A. Related Work

Several existing approaches model uncertainty using feature-space density but underperform without fine-tuning on OoD data. Our work identifies feature collapse and objective mismatch as possible reasons for this. Among these approaches, [43] uses Mahalanobis distances to quantify uncertainty by fitting a class-wise Gaussian distribution (with shared covariance matrices) on the feature space of a pre-trained ResNet encoder. The competitive results they report require input perturbations, ensembling GMM densities from multiple layers, and fine-tuning on OoD hold-out data. They do not discuss any constraints which the ResNet encoder should satisfy, and therefore, are vulnerable to feature collapse. In Fig. 1c, for example, the feature density of a LeNet and a VGG are unable to distinguish OoD from iD samples. [59] also propose a density-based estimation of aleatoric and epistemic uncertainty. Similar to [43], they do not constrain their pre-trained ResNet encoder. They do discuss feature collapse though, noting that they do not address this problem. Moreover, they do not consider the objective mismatch that arises (see Proposition 5.3 below) and use a single estimator for both epistemic and aleatoric uncertainty. Consequently, they report worse epistemic uncertainty: 74% AUROC on CIFAR-10 vs SVHN, which we show to considerably fall behind modern approaches for uncertainty estimation in deep learning in §4. Likewise, [46] compute an unnormalized density based on the softmax logits without taking into account the need for inductive biases to ensure smoothness and sensitivity of the feature space.

[69] use contrastive training on the feature extractor before estimating the feature-space density. Our method is orthogonal from this work as we restrict ourselves to the supervised setting and show that the inductive biases that result in bi-Lipschitzness [45; 65] are sufficient for the feature-space density to reliably capture epistemic uncertainty.

Lastly, our method improves upon [65] and [45] by alleviating the need for additional hyperparameters: DDU only needs minimal changes from the standard softmax setup to outperform DUQ and SNGP on uncertainty benchmarks, and our GMM parameters are optimised for the already trained model using the training set. In particular, DDU does not require training or fine-tuning with OoD data. Moreover, our insights in §5 explain why [45] found that a baseline that uses *the softmax entropy instead of the feature-space density* of a deterministic network with bi-Lipschitz constraint underperforms.

A.1. Predictive Entropy and Confidence in Recent Works

Table 4 shows a selection of recently published papers which use entropy or confidence as OoD score. Only two papers examine using Mutual Information with Deep Ensembles as OoD score at all. None of the papers examines the possible confounding of aleatoric and epistemic uncertainty when using predictive entropy or confidence, or the consistency issues of softmax entropy (and softmax confidence), detailed in §5. This list is not exhaustive, of course.

Figure 8. *Samples from Ambiguous-MNIST.*

B. Ambiguous- and Dirty-MNIST

Each sample in Ambiguous-MNIST is constructed by decoding a linear combination of latent representations of 2 different MNIST digits from a pre-trained VAE [35]. Every decoded image is assigned several labels sampled from the softmax probabilities of an off-the-shelf MNIST neural network ensemble, with points filtered based on an ensemble's MI (to remove 'junk' images) and then stratified class-wise based on their softmax entropy (some classes are inherently more ambiguous, so we "amplify" these; we stratify per-class to try to preserve a wide spread of possible entropy values, and avoid introducing additional ambiguity which will increase all points to have highest entropy). All off-the-shelf MNIST neural networks were then discarded and new models were trained to generate Fig 1 (and as can be seen, the ambiguous points we generate indeed have high entropy regardless of the model architecture used). We create 60K such training and 10K test images to construct Ambiguous-MNIST. Finally, the Dirty-MNIST dataset in this experiment contains MNIST and Ambiguous-MNIST samples in a 1:1 ratio (with 120K training and 20K test samples). In Fig. 8, we provide some samples from Ambiguous-MNIST.

C. Algorithm

C.1. Increasing sensitivity

Using residual connections to enforce sensitivity works well in practice when the layer is defined as $x' = x + f(x)$. However, there are several places in the network where additional spatial downsampling is done in $f(\cdot)$ (through a strided convolution), and in order to compute the residual operation x needs to be downsampled as well. These downsampling

Table 4. *A sample of recently published papers and OoD metrics.* Many recently published papers only use Predictive Entropy or Predictive Confidence (for Deep Ensembles) or Softmax Confidence (for deterministic models) as OoD scores without addressing the possible confounding of aleatoric and epistemic uncertainty, that is ambiguous iD samples with OoD samples. Only two papers examine using Mutual Information with Deep Ensembles as OoD score at all.

Title	Citation	Softmax Confidence	Predictive Confidence	Predictive Entropy	Mutual Information
A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks	[24]	✓	✗	✗	✗
Deep Anomaly Detection with Outlier Exposure	[25]	✓	✗	✗	✗
Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks	[44]	✓	✗	✗	✗
Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples	[42]	✓	✗	✗	✗
Learning Confidence for Out-of-Distribution Detection in Neural Networks	[8]	✓	✗	✗	✗
Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles	[40]	✗	✓	✓	✗
Predictive Uncertainty Estimation via Prior Networks	[50]	✗	✓	✓	✓
Ensemble Distribution Distillation	[51]	✗	✗	✓	✓
Generalized ODIN: Detecting Out-of-Distribution Image Without Learning From Out-of-Distribution Data	[29]	✓	✗	✗	✗
Being Bayesian, Even Just a Bit, Fixes Overconfidence in ReLU Networks	[38]	✗	✓	✗	✗

operations are crucial for managing memory consumption and generalisation. The way this is traditionally done in ResNets is by introducing an additional function $g(\cdot)$ on the residual branch (obtaining $x' = g(x) + f(x)$) which is a strided 1x1 convolution. In practice, the stride is set to 2 pixels, which leads to the output of $g(\cdot)$ only being dependent on the top-left pixel of each 2x2 patch, which reduces sensitivity. We overcome this issue by making an architectural change that improves uncertainty quality without sacrificing accuracy. We use a strided average pooling operation instead of a 1x1 convolution in $g(\cdot)$. This makes the output of $g(\cdot)$ dependent on all input pixels. Additionally, we use leaky ReLU activation functions, which are equivalent to ReLU activations when the input is larger than 0, but below 0 they compute $p * x$ with $p = 0.01$ in practice. These further improve sensitivity as all negative activations still propagate in the network.

C.2. Algorithm Implementation

The algorithm is provided in Algorithm 1. Note that in order to compute thresholds for low and high density or entropy, we can simply use the training set containing iD data. We set all points having density lower than 99% quantile as OoD.

C.3. Computational Complexity

Let N be the number of samples; D, the feature space dimensionality; and C, the number of classes with $\approx N/C$ samples per class (balanced). For fitting the GMM via GDA: computing the covariance matrix per class requires $(C(N/C)D^2) = (ND^2)$ complexity. Computing the inverse and determinant of the covariance matrices via the Cholesky decomposition requires (D^3) per class. Thus, the total computational cost for GDA is $(ND^2 + CD^3)$. Evaluating density of a single point: distance from class means requires (CD), and matrix vector multiplications requires (CD^2). Hence, the total cost for evaluating density on a single point is (CD^2).

D. Experimental Details

D.1. Dirty-MNIST

We train for 50 epochs using SGD with a momentum of 0.9 and an initial learning rate of 0.1. The learning rate drops by a factor of 10 at training epochs 25 and 40. Following SNGP [45], we apply online spectral normalisation with one step of a power iteration on the convolutional weights. For 1x1 convolutions, we use the exact algorithm, and for 3x3 convolutions, the approximate algorithm from [18]. The coefficient for SN is a hyper-parameter which we set to 3 using cross-validation.

D.2. OoD Detection Training Setup

We train the softmax baselines on CIFAR-10/100 for 350 epochs using SGD as the optimiser with a momentum of 0.9, and an initial learning rate of 0.1. The learning rate drops by a factor of 10 at epochs 150 and 250. We train the 5-Ensemble baseline using this same training setup. The SNGP and DUQ models were trained using the setup of SNGP and hyper-parameters mentioned in their respective papers [45; 65]. For models trained on ImageNet, we train for 90 epochs with SGD optimizer, an initial learning rate of 0.1 and a weight decay of 1e-4. We use a learning rate warmup decay of 0.01 along with a step scheduler with step size of 30 and a step factor of 0.1.

Algorithm 1 Deep Deterministic Uncertainty

Definitions:
- Regularized feature extractor $f_\theta : x \to \mathbb{R}^d$
- Softmax output predictions: $p(y|x)$
- GMM density: $q(z) = \sum_y q(z|y=c)\, q(y=c)$
- Dataset (X, Y)

procedure TRAIN
  train NN $p(y|f_\theta(x))$ with (X, Y)
  for each class c with samples $\mathbf{x}_c \subset X$ **do**
    $\mu_c \leftarrow \frac{1}{|\mathbf{x}_c|} \sum_{\mathbf{x}_c} f_\theta(\mathbf{x}_c)$
    $\Sigma_c \leftarrow \frac{1}{|\mathbf{x}_c|-1}(f_\theta(\mathbf{x}_c) - \mu_c)(f_\theta(\mathbf{x}_c) - \mu_c)^T$
    $\pi_c \leftarrow \frac{\sum_{\mathbf{x}_c} 1}{|X|}$
  end for
end procedure

function DISENTANGLE_UNCERTAINTY(sample x)
  compute feature representation $z = f_\theta(x)$
  compute density under GMM: $q(z) = \sum_y \mathrm{q}(z \mid y)\, \mathrm{q}(y)$ with $\mathrm{q}(z \mid y) \sim \mathcal{N}(\mu_y; \sigma_y), \mathrm{q}(y) = \pi_y$
  compute softmax entropy: $H_p[Y|x]$

  if low density $q(z)$ **then**
    return $(-q(z),)$
  else if high density $q(z)$ **then**
    if high entropy $H_p[Y|x]$ **then**
      return ambiguous iD
    else if low entropy $H_p[Y|x]$ **then**
      return $(0, H_p[Y|x])$
    end if
  end if
end function

D.3. Semantic Segmentation Training Setup

In Figure 9, we plot the L2 distance between feature space means of different classes for a pair of randomly chosen distant pixels on the Pascal VOC 2012 val set. We observe that feature space means between pairs of different classes are more distant compared to the same class irrespective of the location of the pixel for the class. This leads us to construct a Gaussian mean and covariance matrix per class as opposed to one mean and one covariance matrix per class per pixel, thereby greatly reducing the computational load of fitting a GMM in semantic segmentation. Similar to classification, we treat each pixel in the training set as a separate sample and fit a single Gaussian mean and covariance matrix per class.

For the semantic segmentation experiment, we use a DeepLab-v3+ [3] model with a ResNet-101 backbone as the architecture of choice. We train each of the models on Pascal VOC for 50 epochs using SGD as the optimizer, with a momentum of 0.9 and a weight decay of $5e-4$. We set the initial learning rate to 0.007 with a polynomial decay during the course of training. Finally, we trained with a batch size of 32 parallelized over 4 GPUs.

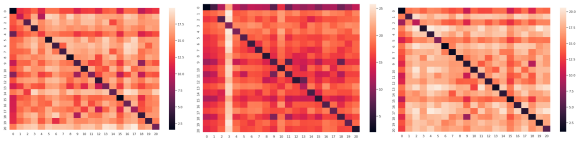


Figure 9. L2 distances between the feature space means of different classes for a pair of distant pixels on the Pascal VOC 2012 val set: (left) Pixels $(10, 255)$ and $(500, 255)$, (middle) Pixels $(234, 349)$ and $(36, 22)$ and (right) Pixels $(300, 500)$ and $(400, 255)$.

D.4. Compute Resources

Each model (ResNet-18, Wide-ResNet-28-10, ResNet-50, ResNet-110, DenseNet-121 or VGG-16) used for the large scale active learning, CIFAR-10 vs SVHN/CIFAR-100/Tiny-ImageNet/CIFAR-10-C and CIFAR-100 vs SVHN/Tiny-ImageNet tasks was trained on a single Nvidia Quadro RTX 6000 GPU. Each model (LeNet, VGG-16 and ResNet-18) used to get the results in Fig. 1 and Tab. 11 was trained on a single Nvidia GeForce RTX 2060 GPU. Each model (ResNet-50, Wide-ResNet-50-2, VGG-16) trained on ImageNet was trained using 8 Nvidia Quadro RTX 6000 GPUs.

E. Additional Results

In this section, we provide details of additional results on the OoD detection task using CIFAR-10 vs SVHN/CIFAR-100/Tiny-ImageNet/CIFAR-10-C and CIFAR-100 vs SVHN/Tiny-ImageNet for ResNet-50, ResNet-110 and DenseNet-121 architectures. We present results on ResNet-50, ResNet-110 and DenseNet-121 for CIFAR-10 vs SVHN/CIFAR-100/Tiny-ImageNet and CIFAR-100 vs SVHN/Tiny-ImageNet in Tab. 5, Tab. 6 and Tab. 7 respectively. We also present results on individual corruption types for CIFAR-10-C for Wide-ResNet-28-10, ResNet-50, ResNet-110 and DenseNet-121 in Fig. 10, Fig. 11, Fig. 12 and Fig. 13 respectively.

Finally, we provide results for various ablations on DDU. As mentioned in §3, DDU consists of a deterministic softmax model trained with appropriate inductive biases. It uses softmax entropy to quantify aleatoric uncertainty and feature-space density to quantify epistemic uncertainty. In the ablation, we try to experimentally evaluate the following scenarios:

1. **Effect of inductive biases (sensitivity + smoothness):** We want to see the effect of removing the proposed inductive biases (i.e. no sensitivity and smoothness constraints) on the OoD detection performance of a model. To do this, we train a VGG-16 with and without spectral nor-

Table 5. *OoD detection performance of different baselines using a ResNet-50 architecture with the CIFAR-10 vs SVHN/CIFAR-100/Tiny-ImageNet and CIFAR-100 vs SVHN/Tiny-ImageNet dataset pairs averaged over 25 runs.* Note: SN stands for Spectral Normalisation, JP stands for Jacobian Penalty. We highlight the best deterministic and best method overall in bold for each metric.

Train Dataset	Method	Penalty	Aleatoric Uncertainty	Epistemic Uncertainty	Test Accuracy (↑)	Test ECE (↓)	AUROC SVHN (↑)	AUROC CIFAR-100 (↑)	AUROC Tiny-ImageNet (↑)
CIFAR-10	Softmax	-	Softmax Entropy	Softmax Entropy	**95.04 ± 0.05**	**0.97 ± 0.04**	93.80 ± 0.41	88.91 ± 0.07	88.32 ± 0.07
	Energy-based [46]	-		Softmax Density			94.48 ± 0.44	88.84 ± 0.08	88.45 ± 0.08
	DUQ [65]	JP	Kernel Distance	Kernel Distance	94.05 ± 0.11	1.71 ± 0.07	93.14 ± 0.43	83.87 ± 0.27	84.28 ± 0.26
	SNGP [45]	SN	Predictive Entropy	Predictive Entropy	94.90 ± 0.11	1.01 ± 0.03	93.15 ± 0.85	89.32 ± 0.10	88.96 ± 0.13
	DDU (ours)	SN	Softmax Entropy	GMM Density	94.92 ± 0.06	1 ± 0.04	**94.77 ± 0.35**	**89.98 ± 0.17**	**89.12 ± 0.13**
	5-Ensemble [40]	-	Predictive Entropy	Predictive Entropy	**96.06 ± 0.04**	1.65 ± 0.07	94.75 ± 0.39	89.87 ± 0.06	88.69 ± 0.05
				Mutual Information			94.09 ± 0.20	89.76 ± 0.06	89.04 ± 0.03
					Test Accuracy (↑)	**Test ECE (↓)**	**AUROC SVHN (↑)**		**AUROC Tiny-ImageNet (↑)**
CIFAR-100	Softmax	-	Softmax Entropy	Softmax Entropy	77.91 ± 0.09	4.32 ± 0.10	81.32 ± 0.65		79.83 ± 0.07
	Energy-based [46]	-		Softmax Density			82.05 ± 0.69		79.61 ± 0.08
	SNGP [45]	SN	Predictive Entropy	Predictive Entropy	74.73 ± 0.22	7.68 ± 0.13	82.50 ± 2.09		77.05 ± 0.16
	DDU (ours)	SN	Softmax Entropy	GMM Density	**79.26 ± 0.16**	**4.07 ± 0.06**	**87.34 ± 0.64**		**82.11 ± 0.20**
	5-Ensemble [40]	-	Predictive Entropy	Predictive Entropy	**81.06 ± 0.07**	**3.54 ± 0.12**	83.42 ± 0.89		77.69 ± 0.12
				Mutual Information			84.24 ± 0.90		81.59 ± 0.05

Table 6. *OoD detection performance of different baselines using a ResNet-110 architecture with the CIFAR-10 vs SVHN/CIFAR-100/Tiny-ImageNet and CIFAR-100 vs SVHN/Tiny-ImageNet dataset pairs averaged over 25 runs.* Note: SN stands for Spectral Normalisation, JP stands for Jacobian Penalty. We highlight the best deterministic and best method overall in bold for each metric.

Train Dataset	Method	Penalty	Aleatoric Uncertainty	Epistemic Uncertainty	Test Accuracy (↑)	Test ECE (↓)	AUROC SVHN (↑)	AUROC CIFAR-100 (↑)	AUROC Tiny-ImageNet (↑)
CIFAR-10	Softmax	-	Softmax Entropy	Softmax Entropy	**95.08 ± 0.04**	1.02 ± 0.04	93.12 ± 0.44	88.7 ± 0.1	88.07 ± 0.11
	Energy-based [46]	-		Softmax Density			93.67 ± 0.47	88.60 ± 0.11	88.13 ± 0.11
	DUQ [65]	JP	Kernel Distance	Kernel Distance	94.32 ± 0.17	1.21 ± 0.07	94.02 ± 0.45	86.17 ± 0.35	85.24 ± 0.21
	SNGP [45]	SN	Predictive Entropy	Predictive Entropy	94.85 ± 0.09	1.04 ± 0.02	93.17 ± 0.53	89.23 ± 0.10	88.80 ± 0.12
	DDU (ours)	SN	Softmax Entropy	GMM Density	94.82 ± 0.06	**1.01 ± 0.04**	**95.48 ± 0.30**	**90.08 ± 0.13**	**89.18 ± 0.15**
	5-Ensemble [40]	-	Predictive Entropy	Predictive Entropy	**96.18 ± 0.05**	1.57 ± 0.05	95.07 ± 0.45	**90.23 ± 0.04**	89 ± 0.03
				Mutual Information			94.72 ± 0.34	89.69 ± 0.05	88.35 ± 0.05
					Test Accuracy (↑)	**Test ECE (↓)**	**AUROC SVHN (↑)**		**AUROC Tiny-ImageNet (↑)**
CIFAR-100	Softmax	-	Softmax Entropy	Softmax Entropy	78.65 ± 0.10	3.93 ± 0.13	82.04 ± 0.57		80.13 ± 0.07
	Energy-based [46]	-		Softmax Density			82.78 ± 0.60		80.01 ± 0.09
	SNGP [45]	SN	Predictive Entropy	Predictive Entropy	76.16 ± 0.27	6.43 ± 0.75	83.94 ± 0.10		78.54 ± 0.28
	DDU (ours)	SN	Softmax Entropy	GMM Density	**78.89 ± 0.17**	**3.79 ± 0.07**	**88.66 ± 0.56**		**82.58 ± 0.24**
	5-Ensemble [40]	-	Predictive Entropy	Predictive Entropy	**81.80 ± 0.10**	**3.67 ± 0.11**	83.68 ± 0.33		81.12 ± 0.13
				Mutual Information			85.11 ± 0.57		81.94 ± 0.06

Table 7. *OoD detection performance of different baselines using a DenseNet-121 architecture with the CIFAR-10 vs SVHN/CIFAR-100/Tiny-ImageNet and CIFAR-100 vs SVHN/Tiny-ImageNet dataset pairs averaged over 25 runs.* Note: SN stands for Spectral Normalisation, JP stands for Jacobian Penalty. We highlight the best deterministic and best method overall in bold for each metric.

Train Dataset	Method	Penalty	Aleatoric Uncertainty	Epistemic Uncertainty	Test Accuracy (↑)	Test ECE (↓)	AUROC SVHN (↑)	AUROC CIFAR-100 (↑)	AUROC Tiny-ImageNet (↑)
CIFAR-10	Softmax	-	Softmax Entropy	Softmax Entropy	95.16 ± 0.03	1.10 ± 0.04	94 ± 0.44	87.55 ± 0.11	86.99 ± 0.12
	Energy-based [46]	-		Softmax Density			94.07 ± 0.54	86.73 ± 0.15	86.43 ± 0.16
	DUQ [65]	JP	Kernel Distance	Kernel Distance	95.02 ± 0.14	1.08 ± 0.08	94.67 ± 0.41	87.38 ± 0.21	86.72 ± 0.14
	SNGP [45]	SN	Predictive Entropy	Predictive Entropy	94.31 ± 0.21	1.08 ± 0.10	94.48 ± 0.34	88.86 ± 0.46	88.40 ± 0.48
	DDU (ours)	SN	Softmax Entropy	GMM Density	**95.21 ± 0.03**	**1.05 ± 0.03**	**96.21 ± 0.31**	**90.84 ± 0.06**	**89.70 ± 0.06**
	5-Ensemble [40]	-	Predictive Entropy	Predictive Entropy	**96.18 ± 0.05**	1.07 ± 0.07	95.78 ± 0.11	90.65 ± 0.03	89.62 ± 0.06
				Mutual Information			95.75 ± 0.10	90.71 ± 0.04	89.34 ± 0.06
					Test Accuracy (↑)	**Test ECE (↓)**	**AUROC SVHN (↑)**		**AUROC Tiny-ImageNet (↑)**
CIFAR-100	Softmax	-	Softmax Entropy	Softmax Entropy	79.02 ± 0.08	4.11 ± 0.08	85.86 ± 0.42		81.10 ± 0.07
	Energy-based [46]	-		Softmax Density			87.09 ± 0.49		80.84 ± 0.08
	SNGP [45]	SN	Predictive Entropy	Predictive Entropy	79.15 ± 0.15	6.73 ± 0.10	85.00 ± 0.12		79.76 ± 0.15
	DDU (ours)	SN	Softmax Entropy	GMM Density	**79.15 ± 0.07**	**4.11 ± 0.06**	**88.44 ± 0.55**		**81.85 ± 0.11**
	5-Ensemble [40]	-	Predictive Entropy	Predictive Entropy	**81.01 ± 0.13**	4.81 ± 0.05	88.32 ± 0.61		81.45 ± 0.12
				Mutual Information			88.36 ± 0.17		81.73 ± 0.06

malisation. Note that VGG-16 does not have residual connections and hence, a VGG-16 does not follow the sensitivity and smoothness (bi-Lipschitz) constraints.

2. **Effect of sensitivity alone:** Since residual connections make a model sensitive to changes in the input space by lower bounding its Lipschitz constant, we also want to see how a network performs with just the sensitivity constraint alone. To observe this, we train a Wide-ResNet-28-10 without spectral normalisation (i.e. no explicit upper bound on the Lipschitz constant of the model).
3. **Metrics for aleatoric and epistemic uncertainty:** With the above combinations, we try to observe how different metrics for aleatoric and epistemic uncertainty perform. To quantify aleatoric uncertainty, we use the softmax

Table 8. *OoD detection performance of different ablations trained on CIFAR-10 using Wide-ResNet-28-10 and VGG-16 architectures with SVHN, CIFAR-100 and Tiny-ImageNet as OoD datasets averaged over 25 runs.* Note: SN stands for Spectral Normalisation. We highlight the best deterministic and best method overall in bold for each metric.

Ablations					Aleatoric Uncertainty	Epistemic Uncertainty	Test Accuracy (↑)	Test ECE (↓)	AUROC SVHN (↑)	AUROC CIFAR-100 (↑)	AUROC Tiny-ImageNet (↑)
Architecture	**Ensemble**	**Residual Connections**	**SN**	**GMM**							
Wide-ResNet-28-10	✗	✓	✗	✗	Softmax Entropy	Softmax Entropy Softmax Density	$\mathbf{95.98 \pm 0.02}$	0.85 ± 0.02	94.44 ± 0.43 94.56 ± 0.51	89.39 ± 0.06 88.89 ± 0.07	88.42 ± 0.05 88.11 ± 0.06
				✓	Softmax Entropy	GMM Density	95.98 ± 0.02	0.85 ± 0.02	96.08 ± 0.25	90.94 ± 0.03	90.62 ± 0.05
			✓	✗	Softmax Entropy	Softmax Entropy Softmax Density	95.97 ± 0.03	0.85 ± 0.04	94.05 ± 0.26 94.31 ± 0.33	90.02 ± 0.07 89.78 ± 0.08	89.07 ± 0.06 88.96 ± 0.07
				✓	**Softmax Entropy**	**GMM Density**	95.97 ± 0.03	$\mathbf{0.85 \pm 0.04}$	$\mathbf{97.86 \pm 0.19}$	$\mathbf{91.34 \pm 0.04}$	$\mathbf{91.07 \pm 0.05}$
	✓	✓	✗	✗	Predictive Entropy	Predictive Entropy Mutual Information	$\mathbf{96.59 \pm 0.02}$	$\mathbf{0.76 \pm 0.03}$	97.73 ± 0.31 97.18 ± 0.19	$\mathbf{92.13 \pm 0.02}$ 91.33 ± 0.03	90.06 ± 0.03 90.90 ± 0.03
VGG-16	✗	✓	✗	✗	Softmax Entropy	Softmax Entropy Softmax Density	93.63 ± 0.04	1.64 ± 0.03	85.76 ± 0.84 84.24 ± 1.04	82.48 ± 0.14 81.91 ± 0.17	83.07 ± 0.12 82.82 ± 0.14
				✓	Softmax Entropy	GMM Density	93.63 ± 0.04	1.64 ± 0.03	89.25 ± 0.36	86.55 ± 0.10	86.78 ± 0.09
			✓	✗	Softmax Entropy	Softmax Entropy Softmax Density	93.62 ± 0.04	1.78 ± 0.04	87.54 ± 0.41 86.28 ± 0.51	82.71 ± 0.09 82.15 ± 0.11	83.33 ± 0.08 83.07 ± 0.10
				✓	Softmax Entropy	GMM Density	93.62 ± 0.04	1.78 ± 0.04	89.62 ± 0.37	86.37 ± 0.14	86.63 ± 0.11
	✓	✓	✗	✗	Predictive Entropy	Predictive Entropy Mutual Information	94.9 ± 0.05	2.03 ± 0.03	92.80 ± 0.18 91 ± 0.22	89.01 ± 0.08 88.43 ± 0.08	87.66 ± 0.08 88.74 ± 0.05

Table 9. *OoD detection performance of different ablations trained on CIFAR-100 using Wide-ResNet-28-10 and VGG-16 architectures with SVHN and Tiny-ImageNet as the OoD dataset averaged over 25 runs.* Note: SN stands for Spectral Normalisation. We highlight the best deterministic and best method overall in bold for each metric.

Ablations					Aleatoric Uncertainty	Epistemic Uncertainty	Test Accuracy (↑)	Test ECE (↓)	AUROC SVHN (↑)	AUROC Tiny-ImageNet (↑)
Architecture	**Ensemble**	**Residual Connections**	**SN**	**GMM**						
Wide-ResNet-28-10	✗	✓	✗	✗	Softmax Entropy	Softmax Entropy Softmax Density	80.26 ± 0.06	4.62 ± 0.06	77.42 ± 0.57 78.00 ± 0.63	81.53 ± 0.05 81.33 ± 0.06
				✓	Softmax Entropy	GMM Density	80.26 ± 0.06	4.62 ± 0.06	87.54 ± 0.61	78.13 ± 0.08
			✓	✗	Softmax Entropy	Softmax Entropy Softmax Density	80.98 ± 0.06	4.10 ± 0.08	85.37 ± 0.36 86.41 ± 0.38	82.57 ± 0.03 82.49 ± 0.04
				✓	**Softmax Entropy**	**GMM Density**	$\mathbf{80.98 \pm 0.06}$	$\mathbf{4.10 \pm 0.08}$	$\mathbf{87.53 \pm 0.62}$	$\mathbf{83.13 \pm 0.06}$
	✓	✓	✗	✗	Predictive Entropy	Predictive Entropy Mutual Information	$\mathbf{82.79 \pm 0.10}$	$\mathbf{3.32 \pm 0.09}$	79.54 ± 0.91 77.00 ± 1.54	82.95 ± 0.09 82.82 ± 0.04
VGG-16	✗	✓	✗	✗	Softmax Entropy	Softmax Entropy Softmax Density	73.48 ± 0.05	4.46 ± 0.05	76.73 ± 0.72 77.70 ± 0.86	76.43 ± 0.05 74.68 ± 0.07
				✓	Softmax Entropy	GMM Density	73.48 ± 0.05	4.46 ± 0.05	75.65 ± 0.95	74.32 ± 1.73
			✓	✗	Softmax Entropy	Softmax Entropy Softmax Density	73.58 ± 0.06	4.32 ± 0.06	77.21 ± 0.77 77.76 ± 0.90	76.59 ± 0.06 74.86 ± 0.08
				✓	Softmax Entropy	GMM Density	73.58 ± 0.06	4.32 ± 0.06	75.99 ± 1.23	74.06 ± 1.67
	✓	✓	✗	✗	Predictive Entropy	Predictive Entropy Mutual Information	77.84 ± 0.11	5.32 ± 0.10	79.62 ± 0.73 72.07 ± 0.48	78.66 ± 06 76.27 ± 0.05

Table 10. *LDA vs GDA ablation for OoD detection performance using Wide-ResNet-50-2, ResNet-50, Wide-ResNet-50-2 architectures (depending on dataset) on CIFAR-10 vs SVHN/CIFAR-100/TinyImageNet, CIFAR-100 vs SVHN/TinyImageNet, and ImageNet vs ImageNet-O [26].* Best AUROC (↑) scores are marked in bold. GDA performs better, except with SVHN as OoD dataset.

Model	WRN-28-10			WRN-28-10		WRN-50-2	RN-50
iD	CIFAR-10			CIFAR-100		ImageNet	
OoD	SVHN	CIFAR-100	TinyImageNet	SVHN	TinyImageNet	ImageNet-O	
LDA (Maha [43])	$\mathbf{98.41 \pm 0.09}$	82.90 ± 0.23	82.48 ± 0.25	$\mathbf{92.53 \pm 0.62}$	68.86 ± 0.13	64.19 ± 0.23	61.68 ± 0.14
GDA (DDU, ours)	97.86 ± 0.19	$\mathbf{91.34 \pm 0.04}$	$\mathbf{91.07 \pm 0.05}$	87.53 ± 0.62	$\mathbf{83.13 \pm 0.06}$	$\mathbf{73.12 \pm 0.19}$	$\mathbf{71.29 \pm 0.08}$

entropy of the model. On the other hand, to quantify the epistemic uncertainty, we use **i)** the softmax entropy, **ii)** the softmax density [46] or **iii)** the GMM feature density (as described in §3).

For the purposes of comparison, we also present scores obtained by a 5-Ensemble of the respective architectures (i.e. Wide-ResNet-28-10 and VGG-16) in Tab. 8 for CIFAR-10 vs SVHN/CIFAR-100 and in Tab. 9 for CIFAR-100 vs SVHN. Based on these results, we can make the following observations (in addition to the ones we make in §4.2):

Inductive biases are important for feature density. From the AUROC scores in Tab. 8, we can see that using the feature density of a GMM in VGG-16 without the proposed inductive biases yields significantly lower AUROC scores as compared to Wide-ResNet-28-10 with inductive biases. In fact, in none of the datasets is the feature density of a VGG able to outperform its corresponding ensemble. This provides yet more evidence (in addition to Fig. 1) to show that the GMM feature density alone cannot estimate epistemic uncertainty in a model that suffers from feature collapse. We need sensitivity and smoothness conditions (see §5) on the feature space of the model to obtain feature densities that

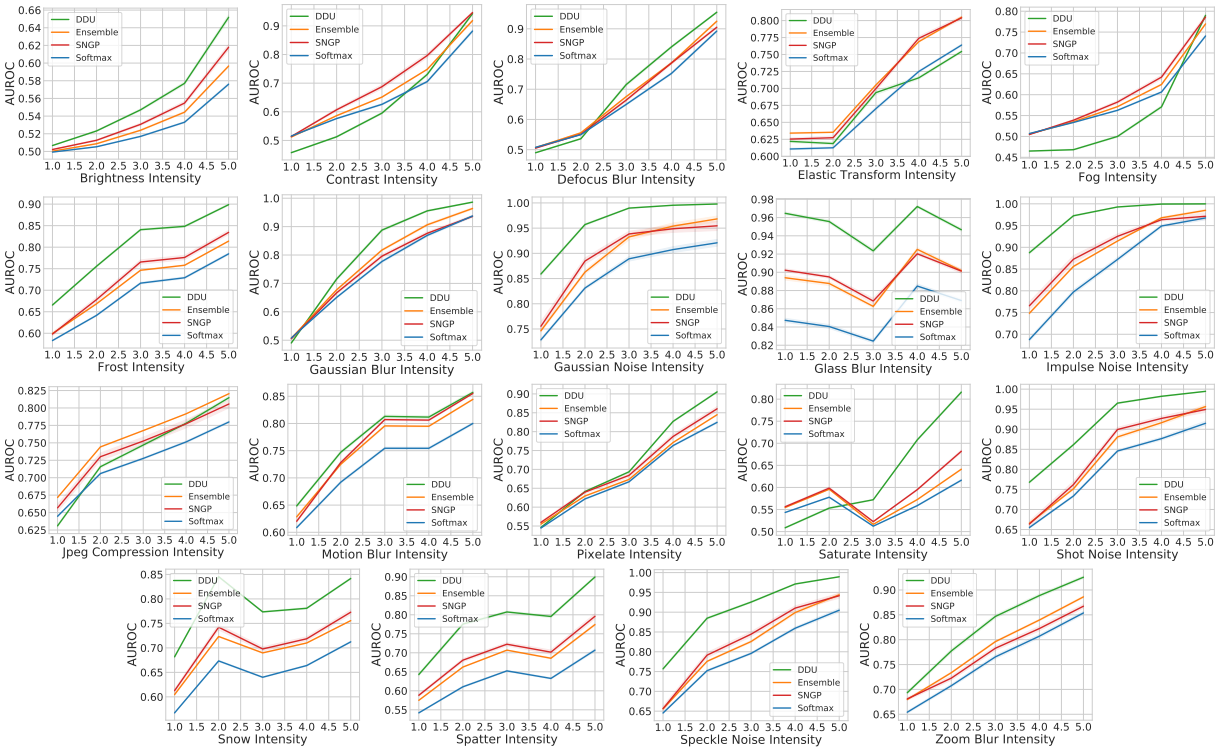


Figure 10. AUROC vs corruption intensity for all corruption types in CIFAR-10-C with Wide-ResNet-28-10 as the architecture and baselines: Softmax Entropy, Ensemble (using Predictive Entropy as uncertainty), SNGP and DDU feature density.

capture epistemic uncertainty.

Sensitivity creates a bigger difference than smoothness. We note that the difference between AUROC obtained from feature density between Wide-ResNet-28-10 models with and without spectral normalisation is minimal. Although Wide-ResNet-28-10 with spectral normalisation (i.e. smoothness constraints) still outperforms its counterpart without spectral normalisation, the small difference between the AUROC scores indicates that it might be the residual connections (i.e. sensitivity constraints) that make the model detect OoD samples better. This observation is also intuitive as a sensitive feature extractor should map OoD samples farther from iD ones.

DDU as a simple baseline. In DDU, we use the softmax output of a model to get aleatoric uncertainty. We use the GMM's feature-density to estimate the epistemic uncertainty. Hence, DDU does not suffer from miscalibration as the softmax outputs can be calibrated using post-hoc methods like temperature scaling. At the same time, the feature-densities of the model are not affected by temperature scaling and capture epistemic uncertainty well.

F. Additional Ablations & Toy Experiments

Here, we provide details for toy experiments from the main paper which are visualized in Fig. 1, Fig. 15 and Fig. 2.

Table 11. *ECE for Dirty-MNIST test set and AUROC for Dirty-MNIST vs Fashion-MNIST as proxies for aleatoric and epistemic uncertainty quality respectively.*

Model	ECE	AUROC for Softmax Entropy (↑)	AUROC for Feature Density (↑)
LeNet	2.22	84.23	71.41
VGG-16	**2.11**	84.04	89.01
ResNet-18+SN (DDU)	2.34	83.01	**99.91**

F.1. Motivational Example in Figure 1

In Fig. 1 we train a LeNet [41], a VGG-16 [60] and a ResNet-18 with spectral normalisation [22; 52] (ResNet-18+SN) on *Dirty-MNIST*, a modified version of MNIST [41] with additional ambiguous digits (Ambiguous-MNIST). *Ambiguous-MNIST* contains samples with multiple plausible labels and thus higher aleatoric uncertainty (see Fig. 1a). We refer to §B for details on how this dataset was generated. With ambiguous data having various levels of aleatoric uncertainty, Dirty-MNIST is more representative of real-world datasets compared to well-cleaned curated datasets, like MNIST and CIFAR-10, commonly used for benchmarking in ML [13; 39]. Moreover, Dirty-MNIST also poses a challenge for recent uncertainty estimation methods, which often confound aleatoric and epistemic uncertainty [65]. Figure 1b shows that the softmax entropy of a deterministic

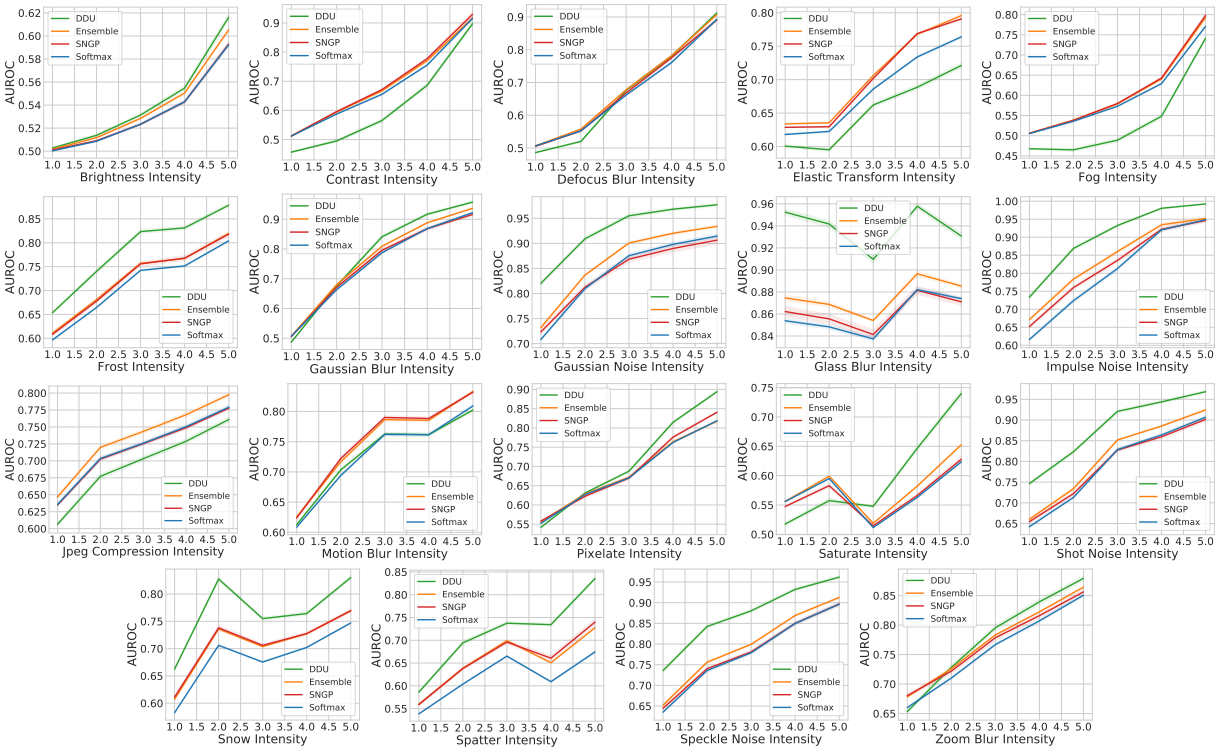


Figure 11. AUROC vs corruption intensity for all corruption types in CIFAR-10-C with ResNet-50 as the architecture and baselines: Softmax Entropy, Ensemble (using Predictive Entropy as uncertainty), SNGP and DDU feature density.

model is unable to distinguish between iD (Dirty-MNIST) and OoD (Fashion-MNIST [70]) samples as the entropy for the latter heavily overlaps with the entropy for Ambiguous-MNIST samples. However, the feature-space density of the model with our inductive biases in Fig. 1c captures epistemic uncertainty reliably and is able to distinguish iD from OoD samples. The same cannot be said for LeNet or VGG in Fig. 1c, whose densities are unable to separate OoD from iD samples. This demonstrates the importance of the inductive bias to ensure the sensitivity and smoothness of the feature space as we further argue below. Finally, Fig. 1b and Fig. 1c demonstrate that our method is able to separate aleatoric from epistemic uncertainty: samples with low feature density have high epistemic uncertainty, whereas those with both high feature density and high softmax entropy have high aleatoric uncertainty—note the high softmax entropy for the most ambiguous Ambiguous-MNIST samples in Fig. 1b.

F.1.1 Disentangling Epistemic and Aleatoric Uncertainty

We used a simple example in §1 to demonstrate that a single softmax model with a proper inductive bias can reliably capture epistemic uncertainty via its feature-space density and aleatoric uncertainty via its softmax entropy. To recreate the natural characteristics of uncurated real-world datasets, which contain ambiguous samples, we use MNIST [41] as an iD dataset of unambiguous samples, Ambiguous-MNIST as an iD dataset of ambiguous samples and Fashion-MNIST [70] as an OoD dataset (see Fig. 1a), with more details in §B. We train a LeNet [41], a VGG-16 [60] and a ResNet-18 [22] with spectral normalisation (SN) on Dirty-MNIST (a mix of Ambiguous- and standard MNIST) with the training setup detailed in §D.1.

Table 11 gives a quantitative evaluation of the qualitative results in §1. The AUROC metric reflects the quality of the epistemic uncertainty as it measures the probability that iD and OoD samples can be distinguished, and OoD samples are never seen during training while iD samples are semantically similar to training samples. The ECE metric measures the quality of the aleatoric uncertainty. The softmax outputs capture aleatoric uncertainty well, as expected, and all 3 models obtain similar ECE scores on the Dirty-MNIST test set. However, with an AUROC of around 84% for all the 3 models, on Dirty-MNIST vs Fashion-MNIST, we conclude that softmax entropy is unable to capture epistemic uncertainty well. This is reinforced in Fig. 1b, which shows a strong overlap between the softmax entropy of OoD and ambiguous iD samples. At the same time, the feature-space densities of LeNet and VGG-16, with AUROC scores around

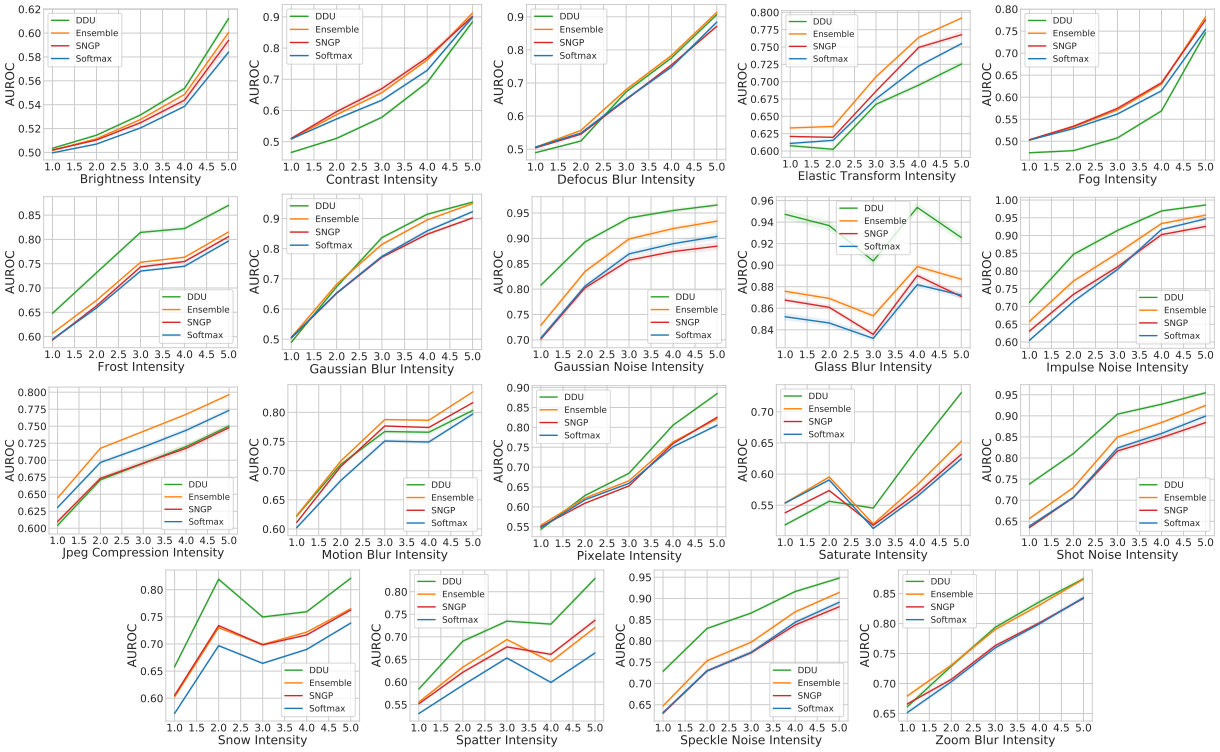


Figure 12. AUROC vs corruption intensity for all corruption types in CIFAR-10-C with ResNet-110 as the architecture and baselines: Softmax Entropy, Ensemble (using Predictive Entropy as uncertainty), SNGP and DDU feature density.

71% and 89% respectively, are unable to distinguish OoD from iD samples, indicating that simply using feature-space density without appropriate inductive biases (as seen in [43]) is not sufficient.

Only by fitting a GMM on top of a feature extractor with appropriate inductive biases (DDU) and using its feature density are we able to obtain performance far better (with AUROC of 99.9%*) than the alternatives in the ablation study (see Tab. 11, also noticeable in Fig. 1c).* The entropy of a softmax model can capture aleatoric uncertainty, even without additional inductive biases, but it *cannot* be used to estimate epistemic uncertainty (see §5). On the other hand, feature-space density can *only* be used to estimate epistemic uncertainty *when the feature extractor is sensitive and smooth*, as achieved by using a ResNet and spectral normalisation in DDU.

F.2. Effects of a well-regularized feature space on feature collapse

The effects of a well-regularized feature space on feature collapse are visible in Fig. 1. In the case of feature collapse, we must have *some* OoD inputs for which the features are mapped on top of the features of iD inputs. The distances of these OoD features to each class centroid must be equal to the distances of the corresponding iD inputs to class centroids, and hence the density for these OoD inputs must be equal to the density of the iD inputs. If the density histograms do not overlap, no feature collapse can be present[3]. We see no overlapping densities in Fig. 1(c, right), therefore we indeed have no feature collapse. First, the effects of having a well-regularized feature space on feature collapse can be analysed from Fig. 1. In case of feature collapse we must have *some* OoD features mapped ontop of iD features, therefore the distances of at least some OoD features to the class centroids must be equal to iD's distances, hence OoD density must overlap with iD density. We see this in Fig. 1(c) (left) in the case without a regularized feature space. On the other hand, when we regularize the feature space, we see the densities do not overlap, i.e. the distances of the features of OoD examples to the centroids are larger than the distances of iD examples (Fig. 1(c) right), hence feature collapse is not present.

F.3. QUBIQ Challenge

In this section, we evaluate DDU's performance on the real-world QUBIQ challenge related to biomedical imaging. QUBIQ has a total of 7 binary segmentation tasks in 4

[3]Note though that the opposite ('if the density histograms overlap then there must be feature collapse') needs not hold: the histograms can also overlap due to other reasons.

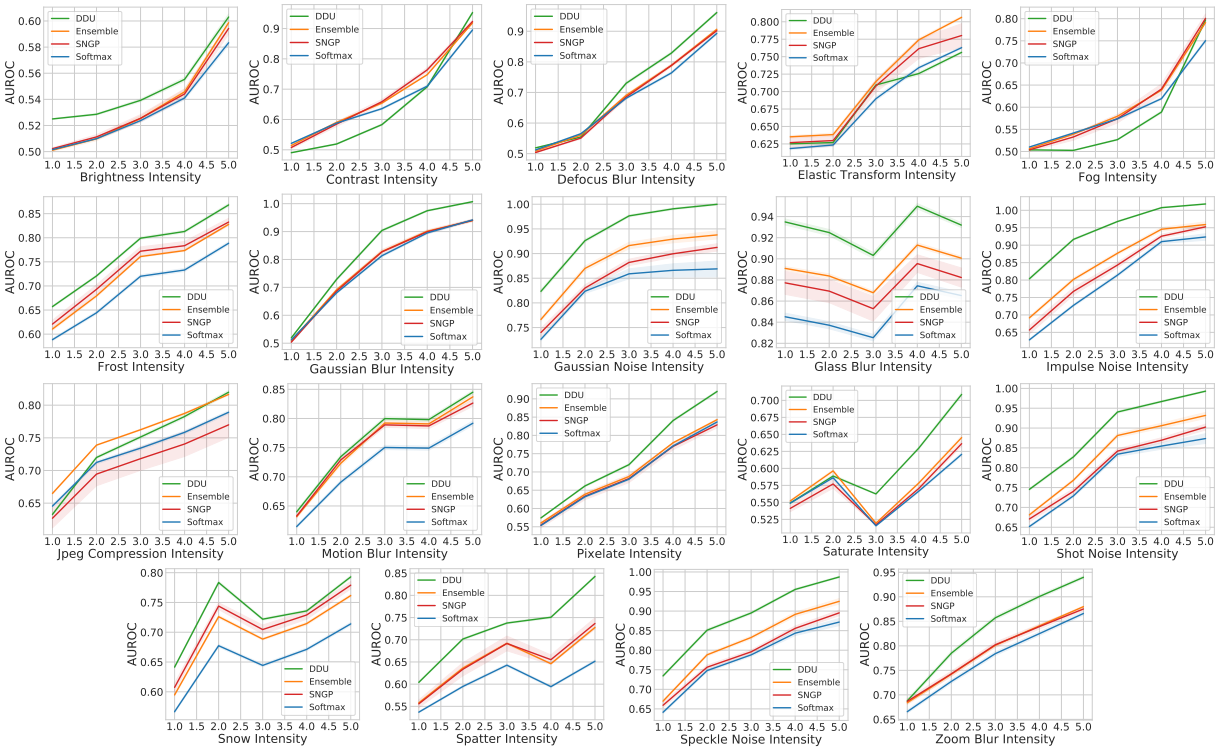


Figure 13. AUROC vs corruption intensity for all corruption types in CIFAR-10-C with DenseNet-121 as the architecture and baselines: Softmax Entropy, Ensemble (using Predictive Entropy as uncertainty), SNGP and DDU feature density.

Softmax	Energy	3-Ensemble PE	DDU
78.4 ± 1.31	77.31 ± 1.5	82.25 ± 0.83	$\mathbf{82.63 \pm 1.08}$

Table 12. Dice scores for the QUBIQ 2021 challenge

biomedical imaging datasets with multiple annotations per image. The task is to predict the distribution of source labels with a mask of values between 0 and 1. For evaluation, the annotations are averaged to provide a continuous ground-truth. The prediction mask and continuous ground-truth are binarized by thresholding between $[0, 1]$ and a Dice score is computed between the resulting binary masks. The average dice score across thresholds, images and tasks is reported. Note that in the continuous ground-truth, 0.5 indicates maximum uncertainty and values above or below indicate lower uncertainty. Thus, for our comparison, we scale all uncertainty values to the range $u \in [0, 0.5]$ and use $p + u$ if $p = 0$ and $p - u$ if $p = 1$, where p is the binary prediction. We use a UNet model with a ResNet encoder and report the Dice scores averaged over 5 runs in Tab. 12. Even on this real-world dataset, DDU performs as well as ensembles and outperforms Softmax and Energy baselines.

F.4. Additional Baselines

In this section, we provide an ablation with additional baselines on OoD detection for comparison with DDU. In particular, we provide comparisons with Feature Space Singularity (FSSD) [31], Batch Ensemble (BE) [67] and SWAG [49] using Wide-ResNet-28-10 as additional recent baselines. We also use the Wide-ResNet-28-10 feature extractor trained using SNGP loss and fit DDU (i.e., GDA) on its feature space. Since SNGP also uses a sensitive smooth feature space with residual connections and spectral normalization, its feature space makes for a good candidate to apply DDU. In Tab. 13, we provide the AUROC scores for models trained on CIFAR-10 (C10) and CIFAR-100 (C100). Broadly, DDU outperforms all competitive baselines. Additionally, we observe a broad improvement in AUROC when DDU is applied on the SNGP feature extractor as compared to vanilla SNGP. However, DDU on a feature extractor trained using softmax loss is still superior to DDU on the SNGP feature extractor.

F.5. Additional Calibration Metrics

In Tab. 14, in addition to ECE%, we provide additional calibration error scores: temperature scaled Thresholded Adaptive Calibration Error (TACE) % and Negative Log Likelihood (NLL) for Wide-ResNet-28-10 trained on CIFAR-10/100 (main results for this can be found in Tab. 1. The results for TACE and NLL are consistent with what we see for ECE. Ensembles produce the most calibrated models and among deterministic baselines, DDU is the best calibrated.

Train dataset	Method	AUROC (↑) *SVHN*	*CIFAR-100/10*	*Tiny-ImageNet*
C10	FSSD [31]	97.24	89.88	90.23
	BE [67]	95.36	87.63	88.14
	SWAG [49]	96.37	90.33	90.24
	SNGP [45]	94.0 ± 1.3	91.13 ± 0.15	89.97 ± 0.19
	5-Ensemble [40]	97.73 ± 0.31	$\mathbf{92.13 \pm 0.02}$	90.06 ± 0.03
	SNGP + DDU	96.47 ± 0.7	89.97 ± 0.13	90.3 ± 0.12
	DDU (Ours)	$\mathbf{97.86 \pm 0.19}$	91.34 ± 0.04	$\mathbf{91.07 \pm 0.05}$
		SVHN		*Tiny-ImageNet*
C100	FSSD [31]	**87.64**		82.2
	BE [67]	86.44		78.33
	SWAG [49]	81.41		81.67
	SNGP [45]	85.71 ± 0.81		78.85 ± 0.43
	5-Ensemble [40]	79.54 ± 0.91		82.95 ± 0.09
	SNGP + DDU	87.34 ± 0.76		79.62 ± 0.36
	DDU(Ours)	87.53 ± 0.62		$\mathbf{83.13 \pm 0.06}$

Table 13. OoD detection ablation with WRN-28-10 model with additional baselines, FSSD [31], Batch Ensemble (BE) [67] and SWAG [49] as well as using DDU with a feature extractor trained on SNGP. For comparison, we also provide performance for vanilla SNGP, deep ensemble and DDU.

Dataset	Metric	Softmax & Energy	DUQ	SNGP	DDU	5-Ensemble
C10	ECE%	$\mathbf{0.85 \pm 0.02}$	1.55 ± 0.08	1.8 ± 0.1	$\mathbf{0.85 \pm 0.04}$	$\mathbf{0.76 \pm 0.03}$
	TACE%	0.63 ± 0.01	0.84 ± 0.03	0.9 ± 0.04	$\mathbf{0.61 \pm 0.01}$	$\mathbf{0.48 \pm 0.01}$
	NLL	0.18 ± 0.06	0.23 ± 0.07	0.27 ± 0.08	$\mathbf{0.16 \pm 0.06}$	$\mathbf{0.11 \pm 0.02}$
C100	ECE%	4.62 ± 0.06	-	4.33 ± 0.01	$\mathbf{4.1 \pm 0.08}$	$\mathbf{3.32 \pm 0.09}$
	TACE%	1.31 ± 0.02	-	1.23 ± 0.04	$\mathbf{1.06 \pm 0.03}$	$\mathbf{0.58 \pm 0.03}$
	NLL	1.17 ± 0.13	-	0.92 ± 0.16	$\mathbf{0.86 \pm 0.14}$	$\mathbf{0.73 \pm 0.09}$

Table 14. Calibration error scores ECE%, TACE% and NLL for WRN-28-10.

F.6. Two Moons

In this section, we evaluate DDU's performance on a well-known toy setup: the Two Moons dataset. We use scikit-learn's *datasets* package to generate 2000 samples with a noise rate of 0.1. We use a 4-layer fully connected architecture, ResFFN-4-128 with 128 neurons in each layer and a residual connection, following [45]. As an ablation, we also train using a 4-layer fully connected architecture with 128 neurons in each layer, but *without the residual connection*. We name this architecture FC-Net. The input is 2-dimensional and is projected into the 128 dimensional space using a fully connected layer. Using the ResFFN-4-128 architecture we train 3 baselines:

1. **Softmax:** We train a single softmax model and use the softmax entropy as the uncertainty metric.
2. **3-ensemble:** We train an ensemble of 3 softmax models and use the predictive entropy of the ensemble as the measure of uncertainty.
3. **DDU:** We train a single softmax model applying spectral normalization on the fully connected layers and using the feature density as the measure of model confidence.

Each model is trained using the Adam optimiser for 150 epochs. In Fig. 14, we show the uncertainty results for all the above 3 baselines. It is clear that both the softmax entropy as well as the predictive entropy of the ensemble is uncertain only along the decision boundary between the two classes whereas DDU is confident only on the data distribution and is not confident anywhere else. It is worth mentioning that even DUQ and SNGP perform well in this setup and deep ensembles have been known to underperform in the Two-Moons setup primarily due to the simplicity of the dataset causing all the ensemble components to generalise in the same way. Finally, also note that the feature space density of FC-Net without residual connections is not able to capture uncertainty well (see Fig. 14d), thereby reaffirming our claim that proper inductive biases are indeed a necessary component to ensure that feature space density captures uncertainty reliably. Thus, in addition to its excellent performance in active learning, CIFAR-10 vs SVHN/CIFAR-100/Tiny-ImageNet/CIFAR-10-C, CIFAR-100 vs SVHN/Tiny-ImageNet, and ImageNet vs ImageNet-O, we note that DDU captures uncertainty reliably even in a small 2D setup like Two Moons.

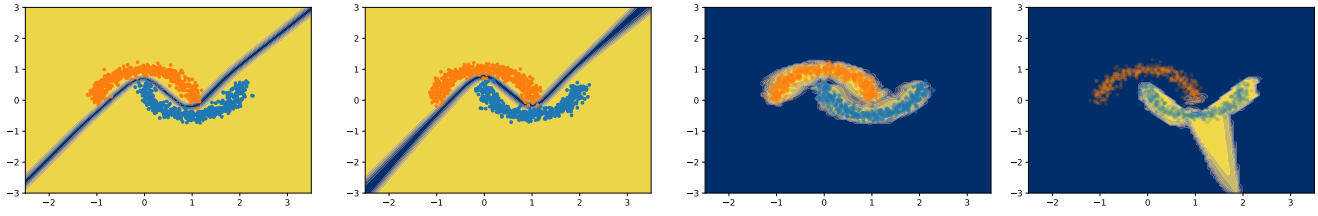
F.7. 5-Ensemble Visualisation

In Fig. 15, we provide a visualisation of a 5-ensemble (with five deterministic softmax networks) to see how softmax entropy fails to capture epistemic uncertainty precisely because the mutual information (MI) of an ensemble does not (see §5). We train the networks on 1-dimensional data with binary labels 0 and 1. The data is shown in Fig. 15a. From Fig. 15a and Fig. 15b, we find that the softmax entropy is high in regions of ambiguity where the label can be both 0 and 1 (i.e. x between -4.5 and -3.5, and between 3.5 and 4.5). This indicates that softmax entropy can capture aleatoric uncertainty. Furthermore, in the x interval $(-2, 2)$, we find that the deterministic softmax networks disagree in their predictions (see Fig. 15a) and have softmax entropies which can be high, low or anywhere in between (see Fig. 15b) following our claim in §5. In fact, this disagreement is the very reason why the MI of the ensemble is high in the interval $(-2, 2)$, thereby reliably capturing epistemic uncertainty. Finally, note that the predictive entropy (PE) of the ensemble is high both in the OoD interval $(-2, 2)$ as well as at points of ambiguity (i.e. at -4 and 4). This indicates that the PE of a Deep Ensemble captures both epistemic and aleatoric uncertainty well. From these visualisations, we draw the conclusion that the softmax entropy of a deterministic softmax model cannot capture epistemic uncertainty precisely because the MI of a Deep Ensemble can.

F.8. Feature-Space Density & Epistemic Uncertainty vs Softmax Entropy & Aleatoric Uncertainty

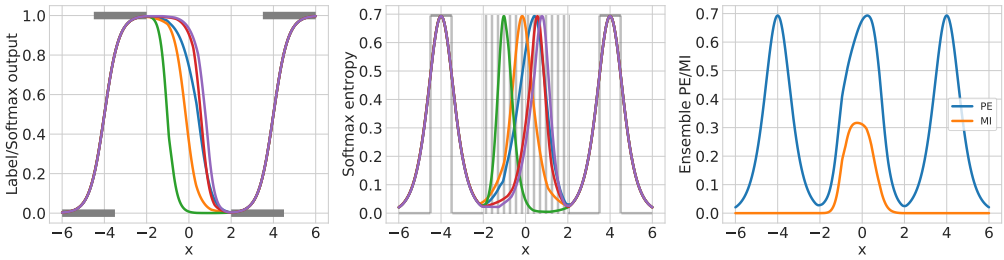
To empirically verify the connection between feature-space density and epistemic uncertainty on the one hand and the connection between softmax entropy and aleatoric uncertainty on the other hand, we train ResNet-18+SN models on increasingly large subsets of DirtyMNIST and CIFAR-10 and evaluate the epistemic and aleatoric uncertainty on the corresponding test sets using the feature-space density and softmax entropy, respectively. Moreover, we also train a 5-ensemble on the same subsets of data and use the ensemble's mutual information as a baseline measure of epistemic uncertainty.

In Fig. 2, Fig. 16 and Tab. 15, we see that with larger



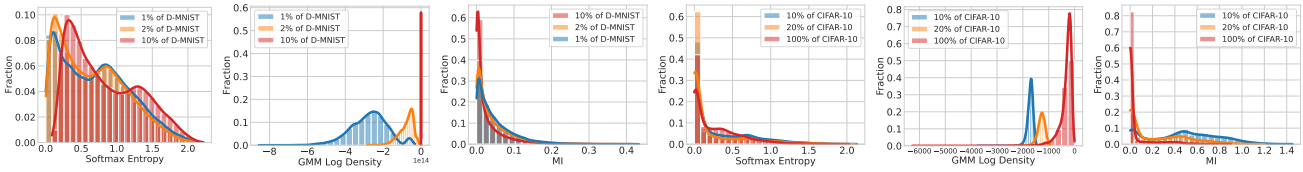
(a) Softmax Entropy (b) Ensemble Predictive Entropy (c) DDU Feature-space density (d) FC-Net Feature-space density

Figure 14. *Uncertainty on Two Moons dataset*. Blue indicates high uncertainty and yellow indicates low uncertainty. Both the softmax entropy of a single model as well as the predictive entropy of a deep ensemble are uncertain only along the decision boundary whereas the feature-space density of DDU is uncertain everywhere except on the data distribution (the ideal behaviour). However, the feature density of a normal fully connected network (FC-Net) without any inductive biases can't capture uncertainty properly.



(a) Softmax output (b) Softmax entropy (c) 5-Ensemble

Figure 15. *Softmax outputs & entropies for 5 softmax models along with the predictive entropy (PE) and mutual information (MI) for the resulting 5-Ensemble.* (a) and (b) show that the softmax entropy is only reliably high for ambiguous iD points ($\pm$3.5–4.5), whereas it can be low or high for OoD points (-2–2). The different colors are the different ensemble components. Similarly, (c) shows that the MI of the ensemble is only high for OoD, whereas the PE is high for both OoD and for regions of ambiguity. See §F.7.



(a) SE D-MNIST (b) Density D-MNIST (c) MI D-MNIST (d) SE CIFAR-10 (e) Density CIFAR-10 (f) MI CIFAR-10

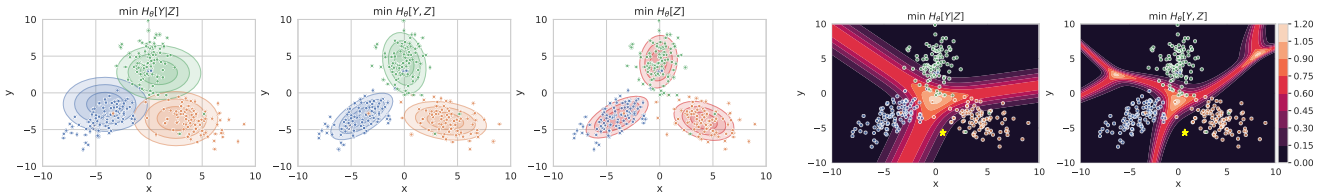
Figure 16. *Comparison of epistemic and aleatoric uncertainty captured b(ResNet-18+SN) on increasingly large subsets of Dirty-MNIST and CIFAR-10. Clearly, feature density captures epistemic uncertainty which reduces when the model is trained on increasingly large subsets of training data, whereas softmax entropy (SE) does not. For comparison, we also plot a deep-ensemble's epistemic uncertainty, through mutual information (MI) for the same settings. For more details, see Tab. 15.*

training sets, the average feature-space density increases which is consistent with the epistemic uncertainty decreasing as more data is available as reducible uncertainty. This is also evident from the consistent strong positive correlation between the negative log density and mutual information of the ensemble.

On the other hand, the softmax entropy stays roughly the same which is consistent with aleatoric uncertainty as irreducible uncertainty, which is independent of the training data. Importantly, all of this is also consistent with the experiments comparing epistemic and aleatoric uncertainty on increasing training set sizes in Table 3 of [33].

F.9. Objective Mismatch Ablation with Wide-ResNet-28-10 on CIFAR-10

We further validate Proposition 5.3 by running an ablation on Wide-ResNet-28-10 on CIFAR-10. Table 16 shows that the feature-space density estimator indeed performs worse than the softmax layer for aleatoric uncertainty (accuracy and ECE).



(a) *Density.* Contours at 68.26%, 95.44%, and 99.7%.

(b) *Entropy.* Darker is lower.

Figure 17. *3-component GMM fitted to a synthetic dataset with 3 different classes (differently colored) with 4% label noise using different objectives*. **(a):** The optimas for conditional log-likelihood $\mathbb{H}_\theta[Y \mid Z]$, joint log-likelihood $\mathbb{H}_\theta[Y, Z]$, and marginalised log-likehood $\mathbb{H}_\theta[Z]$ all differ. Hence, the best calibrated model ($\mathbb{H}_\theta[Y \mid Z]$) will not provide the best density estimate ($\mathbb{H}_\theta[Z]$), and vice-versa. **(b):** A mixture model that optimizes $\mathbb{H}_\theta[Y, Z]$ (GDA) does not have calibrated decision boundaries for aleatoric uncertainty: the ambiguous sample (due to label noise) marked by the yellow star has no aleatoric uncertainty under the GDA model. See §G.2.3 for details.

Table 15. *Average softmax entropy (SE) and feature-space density of the test set for models trained on different amounts of the training set (Dirty-MNIST and CIFAR-10) behave consistently with aleatoric and epistemic uncertainty.* Aleatoric uncertainty for individual samples does not change much as more data is added to the training set while epistemic uncertainty decreases as more data is added. This is also consistent with Table 3 in [33]. Finally, we observe a consistent strong positive correlation between the negative log feature space density and the mutual information (MI) of a deep ensemble trained on the same subsets of data for both Dirty-MNIST and CIFAR-10. However, the correlation between softmax entropy and MI is not consistent.

Training Set	Avg Test SE (≈)	Avg Test Log GMM Density (↑)	Avg Test MI	Correlation(SE ‖ MI)	Correlation(-Log GMM Density ‖ MI)
1% of D-MNIST	0.7407	$-2.7268e+14$	0.0476		
2% of D-MNIST	0.6580	$-7.8633e+13$	0.0447	−0.79897	0.8132
10% of D-MNIST	0.8295	−1279.1753	0.0286		
10% of CIFAR-10	0.3189	−1715.3516	0.4573		
20% of CIFAR-10	0.2305	−1290.1726	0.2247	0.5663	0.9556
100% of CIFAR-10	0.2747	−324.8040	0.0479		

Table 16. *Objective Mismatch Ablation with WideResNet-28-10 models with and without spectral normalisation on CIFAR-10.* While GMMs perform much better than Softmax Entropy for feature-space density/epistemic uncertainty estimation, they underperform for aleatoric uncertainty estimation: both accuracy and in particular ECE are much worse than a regular softmax layer. Averaged over 25 runs.

Model	Prediction Source	Accuracy in % (↑)	ECE (↓)
WideResNet-28-10	Softmax	95.98 ± 0.02	2.29 ± 0.02
	GMM	95.86 ± 0.02	4.13 ± 0.02
WideResNet-28-10+SN	Softmax	95.97 ± 0.03	2.23 ± 0.03
	GMM	95.88 ± 0.02	4.12 ± 0.02

G. Theoretical Results

G.1. Softmax entropy "cannot" capture epistemic uncertainty because Deep Ensembles "can"

G.1.1 Qualitative Statement

We start with a proof of Proposition 5.2, which quantitatively examines the qualitative statemets that given the same predictive entropy, higher epistemic uncertainty for one point than another will cause some ensemble members to have lower softmax entropy.

Proposition 5.2. *Let x_1 and x_2 be points such that x_1 has **higher** epistemic uncertainty than x_2 under the ensemble: $\mathbb{I}[Y_1; \omega \,|\, x_1, \mathcal{D}] > \mathbb{I}[Y_2; \omega \,|\, x_2, \mathcal{D}] + \delta$, $\delta \geq 0$. Further assume both have similar predictive entropy $|\, \mathbb{H}[Y_1 \mid x_1, \mathcal{D}] - \mathbb{H}[Y_2 \mid x_2, \mathcal{D}] | \leq \epsilon$, $\epsilon \geq 0$. Then, there exist sets of ensemble members Ω with $\mathrm{p}(\Omega \mid \mathcal{D}) > 0$, such that for all softmax models $\omega \in \Omega$ the softmax entropy of x_1 is **lower** than the softmax entropy of x_2: $\mathbb{H}[Y_1 \,|\, x_1, \omega] < \mathbb{H}[Y_2 \,|\, x_2, \omega] - (\delta - \epsilon)$.*

Proof. From Eq. (1), we obtain

$$\begin{aligned} &|\, \mathbb{I}[Y_1; \omega \mid x_1, \mathcal{D}] + \mathbb{E}_{\mathrm{p}(\omega|\mathcal{D})} \left[\mathbb{H}[Y_1 \mid x_1, \omega]\right] \\ &- \mathbb{I}[Y_2; \omega \mid x_2, \mathcal{D}] - \mathbb{E}_{\mathrm{p}(\omega|\mathcal{D})} \left[\mathbb{H}[Y_2 \mid x_2, \omega]\right] | \leq \epsilon. \end{aligned} \tag{2}$$

and hence we have

$$\begin{aligned} &\mathbb{E}_{\mathrm{p}(\omega|\mathcal{D})} \left[\mathbb{H}[Y_1 \mid x_1, \omega]\right] - \mathbb{E}_{\mathrm{p}(\omega|\mathcal{D})} \left[\mathbb{H}[Y_2 \mid x_2, \omega]\right] \\ &+ \underbrace{\left(\mathbb{I}[Y_1; \omega \mid x_1, \mathcal{D}] - \mathbb{I}[Y_2; \omega \mid x_2, \mathcal{D}]\right)}_{>\delta} \leq \epsilon. \end{aligned} \tag{3}$$

We rearrange the terms:

$$\mathbb{E}_{\mathrm{p}(\omega|\mathcal{D})} \left[\mathbb{H}[Y_1 \mid x_1, \omega]\right] < \mathbb{E}_{\mathrm{p}(\omega|\mathcal{D})} \left[\mathbb{H}[Y_2 \mid x_2, \omega]\right] - (\delta - \epsilon). \tag{4}$$

Now, the statement follows by contraposition: if $\mathbb{H}[Y_1 \mid x_1, \omega] \geq \mathbb{E}_{\mathrm{p}(\omega|\mathcal{D})} \left[\mathbb{H}[Y_2 \mid x_2, \omega]\right] - (\delta - \epsilon)$ for all ω, the monotonicity of the expectation would yield $\mathbb{E}_{\mathrm{p}(\omega|\mathcal{D})} \left[\mathbb{H}[Y_1 \mid x_1, \omega]\right] \geq \mathbb{E}_{\mathrm{p}(\omega|\mathcal{D})} \left[\mathbb{H}[Y_2 \mid x_2, \omega]\right] - (\delta - \epsilon)$. Thus, there is a non-null-set Ω' with $\mathrm{p}(\Omega') > 0$, such that

$$\mathbb{H}[Y_1 \mid x_1, \omega] < \mathbb{H}[Y_2 \mid x_2, \omega] - (\delta - \epsilon), \tag{5}$$

for all $\omega \in \Omega'$. □

While this statement provides us with an intuition for why ensemble members and thus deterministic models cannot provide epistemic uncertainty reliably through their softmax entropies, we can examine this further by establishing some upper bounds.

G.1.2 Infinite Deep Ensemble

There are two interpretations of the ensemble parameter distribution $\mathrm{p}(\omega \,|\, \mathcal{D})$: we can view it as an empirical distribution given a specific ensemble with members $\omega_{i \in \{1,\dots,K\}}$, or we can view it as a distribution over all possible trained models, given: random weight initializations, the dataset, stochasticity in the minibatches and the optimization process. In that case, any Deep Ensemble with K members can be seen as finite Monte-Carlo sample of this posterior distribution. The predictions of an ensemble then are an unbiased estimate of the predictive distribution $\mathbb{E}_{\mathrm{p}(\omega|\mathcal{D})}\left[\mathrm{p}(y|x,\omega|)\right]$, and similarly the expected information gain computed using the members of the Deep Ensemble is just a (biased) estimator of $\mathbb{I}[Y; \omega \mid x, \mathcal{D}]$.

G.1.3 Analysis of Softmax Entropy of a Single Deterministic Model on OoD Data using Properties of Deep Ensembles

Based on the interpretation of Deep Ensembles as a distribution over model parameters, we can walk backwards and, given *some value* for the predictive distribution and epistemic uncertainty of a Deep Ensemble, estimate what the softmax entropies from each ensemble component must have been. I.e. if we observe Deep Ensembles to have high epistemic uncertainty on OoD data, we can deduce from that what the softmax entropy of deterministic neural nets (the ensemble components) must look like. More specifically, given a predictive distribution $\mathrm{p}(y \mid x)$ and epistemic uncertainty, that is expected information gain $\mathbb{I}[Y; \omega \mid x]$, of the infinite Deep Ensemble, we estimate the expected softmax entropy from a single deterministic model, considered as a sample $\omega \sim \mathrm{p}(\omega \mid \mathcal{D})$ and model the variance. Empirically, we find the real variance to be higher by a large amount for OoD samples, showing that softmax entropies do not capture epistemic uncertainty well for samples with high epistemic uncertainty.

We will need to make several strong assumptions that limit the generality of our estimation, but we can show that our analysis models the resulting softmax entropy distributions appropriately. This will show that deterministic softmax models can have widely different entropies and confidence values.

Given the predictive distribution $\mathrm{p}(y \mid x)$ and epistemic uncertainty $\mathbb{I}[Y; \omega \mid x]$, we can approximate the distribution over softmax probability vectors $p(y|x,\omega)$ for different ω using its maximum-entropy estimate: a Dirichlet distribution $(Y_1, \dots, Y_K) \sim \mathrm{Dir}(\alpha)$ with non-negative concentration parameters $\alpha = (\alpha_1, \dots, \alpha_K)$ and $\alpha_0 := \sum \alpha_i$. Note that the Dirichlet distribution is used *only as an analysis tool*, and at no point do we need to actually fit Dirichlet distributions to our data.

Preliminaries

Before we can establish our main result, we need to look more closely at Dirichlet-Multinomial distributions. Given a Dirichlet distribution $\mathrm{Dir}(\alpha)$ and a random variable $\mathbf{p} \sim \mathrm{Dir}(\alpha)$, we want to quantify the expected entropy $\mathbb{E}_{\mathbf{p}\sim\mathrm{Dir}(\alpha)}\, \mathbb{H}_{Y\sim\mathrm{Cat}(\mathbf{p})}[Y]$ and its variance $\mathrm{Var}_{\mathbf{p}\sim\mathrm{Dir}(\alpha)}\, \mathbb{H}_{Y\sim\mathrm{Cat}(\mathbf{p})}[Y]$. For this, we need to develop more theory. In the following, Γ denotes the Gamma function, ψ denotes the Digamma function, ψ' denotes the Trigamma function.

Lemma G.1. *Given a Dirichlet distribution and random variable $\mathbf{p} \sim \mathrm{Dir}(\alpha)$, the following hold:*
1. The expectation $\mathbb{E}\left[\log \mathbf{p}_i\right]$ is given by:

$$\mathbb{E}\left[\log \mathbf{p}_i\right] = \psi(\alpha_i) - \psi(\alpha_0). \tag{6}$$

2. The covariance $\mathrm{Cov}[\log \mathbf{p}_i, \log \mathbf{p}_j]$ is given by

$$\mathrm{Cov}[\log \mathbf{p}_i, \log \mathbf{p}_j] = \psi'(\alpha_i)\, \delta_{ij} - \psi'(\alpha_0). \tag{7}$$

3. The expectation $\mathbb{E}\left[\mathbf{p}_i^n \mathbf{p}_j^m \log \mathbf{p}_i\right]$ is given by:

$$\begin{aligned} &\mathbb{E}\left[\mathbf{p}_i^n \mathbf{p}_j^m \log \mathbf{p}_i\right] \\ &\quad = \frac{\alpha_i^{\overline{n}}\, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n+m}}} \left(\psi(\alpha_i + n) - \psi(\alpha_0 + n + m)\right), \end{aligned} \tag{8}$$

where $i \neq j$, and $n^{\overline{k}} = n\,(n+1)\,\dots\,(n+k-1)$ denotes the rising factorial.

Proof. 1. The Dirichlet distribution is members of the exponential family. Therefore the moments of the sufficient statistics are given by the derivatives of the partition function with respect to the natural parameters. The natural parameters of the Dirichlet distribution are just its concentration parameters α_i. The partition function is

$$A(\alpha) = \sum_{i=1}^{k} \log \Gamma\left(\alpha_i\right) - \log \Gamma\left(\alpha_0\right), \tag{9}$$

the sufficient statistics is $T(x) = \log x$, and the expectation $\mathbb{E}\left[T\right]$ is given by

$$\mathbb{E}\left[T_i\right] = \frac{\partial A(\alpha)}{\partial \alpha_i} \tag{10}$$

as the Dirichlet distribution is a member of the exponential family. Substituting the definitions and evaluating the partial derivative yields

$$\mathbb{E}\left[\log \mathbf{p}_i\right] = \frac{\partial}{\partial \alpha_i} \left[\sum_{i=1}^{k} \log \Gamma\left(\alpha_i\right) - \log \Gamma\left(\sum_{i=1}^{k} \alpha_i\right)\right] \tag{11}$$

$$= \psi\left(\alpha_i\right) - \psi\left(\alpha_0\right) \frac{\partial}{\partial \alpha_i} \alpha_0, \tag{12}$$

where we have used that the Digamma function ψ is the log derivative of the Gamma function $\psi(x) = \frac{d}{dx} \ln \Gamma(x)$. This proves (6) as $\frac{\partial}{\partial \alpha_i} \alpha_0 = 1$.

2. Similarly, the covariance is obtained using a second-order partial derivative:

$$\mathrm{Cov}[T_i, T_j] = \frac{\partial^2 A(\alpha)}{\partial \alpha_i \, \partial \alpha_i}. \tag{13}$$

Again, substituting yields

$$\mathrm{Cov}[\log \mathbf{p}_i, \log \mathbf{p}_j] = \frac{\partial}{\partial \alpha_j} \left[\psi\left(\alpha_i\right) - \psi\left(\alpha_0\right)\right] \tag{14}$$

$$= \psi'\left(\alpha_i\right) \delta_{ij} - \psi'\left(\alpha_0\right). \tag{15}$$

3. We will make use of a simple reparameterization to prove the statement using Eq. (6). Expanding the expectation and substituting the density $\mathrm{Dir}(\mathbf{p}; \alpha)$, we obtain

$$\mathbb{E}\left[\mathbf{p}_i^n \mathbf{p}_j^m \log \mathbf{p}_i\right] = \int \mathrm{Dir}(\mathbf{p}; \alpha) \, \mathbf{p}_i^n \, \mathbf{p}_j^m \, \log \mathbf{p}_i \, d\mathbf{p} \tag{16}$$

$$= \int \frac{\Gamma\left(\alpha_0\right)}{\prod_{i=1}^K \Gamma\left(\alpha_i\right)} \prod_{k=1}^K \mathbf{p}_k^{\alpha_k - 1} \, \mathbf{p}_i^n \, \mathbf{p}_j^m \, \log \mathbf{p}_i \, d\mathbf{p} \tag{17}$$

$$= \frac{\Gamma(\alpha_i + n)\Gamma(\alpha_j + m)\Gamma(\alpha_0 + n + m)}{\Gamma(\alpha_i)\Gamma(\alpha_j)\Gamma(\alpha_0)} \int \mathrm{Dir}(\hat{\mathbf{p}}; \hat{\alpha}) \, \hat{\mathbf{p}}_i^n \, \hat{\mathbf{p}}_j^m \, \log \hat{\mathbf{p}}_i \, d\hat{\mathbf{p}} \tag{18}$$

$$= \frac{\alpha_i^{\overline{n}} \, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n+m}}} \, \mathbb{E}\left[\log \hat{\mathbf{p}}_i\right], \tag{19}$$

where $\hat{\mathbf{p}} \sim \mathrm{Dir}(\hat{\alpha})$ with $\hat{\alpha} = (\alpha_0, \ldots, \alpha_i + n, \ldots, \alpha_j + m, \ldots, \alpha_K)$ and we made use of the fact that $\frac{\Gamma(z+n)}{\Gamma(z)} = z^{\overline{n}}$. Finally, we can apply Eq. (6) on $\hat{\mathbf{p}} \sim \mathrm{Dir}(\hat{\alpha})$ to show

$$= \frac{\alpha_i^{\overline{n}} \, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n+m}}} \left(\psi(\alpha_i + n) - \psi(\alpha_0 + n + m)\right). \tag{20}$$

□

With this, we can already quantify the expected entropy $\mathbb{E}_{\mathbf{p} \sim \mathrm{Dir}(\alpha)} \mathbb{H}_{Y \sim \mathrm{Cat}(\mathbf{p})}[Y]$:

Lemma G.2. *Given a Dirichlet distribution and a random variable* $\mathbf{p} \sim Dir(\alpha)$*, the expected entropy* $\mathbb{E}_{\mathbf{p} \sim \mathrm{Dir}(\alpha)} \mathbb{H}_{Y \sim \mathrm{Cat}(\mathbf{p})}[Y]$ *of the categorical distribution* $Y \sim \mathrm{Cat}(\mathbf{p})$ *is given by*

$$\mathbb{E}_{\mathrm{p}(\mathbf{p}|\alpha)} \mathbb{H}[Y \mid \mathbf{p}] = \psi(\alpha_0 + 1) - \sum_{y=1}^K \frac{\alpha_i}{\alpha_0} \psi(\alpha_i + 1). \tag{21}$$

Proof. Applying the sum rule of expectations and Eq. (8) from Lemma G.1, we can write

$$\mathbb{E}\, \mathbb{H}[Y \mid \mathbf{p}] = \mathbb{E}\left[-\sum_{i=1}^K \mathbf{p}_i \log \mathbf{p}_i\right] = -\sum_i \mathbb{E}\left[\mathbf{p}_i \log \mathbf{p}_i\right] \tag{22}$$

$$= -\sum_i \frac{\alpha_i}{\alpha_0} \left(\psi(\alpha_i + 1) - \psi(\alpha_0 + 1)\right). \tag{23}$$

The result follows after rearranging and making use of $\sum_i \frac{\alpha_i}{\alpha_0} = 1$. □

With these statements, we can answer a slightly more complex problem:

Lemma G.3. *Given a Dirichlet distribution and a random variable* $\mathbf{p} \sim \mathrm{Dir}(\alpha)$*, the covariance* $\mathrm{Cov}[\mathbf{p}_i^n \log \mathbf{p}_i, \mathbf{p}_j^m \log \mathbf{p}_j]$ *is given by*

$$\mathrm{Cov}[\mathbf{p}_i^n \log \mathbf{p}_i, \mathbf{p}_j^m \log \mathbf{p}_j] \tag{24}$$

$$= \frac{\alpha_i^{\overline{n}} \, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n+m}}} \left(\left(\psi(\alpha_i + n) - \psi(\alpha_0 + n + m)\right)\right.$$
$$\left(\psi(\alpha_j + m) - \psi(\alpha_0 + n + m)\right)$$
$$\left. -\psi'(\alpha_0 + n + m)\right) \tag{25}$$
$$+ \frac{\alpha_i^{\overline{n}} \, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n}} \, \alpha_0^{\overline{m}}} (\psi(\alpha_i + n) - \psi(\alpha_0 + n))$$
$$(\psi(\alpha_j + m) - \psi(\alpha_0 + n)),$$

for $i \neq j$*, where* ψ *is the Digamma function and* ψ' *is the Trigamma function. Similarly, the covariance* $\mathrm{Cov}[\mathbf{p}_i^n \log \mathbf{p}_i, \mathbf{p}_i^m \log \mathbf{p}_i]$ *is given by*

$$\mathrm{Cov}[\mathbf{p}_i^n \log \mathbf{p}_i, \mathbf{p}_i^m \log \mathbf{p}_i] \tag{26}$$

$$= \frac{\alpha_i^{\overline{n+m}}}{\alpha_0^{\overline{n+m}}} \left(\left(\psi(\alpha_i + n + m) - \psi(\alpha_0 + n + m)\right)^2\right.$$
$$\left. + \psi'(\alpha_i + n + m) - \psi'(\alpha_0 + n + m)\right)$$
$$+ \frac{\alpha_i^{\overline{n}} \, \alpha_i^{\overline{m}}}{\alpha_0^{\overline{n}} \, \alpha_0^{\overline{m}}} (\psi(\alpha_i + n) - \psi(\alpha_0 + n))$$
$$(\psi(\alpha_i + m) - \psi(\alpha_0 + n)). \tag{27}$$

Regrettably, the equations are getting large. By abuse of notation, we introduce a convenient shorthand before proving the lemma.

Definition G.4. We will denote by

$$\overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^{n,m}\right]} = \psi(\alpha_i + n) - \psi(\alpha_0 + n + m), \tag{28}$$

and use $\overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^n\right]}$ for $\overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^{n,0}\right]}$. Likewise,

$$\overline{\mathrm{Cov}[\log \hat{\mathbf{p}}_i^{n,m}, \log \hat{\mathbf{p}}_j^{n,m}]} = \psi'(\alpha_i + n)\delta_{ij} - \psi'(\alpha_0 + n + m). \tag{29}$$

This notation agrees with the proof of Eq. (6) and (7) in Lemma G.1. With this, we can significantly simplify the previous statements:

Corollary G.5. *Given a Dirichlet distribution and random variable* $\mathbf{p} \sim \mathrm{Dir}(\alpha)$*,*

$$\mathbb{E}\left[\mathbf{p}_i^n \mathbf{p}_j^m \log \mathbf{p}_i\right] = \frac{\alpha_i^{\overline{n}}\, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n+m}}} \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^{n,m}\right]}, \tag{30}$$

$$\begin{aligned} &\mathrm{Cov}[\mathbf{p}_i^n \log \mathbf{p}_i, \mathbf{p}_j^m \log \mathbf{p}_j] \qquad (31)\\ &= \frac{\alpha_i^{\overline{n}}\, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n+m}}} \Big(\overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^{n,m}\right]} \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_j^{m,n}\right]} \\ &\qquad \overline{\mathrm{Cov}[\log \hat{\mathbf{p}}_i^{n,m}, \log \hat{\mathbf{p}}_j^{n,m}]}\Big) \\ &\quad + \frac{\alpha_i^{\overline{n}}\, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n}}\, \alpha_0^{\overline{m}}} \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^{n}\right]} \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_j^{m}\right]} \quad \textit{for } i \neq j\textit{, and} \\ &\qquad (32) \end{aligned}$$

$$\begin{aligned} &\mathrm{Cov}[\mathbf{p}_i^n \log \mathbf{p}_i, \mathbf{p}_i^m \log \mathbf{p}_i] \qquad (33)\\ &= \frac{\alpha_i^{\overline{n+m}}}{\alpha_0^{\overline{n+m}}} \Big(\overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^{n+m}\right]}^2 \\ &\quad + \overline{\mathrm{Cov}[\log \hat{\mathbf{p}}_i^{n+m}, \log \hat{\mathbf{p}}_i^{n+m}]}\Big) \qquad (34)\\ &\quad + \frac{\alpha_i^{\overline{n}}\, \alpha_i^{\overline{m}}}{\alpha_0^{\overline{n}}\, \alpha_0^{\overline{m}}} \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^{n}\right]} \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_j^{m}\right]}. \end{aligned}$$

Proof of Lemma G.3. This proof applies the well-know formula **(cov)** $\mathrm{Cov}[X, Y] = \mathbb{E}[X\,Y] - \mathbb{E}[X]\,\mathbb{E}[Y]$ once forward and once backward **(rcov)** $\mathbb{E}[X\,Y] = \mathrm{Cov}[X, Y] + \mathbb{E}[X]\,\mathbb{E}[Y]$ while applying Eq. (8) several times:

$$\begin{aligned} &\mathrm{Cov}[\mathbf{p}_i^n \log \mathbf{p}_i, \mathbf{p}_j^m \log \mathbf{p}_j] \qquad (35)\\ &\overset{\textbf{cov}}{=} \mathbb{E}\left[\mathbf{p}_i^n \log(\mathbf{p}_i)\, \mathbf{p}_j^m \log(\mathbf{p}_j)\right] \\ &\quad - \mathbb{E}\left[\mathbf{p}_i^n \log \mathbf{p}_i\right] \mathbb{E}\left[\mathbf{p}_j^m \log \mathbf{p}_j\right] \qquad (36)\\ &= \frac{\alpha_i^{\overline{n}}\, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n+m}}} \mathbb{E}\left[\log(\hat{\mathbf{p}}_i^{i,j}) \log(\hat{\mathbf{p}}_j^{i,j})\right] \\ &\quad - \mathbb{E}\left[\log \hat{\mathbf{p}}_i^{i}\right] \mathbb{E}\left[\log \mathbf{p}_j^{j}\right] \qquad (37)\\ &\overset{\textbf{(rcov)}}{=} \frac{\alpha_i^{\overline{n}}\, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n+m}}} \Big(\mathrm{Cov}[\log \hat{\mathbf{p}}_i^{i,j}, \log \hat{\mathbf{p}}_j^{i,j}] \\ &\qquad + \mathbb{E}\left[\log \hat{\mathbf{p}}_i^{i,j}\right] \mathbb{E}\left[\log \hat{\mathbf{p}}_j^{i,j}\right]\Big) \qquad (38)\\ &\quad - \frac{\alpha_i^{\overline{n}}\, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n}}\, \alpha_0^{\overline{m}}} \mathbb{E}\left[\log \hat{\mathbf{p}}_i^{i}\right] \mathbb{E}\left[\log \mathbf{p}_j^{j}\right], \end{aligned}$$

where $\mathbf{p}^{i,j} \sim \mathrm{Dir}(\alpha^{i,j})$ with $\alpha^{i,j} = (\ldots, \alpha_i + n, \ldots, \alpha_j + m, \ldots)$. $\mathbf{p}^{i/j}$ and $\alpha^{i/j}$ are defined analogously. Applying Eq. (7) and Eq. (6) from Lemma G.1 yields the statement. For $i = j$, the proof follows the same pattern. □

Now, we can prove the theorem that quantifies the variance of the entropy of Y:

Theorem G.6. *Given a Dirichlet distribution and a random variable* $\mathbf{p} \sim Dir(\alpha)$*, the variance of the entropy* $\mathrm{Var}_{\mathbf{p}\sim\mathrm{Dir}(\alpha)}\, \mathbb{H}_{Y\sim\mathrm{Cat}(\mathbf{p})}[Y]$ *of the categorical distribution* $Y \sim \mathrm{Cat}(\mathbf{p})$ *is given by*

$$\begin{aligned} &\mathrm{Var}[\mathbb{H}[Y \mid \mathbf{p}]] \qquad (39)\\ &= \sum_i \frac{\alpha_i^{\overline{2}}}{\alpha_0^{\overline{2}}} \left(\overline{\mathrm{Cov}[\log \hat{\mathbf{p}}_i^2, \log \hat{\mathbf{p}}_i^2]} + \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^2\right]}^2\right) \\ &\quad + \sum_{i\neq j} \frac{\alpha_i\, \alpha_j}{\alpha_0^{\overline{2}}} \Big(\overline{\mathrm{Cov}[\log \hat{\mathbf{p}}_i^1, \log \hat{\mathbf{p}}_j^1]} \\ &\qquad + \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^{1,1}\right]}\, \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_j^{1,1}\right]}\Big) \qquad (40)\\ &\quad - \sum_{i,j} \frac{\alpha_i\, \alpha_j}{\alpha_0^2} \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^{1}\right]} \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_j^{1}\right]}. \end{aligned}$$

Proof. We start by applying the well-known formula $\mathrm{Var}[\sum_i X_i] = \sum_{i,j} \mathrm{Cov}[X_i, X_j]$ and then apply Lemma G.3 repeatedly. □

Main Result

Given that we can view an ensemble member as a single deterministic model and vice versa, this provides an intuitive explanation for why single deterministic models report inconsistent and widely varying predictive entropies and confidence scores for OoD samples for which a Deep Ensemble would report high epistemic uncertainty (expected information gain) and high predictive entropy.

Assuming that $p(y|x, \omega)$ only depends on $\mathrm{p}(y \mid x)$ and $\mathbb{I}[Y; \omega \mid x]$, we model the distribution of $p(y|x, \omega)$ (as a function of ω) using a Dirichlet distribution $\mathrm{Dir}(\alpha)$ which satisfies:

$$\mathrm{p}(y \mid x) = \frac{\alpha_i}{\alpha_0} \tag{41}$$

$$\mathbb{H}[Y \mid x] - \mathbb{I}[Y; \omega \mid x] = \psi(\alpha_0 + 1) \tag{42}$$

$$- \sum_{y=1}^{K} \mathrm{p}(y \mid x) \psi(\alpha_0\ \mathrm{p}(y \mid x) + 1).. \tag{43}$$

Then, we can model the softmax distribution using a random variable $\mathbf{p} \sim \mathrm{Dir}(\alpha)$ as:

$$\mathrm{p}(y \mid x, \omega) \overset{\approx}{\sim} \mathrm{Cat}(\mathbf{p}). \tag{44}$$

The variance $\mathrm{Var}[\mathbb{H}[Y \mid x, \omega]]$ of the softmax entropy for different samples x given $\mathrm{p}(y \mid x)$ and $\mathbb{I}[Y; \omega \mid x]$ is then approximated by $\mathrm{Var}[\mathbb{H}[Y \mid \mathbf{p}]]$:

$$\mathrm{Var}_\omega[\mathbb{H}[Y \mid x, \omega]] \approx \mathrm{Var}_{\mathbf{p}}[\mathbb{H}[Y \mid \mathbf{p}]] \tag{45}$$

with the latter term given in eq. (40). We empirically find this to provide a lower bound on the true variance $\mathrm{Var}_\omega[\mathbb{H}[Y \mid x, \omega]]$.

Empirical Results

We empirically verify that softmax entropies vary considerably in Fig. 18. In Fig. 19, we verify that the predicted softmax entropy variance indeed lower-bounds the empirical softmax entropy variance. Morever, Fig. 19c shows both **i)** the non-linear relationship between epistemic uncertainty and variance in the softmax entropies and **ii)** that Dirichlet distributions cannot capture it and can only provide a lower bound. Nonetheless, this simple approximation seems to be able to capture the empirical entropy distribution quite well as shown in Fig. 20.

G.2. Objective Mismatch

In §5, we noted that the objectives that lead to optimal estimators for aleatoric and epistemic uncertainty via softmax entropy and feature-space density do not match, and DDU therefore uses the softmax layer as a discriminative classifier (implicit LDA) to estimate the predictive entropy, while it is using a GMM as generative classifier to estimate the feature-space density. Here we prove this.

G.2.1 Preliminaries

Before we prove Proposition 5.3, we will introduce some additional notation following [36].

Definition G.7. 1. $\hat{\mathrm{p}}(y, z)$ is the data distribution of the $\mathcal{D}$ in feature space with class labels y and feature representation z.
2. $\mathrm{p}_\theta(\cdot)$ is a probability distribution parameterized by θ.
3. Entropies and conditional entropies are over the empirical data distribution $\hat{\mathrm{p}}(\cdot)$:

$$\mathbb{H}[\cdot] = \mathbb{H}\left(\hat{\mathrm{p}}(\cdot)\right) = \mathbb{E}_{\hat{\mathrm{p}}(\cdot)}\left[-\log \hat{\mathrm{p}}(\cdot)\right]. \tag{46}$$

4. $\mathbb{H}[Y \mid z]$ is the entropy of $\hat{\mathrm{p}}(y \mid z)$ for a given z, whereas $\mathbb{H}[Y \mid Z]$ is the conditional entropy:

$$\mathbb{H}[Y \mid Z] = \mathbb{E}_{\hat{\mathrm{p}}(z)}\, \mathbb{H}[Y \mid z]. \tag{47}$$

5. $\mathbb{H}(\mathrm{p}(y, z) \,||\, \mathrm{q}(y|z|))$ is the cross-entropy of $\mathrm{q}(y \,|\, z)$ under $\mathrm{p}(y \mid z)$ in expectation over $\mathrm{p}(z)$:

$$\begin{aligned}\mathbb{H}(\mathrm{p}(y, z) \,||\, \mathrm{q}(y \mid z)) &= \mathbb{E}_{\mathrm{p}(z)}\, \mathbb{H}(\mathrm{p}(y \mid z) \,||\, \mathrm{q}(y \mid z)) \\ &= \mathbb{E}_{\mathrm{p}(y,z)}\left[-\log \mathrm{q}(y \mid z)\right].\end{aligned}$$

6. Similarly, $D_{\mathrm{KL}}(\mathrm{p}(y, z) \,||\, \mathrm{q}(y|z|))$ is the Kullback-Leibler divergence of $\mathrm{q}(y \,|\, z)$ under $\mathrm{p}(y \,|\, z)$ in expectation over $\mathrm{p}(z)$:

$$\begin{aligned}D_{\mathrm{KL}}(\mathrm{p}(y, z) \,||\, \mathrm{q}(y|z|)) &= \mathbb{E}_{\mathrm{p}(z)}\, D_{\mathrm{KL}}(\mathrm{p}(y \mid z) \,||\, \mathrm{q}(y|z|)) \\ &= \mathbb{H}(\mathrm{p}(y, z) \,||\, \mathrm{q}(y|z|)) - \mathbb{H}[Y \mid Z]\end{aligned}$$

7. For cross-entropies of $\mathrm{p}_\theta(\cdot)$ under $\hat{\mathrm{p}}(z, y)$, we use the convenient short-hand $\mathbb{H}_\theta[\cdot] = \mathbb{H}(\hat{\mathrm{p}}(z, y) \,||\, \mathrm{p}_\theta(\cdot))$.

Then we can observe the following connection between $\mathbb{H}_\theta[\cdot]$ and $\mathbb{H}[\cdot]$:

Lemma G.8. *Cross-entropies upper-bound the respective entropy with equality when $\mathrm{p}_\theta(\cdot) = \hat{\mathrm{p}}(\cdot)$, which is important for variational arguments:*
1. $\mathbb{H}_\theta[Y, Z] \geq \mathbb{H}[Y, Z]$,
2. $\mathbb{H}_\theta[Z] \geq \mathbb{H}[Z]$, and
3. $\mathbb{H}_\theta[Y \mid Z] \geq \mathbb{H}[Y \mid Z]$.

Proof. 1. $\mathbb{H}_\theta[Y, Z] - \mathbb{H}[Y, Z] = D_{\mathrm{KL}}(\hat{\mathrm{p}}(y, z) || \mathrm{p}_\theta(y, z)) \geq 0$.
2. follows from Item 1.
3. We expand the expectations and note that inequality commutes with expectations:

$$\mathbb{H}_\theta[Y \mid Z] - \mathbb{H}[Y \mid Z] = \mathbb{E}_{\hat{\mathrm{p}}(z)}\left[\mathbb{H}_\theta[Y \mid z] - \mathbb{H}[Y \mid z]\right] \geq 0,$$

because $\mathbb{H}_\theta[Y \mid z] - \mathbb{H}[Y \mid z] \geq 0$ for all z. The equality conditions follows from the properties of the Kullback-Leibler divergence as well.

□

We also have:

Lemma G.9.

$$\begin{aligned}\mathbb{H}_\theta[Y, Z] &= \mathbb{H}_\theta[Y \mid Z] + \mathbb{H}_\theta[Z] && (48)\\ &= \mathbb{H}_\theta[Z \mid Y] + \mathbb{H}_\theta[Y]. && (49)\end{aligned}$$

Proof. We substitute the definitions and obtain:

$$\begin{aligned}\mathbb{H}_\theta[Y, Z] &= \mathbb{E}_{\mathrm{p}(y,z)}\left[-\log \mathrm{q}(y, z)\right] && (50)\\ &= \mathbb{E}_{\mathrm{p}(y,z)}\left[-\log \mathrm{q}(y \mid z)\right] + \mathbb{E}_{\mathrm{p}(y,z)}\left[-\log \mathrm{q}(z)\right] && (51)\\ &= \mathbb{H}_\theta[Y \mid Z] + \mathbb{H}_\theta[Z]. && (52)\end{aligned}$$

□

The same holds for entropies: $\mathbb{H}[Y, Z] = \mathbb{H}[Y \mid Z] + \mathbb{H}[Z] = \mathbb{H}[Y \mid Z] + \mathbb{H}[Y]$ [5].

G.2.2 Proof

We can now prove the observation.

Proposition 5.3. *For an input x, let $z = f_\theta(x)$ denote its feature representation in a feature extractor f_θ with parameters θ. Then the following hold:*
1. A discriminative classifier $\mathrm{p}(y \mid z)$, e.g. a softmax layer, is well-calibrated in its predictions when it maximises the conditional log-likelihood $\log \mathrm{p}(y \mid z)$;
2. A feature-space density estimator $\mathrm{q}(z)$ is optimal when it maximises the marginalised log-likelihood $\log \mathrm{q}(z)$;

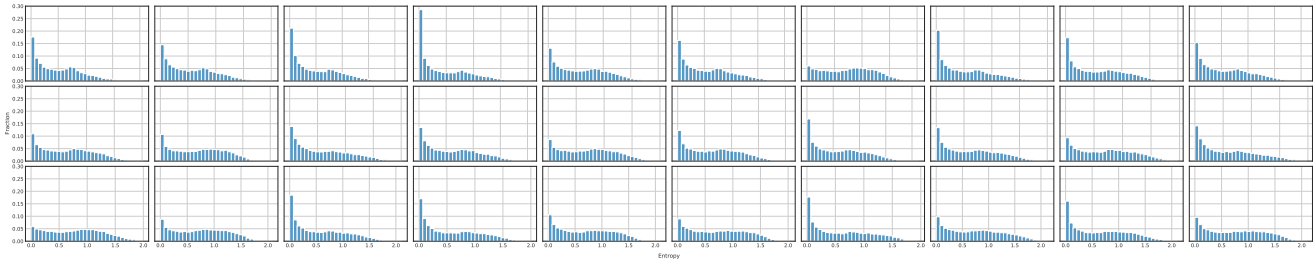
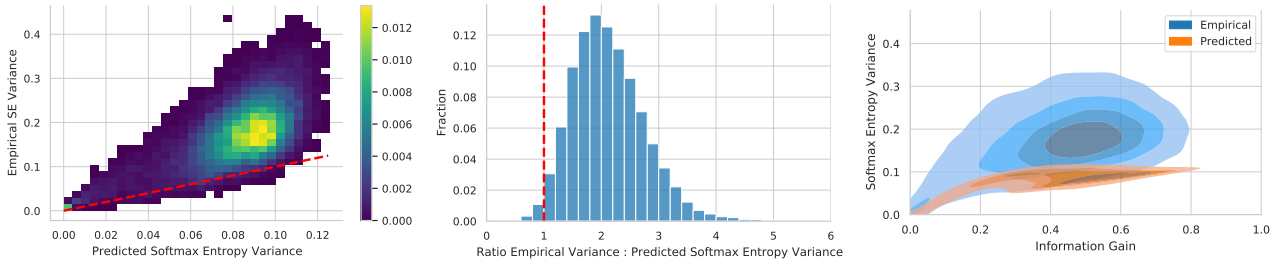
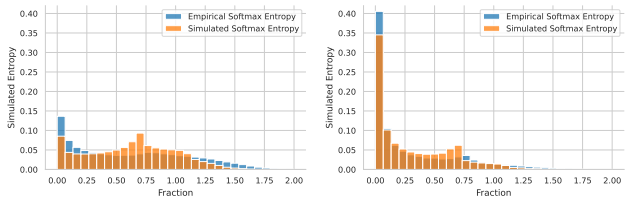


Figure 18. *Softmax entropy histograms of 30* Wide-ResNet-28-10+SN *models trained on CIFAR-10, evaluated on SVHN (OoD).* The softmax entropy distribution of the different models varies considerably.



(a) Simulated vs Ground-Truth (b) Ratio Histogram (c) Density Plot

Figure 19. *The variance of softmax entropies can be lower-bounded by fitting Dirichlet distributions on the samples* $\mathrm{p}(y \mid x, \omega)$. **(a)** The empirical variance of softmax entropies is lower-bounded by $\mathrm{Var}[\mathbb{H}[Y \mid \mathbf{p}]]$. The red dashed line depicts equality. **(b)** The ratio histogram shows that there are only few violations due to precision issues ($< 2\%$).**(c)** The variance of the softmax entropy is not linearly correlated to the epistemic uncertainty. For both high and low epistemic uncertainty, the variance decreases.



(a) WideResNet-28-10+SN (b) VGG16

Figure 20. *Simulated vs empirical softmax entropy on WideResNet-28-10+SN and VGG16.* Even though the Dirichlet variance approximation lower-bounds the empirical softmax entropy variance, sampling from the fitted Dirichlet distributions does approximate the empirical entropy distribution quite well.

3. *A mixture model* $\mathrm{q}(y, z) = \sum_y \mathrm{q}(z \mid y)\,\mathrm{q}(y)$ *might not maximise both objectives, conditional log-likelihood and marginalised log-likelihood, at the same time. In the specific instance that a GMM with one component per class does maximise both, the resulting model must be a GDA (but the opposite does not hold).*

Proof. 1. The conditional log-likelihood is a strictly proper scoring rule [17]. The optimization objective can be rewritten as

$$\max_\theta \mathbb{E}_{\log \mathrm{p}_\theta(y|z)} = \min_\theta \mathbb{H}_\theta[Y \mid Z] \geq \mathbb{H}[Y \mid Z]. \tag{53}$$

An optimal discriminative classifier $\mathrm{p}_\theta(y \mid z)$ would thus capture the true (empirical) distribution everywhere: $\mathrm{p}_\theta(y \mid z) = \hat{\mathrm{p}}(y \mid z)$. This means the negative conditional log-likelihood will be equal $\mathbb{H}[Y \mid Z]$ and $\mathbb{H}_\theta[Y \mid z] = \mathbb{H}[Y \mid z]$ for all z.

2. For density estimation $\mathrm{q}(z)$, the maximum likelihood $\mathbb{E}\left[\log \mathrm{q}(z)\right]$ using the empirical data distribution is maximized. We can rewrite this as

$$\max_\theta \mathbb{E}_{\hat{\mathrm{p}}(y,z)} \log \mathrm{p}_\theta(z) = \min_\theta \mathbb{H}_\theta[Z] \geq \mathbb{H}[Z]. \tag{54}$$

We see that the negative marginalized likelihood of the density estimator upper-bounds the entropy of the feature representations $\mathbb{H}[Z]$. We have equality and $\mathrm{p}_\theta(z) = \hat{\mathrm{p}}(z)$ in the optimum case.

3. Using $\mathbb{H}_\theta[Y, Z] = \mathbb{H}_\theta[Y \mid Z] + \mathbb{H}_\theta[Z]$, we can relate the objectives from Eq. (53) and (54) to each other. First, we characterize a shared optimum, and then we show that both objectives are generally not minimized at the same time. For both objectives to be minimized, we have $\nabla\, \mathbb{H}_\theta[Y \mid Z] = 0$ and $\nabla\, \mathbb{H}_\theta[Z] = 0$, and we obtain

$$\nabla\, \mathbb{H}_\theta[Y, Z] = \nabla\, \mathbb{H}_\theta[Y \mid Z] + \nabla\, \mathbb{H}_\theta[Z] = 0. \tag{55}$$

From this we conclude that minimizing both objectives also minimizes $\mathbb{H}_\theta[Y, Z]$, and that generally the objectives trade-

off with each other at stationary points θ of $\mathbb{H}_\theta[Y, Z]$:

$$\nabla \mathbb{H}_\theta[Y \mid Z] = -\nabla \mathbb{H}_\theta[Z] \quad \text{when } \nabla \mathbb{H}_\theta[Y, Z] = 0. \tag{56}$$

This tells us that to construct a case where the optima do not coincide, discriminative classification needs to be opposed better density estimation.

Specifically, when we have a GMM with one component per class, minimizing $\mathbb{H}_\theta[Y, Z]$ on an empirical data distribution is equivalent to Gaussian Discriminant Analysis, as is easy to check, and minimizing $\mathbb{H}_\theta[Z]$ is equivalent to fitting a density estimator, following Eq. (54). The difference is that using a GMM as a density estimator does not constrain the component assignment, unlike in GDA.

Consequently, we see that *both objectives can be minimized at the same time exactly when the feature representations of different classes are perfectly separated*, such that a GMM fit as density estimator would assign each class's feature representations to a single component.

By the above, we can construct a simple case: if the samples of different classes are not separated in feature-space, optimas for the objectives will not coincide, so for example if samples were drawn from the same Gaussian and labeled randomly. On the other hand, if we have classes whose features lie in well-separated clusters, GDA will minimize all objectives.

□

Given that perfect separation is impossible with ambiguous data for a GMM, a shared optimum will be rare with noisy real-world data, but only then would GDA be optimal. In all other cases, GDA does not optimize both objectives, and neither can any other GMM with one component per class. Moreover, Eq. (56) shows that a GMM fit using EM is a better density estimator than GDA, and a softmax layer is a better classifier, as optimizing the softmax objective $\mathbb{H}_\theta[Y \mid Z]$ or density objective $\mathbb{H}[Z]$ using gradient descent will move away from the GDA optimum.

As can easily be verified, a trivial optimal minimizer $\mathrm{q}^*(y, z)$ for $\mathbb{H}_\theta[Y, Z]$ given an empirical data distribution $\hat{\mathrm{p}}(y, z)$ is an adapted Parzen estimator:

$$\mathrm{q}^*(y, z) = \sum_y \hat{\mathrm{p}}(y)\, \mathbb{E}_{\hat{z} \sim \hat{\mathrm{p}}(z \mid y)}\, \mathcal{N}(z;\, \hat{z},\, \sigma^2 \mathbf{I}), \tag{57}$$

for small enough σ. This shows that above proposition is not general.

G.2.3 Intuitions & Validation with a Toy Example

Figure 17 visualises this on a synthetic 2D dataset with three classes and 4% label noise, which causes the optima to diverge as described in the proof. Label noise is a common issue in real-world datasets. Non-separability even more so. To explain Proposition 5.3 in an intuitive way, we focus on on a simple 2D toy case and fit a GMM using the different objectives. We sample "latents" z from 3 Gaussians (each representing a different class y) with 4% label noise. Following the construction in the proof, this will lead the objectives to have different optima.

We know discuss the different objectives in Fig. 17 and the resulting scores in more detail:

$\mathbf{min}\, \mathbb{H}_{\boldsymbol{\theta}}[\boldsymbol{Y} \mid \boldsymbol{Z}]$. A softmax linear layer is equivalent to an LDA (Linear Discriminant Analysis) with conditional likelihood as detailed in [54], for example. We optimize an LDA with the usual objective "$\min -1/N \sum \log \mathrm{p}(y \mid z)$", i.e. the cross-entropy of $\mathrm{p}(y \mid z)$ or (average) negative log-likelihood (NLL). Following Definition G.7, we use the short-hand "$\min \mathbb{H}_\theta[Y \mid Z]$" for this cross-entropy.

Because we optimize only $\mathrm{p}(y|z)$, $\mathrm{p}(z)$ does not affect the objective and is thus not optimized. Indeed, the components do not actually cover the latents well, as can be seen in the first density plot of Fig. 17a. However, it does provide the lowest NLL.

$\mathbf{min}\, \mathbb{H}_{\boldsymbol{\theta}}[\boldsymbol{Y}, \boldsymbol{Z}]$. We optimize a GDA for the combined objective "$\min -1/N \sum \log \mathrm{q}(y, z)$", i.e. the cross-entropy of $\mathrm{q}(y, z)$. We use the short-hand "$\min \mathbb{H}_\theta[Y \mid Z]$" for this.

$\mathbf{min}\, \mathbb{H}_{\boldsymbol{\theta}}[\boldsymbol{Z}]$. We optimize a GMM for the objective "$\min -1/N \sum \log \mathrm{q}(z)$", i.e. the cross-entropy of $\mathrm{q}(z)$. We use the short-hand "$\min \mathbb{H}_\theta[Z]$" for this.

We do not provide scores for $\mathbb{H}_\theta[Y \mid Z]$ and $\mathbb{H}_\theta[Y, Z]$ for the third objective $\min \mathbb{H}_\theta[Z]$ in Tab. 17 as it does not depend on Y, and hence the different components do not actually model the different classes necessarily. Hence, we also use a single color to visualize the components for this objective in Fig. 17a.

In Tab. 17 and Fig. 17a, we see that each solution minimizes its own objective best. The GMM provides the best density model (best fit according to the entropy), while the LDA (like a softmax linear layer) provides the best NLL for the labels. The GDA provides a density model that is almost as good.

Entropy. Looking at the entropy plots in Fig. 17b, we first notice that the LDA solution optimized for $\min \mathbb{H}_\theta[Y \mid Z]$ has a wide decision boundary. This is due to the overlap

Table 17. *Realized objective scores (columns) for different optimization objectives (rows) for the synthetic 2D toy example depicted in Fig. 17.* Smaller is better. We see that each objectives minimizes its own score while being suboptimal in regards to the other two objectives (when it is possible to compute the scores). This empirically further validates Proposition 5.3.

Objective	$\mathbb{H}_{\boldsymbol{\theta}}[\boldsymbol{Y} \mid \boldsymbol{Z}]$ (↓)	$\mathbb{H}_{\boldsymbol{\theta}}[\boldsymbol{Y}, \boldsymbol{Z}]$ (↓)	$\mathbb{H}_{\boldsymbol{\theta}}[\boldsymbol{Z}]$ (↓)
$\min \mathbb{H}_\theta[Y \mid Z]$	0.1794	5.4924	5.2995
$\min \mathbb{H}_\theta[Y, Z]$	0.2165	4.9744	4.7580
$\min \mathbb{H}_\theta[Z]$	n/a	n/a	4.7073

of the Gaussian components, which is necessary to provide the right aleatoric uncertainty.

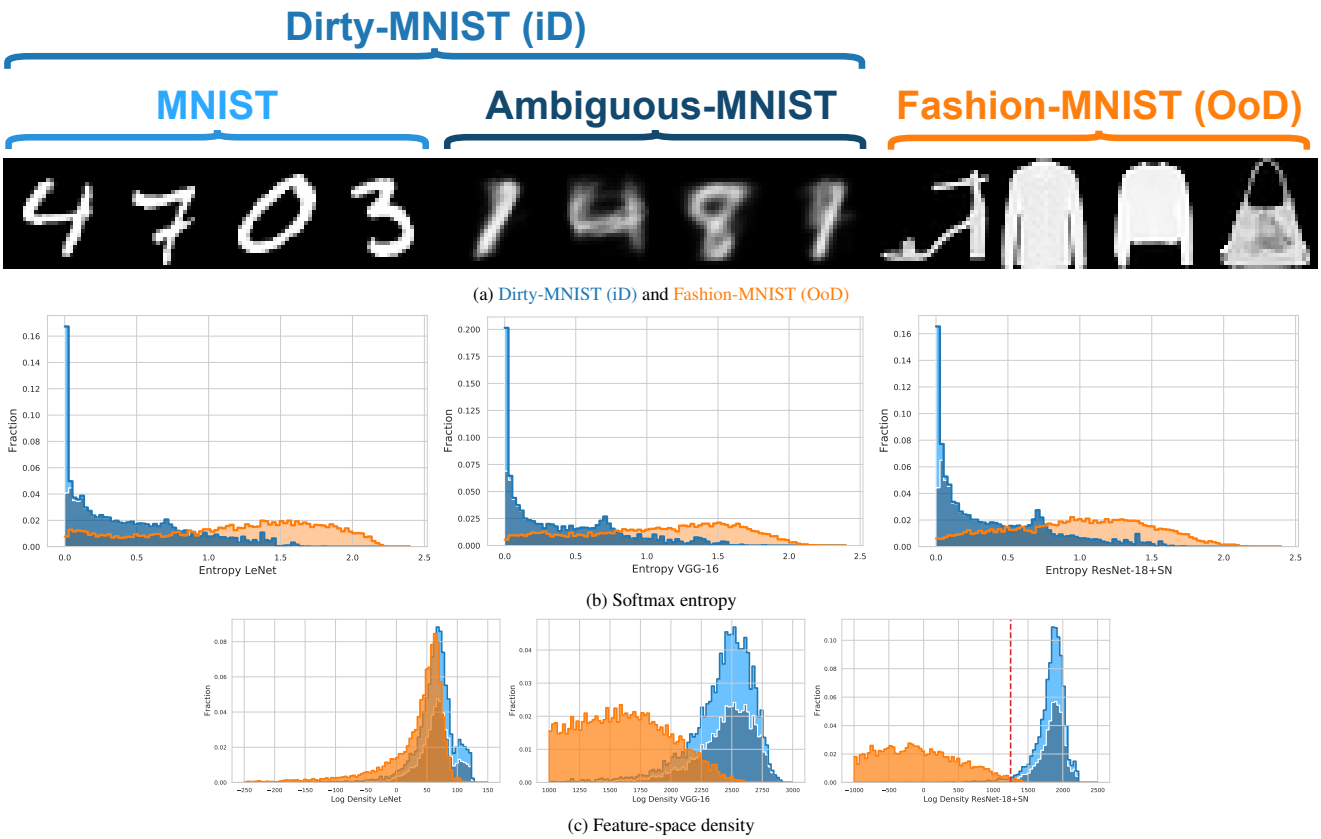
Optimizing the negative log-likehood $-\log \mathrm{p}(y \mid z)$ is a proper scoring rule, and hence is optimized for calibrated predictions.

Compared to this, the GDA solution (optimized for $\min \mathbb{H}_\theta[Y, Z]$ has a much narrower decision boundary and cannot capture aleatoric uncertainty as well. This is reflected in the higher NLL. Moreover, unlike for LDA, GDA decision boundaries behave differently than one would naively expect due to the untied covariance matrices. They can be curved and the decisions change far away from the data [54].

To show the difference between the two objectives we have marked an ambiguous point near $(0, -5)$ with a yellow star . Under the first objective $\min \mathbb{H}_\theta[Y, Z]$, the point has high aleatoric uncertainty (high entropy), as seen in the left entropy plot while under the second objective ($\min \mathbb{H}_\theta[Y, Z]$) the point is only assigned very low entropy. The GDA optimized for the second objective thus is overconfident.

As above explained above, we do not show an entropy plot of $Y \mid Z$ for the third objective $\min \mathbb{H}_\theta[Z]$ in Fig. 17b because the objective does not depend on Y, and there are thus no class predictions.

Intuitively, for aleatoric uncertainty, the Gaussian components need to overlap to express high aleatoric uncertainty (uncertain labelling). At the same time, this necessarily provides looser density estimates. On the other hand, the GDA density is much tighter, but this comes at the cost of NLL for classification because it cannot express aleatoric uncertainty that well. Figure 17 visualizes how the objectives tradeoff between each other, and why we use the softmax layer trained for $\mathrm{p}(y \mid z)$ for classification and aleatoric uncertainty, and GDA as density model for $\mathrm{q}(z)$.



(a) Dirty-MNIST (iD) and Fashion-MNIST (OoD)

(b) Softmax entropy

(c) Feature-space density

Figure 21. *Disentangling aleatoric and epistemic uncertainty on Dirty-MNIST (iD) and Fashion-MNIST (OoD) **(a)** requires using* softmax entropy *__(b)__ and* feature-space density (GMM) *__(c)__ with appropriate inductive biases (*ResNet-18+SN *vs* LeNet *&* VGG-16 *without them). Enlarged version.* **(b):** Softmax entropy captures aleatoric uncertainty for iD data (Dirty-MNIST), thereby separating unambiguous MNIST samples and Ambiguous-MNIST samples. However, iD and OoD are confounded: softmax entropy has arbitrary values for OoD, indistinguishable from iD. **(c):** With appropriate inductive biases (DDU with ResNet-18+SN), iD and OoD densities do not overlap, capturing epistemic uncertainty. However, without appropriate inductive biases (LeNet & VGG-16), feature density suffers from *feature collapse*: iD and OoD densities overlap. Generally, feature-space density confounds unambiguous and ambiguous iD samples as their densities overlap. **Note:** Unambiguous MNIST samples and Ambiguous-MNIST samples are shown as stacked histograms with the total fractions adding up to 1 for Dirty-MNIST.