenet-21k gAP	Imagenet-21k NLL	JFT gAP	JFT NLL
Homoscedastic	45.9† ($\pm$0.14)	3.65† ($\pm$0.010)	63.1† ($\pm$0.22)	20.12† ($\pm$0.090)
Heteroscedastic Diag $\tau = 1.0$ [9]	45.9† ($\pm$0.06)	3.64† ($\pm$0.003)	63.1† ($\pm$0.03)	19.95† ($\pm$0.022)
Heteroscedastic Diag $\tau^* = 0.15$ [9]	46.8† ($\pm$0.04)	3.63 ($\pm$0.002)	64.1† ($\pm$0.07)	19.61† ($\pm$0.030)
Heteroscedastic PE $\tau^* = 0.15$ (ours)	**47.0** ($\pm$0.08)	**3.62** ($\pm$0.005)	**64.7** ($\pm$0.06)	**19.34** ($\pm$0.026)

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 prevision (gAP) and negative log-likelihood $\pm$ 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 martix provides further impovements in gAP for both datasets. On Imagenet-21k 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.

Class A	Class B	Avg. Cov.
partridge	ruffed grouse, partridge, Bonasa umbellus	-0.46
projectile, missile	missile	-0.44
maillot, tank suit	maillot	-0.42
screen, CRT screen	monitor	-0.37
tape player	cassette player	-0.34
Welsh springer spaniel	Blenheim spaniel	0.28
standard schnauzer	miniature schnauzer	0.27
French bulldog	Boston bull, Boston terrier	0.27
EntleBucher	standard schnauzer	0.26
tennis ball	racket, racquet	-0.26

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 $\Sigma(\mathbf{x}) = diag(\mathbf{d}(\mathbf{x})^2) + V(\mathbf{x})V(\mathbf{x})^\intercal$.

In Table 5, we list 8 cases when the homoscedastic prediction has the highest absolute covariance with the correct

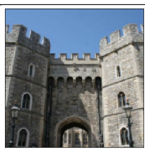
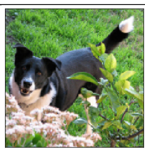
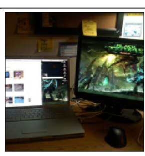
Image	Predictions & covariance
	Het pred: cowboy hat Hom pred: ten-gallon hat, cowboy boot Covariance: -4.26
	Het pred: castle Hom pred: palace Covariance: -1.17
	Het pred: African elephant, Loxodonta africana Hom pred: Indian elephant, Elephas maximus Covariance: 0.78
	Het pred: tractor Hom pred: harvester, reaper Covariance: 1.37
	Het pred: Border collie Hom pred: Cardigan, Cardigan Welsh corgi Covariance: 1.55
	Het pred: computer keyboard, keypad Hom pred: space bar Covariance: 1.82
	Het pred: analog clock Hom pred: wall clock Covariance: 2.46
	Het pred: monitor Hom pred: desktop computer Covariance: 3.07

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.

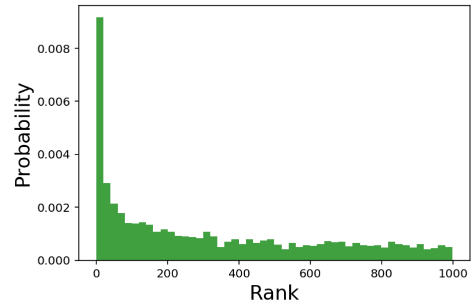


Figure 2: Histogram of the sorted rank of the absolute covariance between the ground truth class and the homoscedastic model's predicted class for ILSVRC12 validation set examples which the heteroscedastic model predicted correctly and the homoscedastic model did not.

class and the heteroscedastic prediction is correct but the homoscedastic prediction is incorrect. We see that these cases broadly fall in two categories which can easily contribute to noisy labels; 1) **substitutes**, e.g., a castle and a palace are two easily confused classes and 2) **co-occurrence**, e.g., a computer keyboard and a space bar are likely to occur in the same image but which one is considered the most prominent class by the annotator may be unclear.

In Table 4 we look at the class pairs with the highest absolute covariance *averaged* over the ILSVRC12 validation set. We compute the average covariance matrix over the 50,000 validation set images and extract the top-10 entries by absolute value. The class pairs, all are substitutes for each other or commonly co-occur. There are just under 1M possible class pairs, it is remarkable that the average covariance matrix exhibits such clear and consistent structure.

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.

	VTAB$_{1K}$ score
Homoscedastic	$70.46^{\dagger} \pm 0.5$
Heteroscedastic Diag	71.12 ± 0.19
Heteroscedastic PE	**71.34** $\pm$ 0.23

Table 6: VTAB$_{1K}$ score $\pm$ 1 standard deviation for ResNet50 models pre-trained on JFT and fine-tuned on 19 diverse image classification datasets. † p-value < 0.05.

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 VTAB$_{1K}$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 VTAB$_{1K}$ score is an average of the accuracy on all 19 datasets.

Table 6 shows VTAB$_{1K}$ scores. Our parameter-efficient heteroscedastic model, which captures correlations in the JFT label noise, improves the VTAB$_{1K}$ 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

Method	Top-1 Acc	NLL	ECE
Hom. Single Model	76.1	0.943	0.0392
Hom. Ensemble 4×	77.5	0.877	0.0305
Het. Single Model	77.5	0.898	0.033
Het. Ensemble 4×	**79.5**	**0.79**	**0.015**

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 a 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[1].

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.

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

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016. 5

[2] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020. 1, 5, 6, 14

[3] Christopher M Bishop and Cazhaow S Quazaz. Regression with input-dependent noise: A bayesian treatment. In *Advances in neural information processing systems*, pages 347–353, 1997. 3

[4] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on Machine Learning-Volume 37*, pages 1613–1622, 2015. 8

[5] Moshe Ben-AkiWand Denis Bolduc. Multinomial probit with a logit kernel and a general parametric specification of the covariatice structure. Technical report, MIT Working Paper, 1996. 4

[6] Kaidi Cao, Yining Chen, Junwei Lu, Nikos Arechiga, Adrien Gaidon, and Tengyu Ma. Heteroskedastic and imbalanced deep learning with adaptive regularization. *arXiv preprint arXiv:2006.15766*, 2020. 1, 4, 5

[7] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 6

[8] François Chollet et al. Keras. `https://keras.io`, 2015. 5

[9] Mark Collier, Basil Mustafa, Efi Kokiopoulou, and Jesse Berent. A simple probabilistic method for deep classification underinput-dependent label noise. *arXiv preprint arXiv:2003.06778*, 2020. 1, 2, 4, 5, 6

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1, 6

[11] Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, Sylvain Gelly, Neil Houlsby, Xiaohua Zhai, and Mario Lucic. On robustness and transferability of convolutional neural networks. *arXiv:2007.08558*, 2020. 7

[12] Michael W Dusenberry, Ghassen Jerfel, Yeming Wen, Yian Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. Efficient and scalable bayesian neural nets with rank-1 factors. *arXiv preprint arXiv:2005.07186*, 2020. 8

[13] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059, 2016. 8

[14] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. In *Advances in Neural Information Processing Systems*, pages 3581–3590, 2017. 8

[15] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330, 2017. 8

[16] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150, 2018. 1, 5

[17] Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 318–319, 2020. 8

[18] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. Co-teaching: Robust Training of Deep Neural Networks with Extremely Noisy Labels. *Advances in Neural Information Processing Systems (NIPS)*, 2018. 1, 4

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5, 6, 8, 11

[20] Daniel Hernández-Lobato, Viktoriia Sharmanska, Kristian Kersting, Christoph H Lampert, and Novi Quadrianto. Mind the nuisance: Gaussian process classification using privileged noise. In *Advances in Neural Information Processing Systems*, pages 837–845, 2014. 2, 4

[21] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL `http://arxiv.org/abs/1503.02531`. 1, 6

[22] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *Proceedings of the 5th International Conference on Learning Representations*, 2017. URL `https://arxiv.org/abs/1611.01144`. 3

[23] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. *International Conference on Machine Learning (ICML)*, 2018. 1, 4, 5

[24] Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. Beyond synthetic noise: Deep learning on controlled noisy labels. In *ICML*, 2020. 1, 4, 5

[25] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, pages 5574–5584, 2017. 2, 3, 4, 6

[26] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. *arXiv preprint arXiv:1912.11370*, 2019. 6, 7, 8

[27] David Krueger, Nicolas Ballas, Stanislaw Jastrzebski, Devansh Arpit, Maxinder S Kanwal, Tegan Maharaj, Emmanuel Bengio, Asja Fischer, Aaron Courville, Simon Lacoste-Julien, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning (ICML)*, 2017. 5

[28] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017. 3, 4, 8

[29] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5447–5456, 2018. 1, 5

[30] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017. 1, 5

[31] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the 5th International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1611.00712. 3

[32] Daniel McFadden. A method of simulated moments for estimation of discrete response models without numerical integration. *Econometrica: Journal of the Econometric Society*, pages 995–1026, 1989. 4

[33] Miguel Monteiro, Loïc Le Folgoc, Daniel Coelho de Castro, Nick Pawlowski, Bernardo Marques, Konstantinos Kamnitsas, Mark van der Wilk, and Ben Glocker. Stochastic segmentation networks: Modelling spatially correlated aleatoric uncertainty. *arXiv preprint arXiv:2006.06015*, 2020. 4

[34] Radford M Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995. 8

[35] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. In *Advances in Neural Information Processing Systems*, pages 3347–3357, 2019. 7

[36] S Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014. 1, 4

[37] Shreyas Saxena, Oncel Tuzel, and Dennis DeCoste. Data parameters: A new family of parameters for learning a differentiable curriculum. In *Advances in Neural Information Processing Systems*, pages 11095–11105, 2019. 1, 5

[38] Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D Sculley, Joshua Dillon, Jie Ren, and Zachary Nado. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pages 13969–13980, 2019. 8

[39] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 6, 7

[40] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4278–4284. AAAI Press, 2017. 5, 11

[41] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019. 11

[42] Kenneth E Train. *Discrete choice methods with simulation*. Cambridge university press, 2009. 2, 3, 4

[43] Yeming Wen, Ghassen Jerfel, Rafael Muller, Michael W Dusenberry, Jasper Snoek, Balaji Lakshminarayanan, and Dustin Tran. Combining ensembles and data augmentation can harm your calibration. *arXiv preprint arXiv:2010.09875*, 2020. 8

[44] Florian Wenzel, Kevin Roth, Bastiaan S Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? *arXiv preprint arXiv:2002.02405*, 2020. 8

[45] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006. 2, 3, 4

[46] Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*, 2020. 8

[47] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip 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

[48] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 4