Deterministic Neural Networks with Inductive Biases Capture Epistemic and Aleatoric Uncertainty

Jishnu Mukhoti [*] [1] [2] Andreas Kirsch [*] [1] Joost van Amersfoort [1] Philip H.S. Torr [2] Yarin Gal [1]

Abstract

We show that a single softmax neural net with minimal changes can beat the uncertainty predictions of Deep Ensembles and other more complex single-forward-pass uncertainty approaches. Standard softmax neural nets suffer from feature collapse and extrapolate arbitrarily for OoD points. This results in arbitrary softmax entropies for OoD points which can have high entropy, low, or anything in between, thus cannot capture epistemic uncertainty reliably. We prove that this failure lies at the core of "why" Deep Ensemble Uncertainty works well. Instead of using softmax entropy, we show that with appropriate inductive biases softmax neural nets trained with maximum likelihood reliably capture epistemic uncertainty through their feature-space density. This density is obtained using simple Gaussian Discriminant Analysis, but it cannot represent aleatoric uncertainty reliably. We show that it is necessary to combine feature-space density with softmax entropy to disentangle uncertainties well. We evaluate the epistemic uncertainty quality on active learning and OoD detection, achieving SOTA $\sim$ 98 AUROC on CIFAR-10 vs SVHN without fine-tuning on OoD data.

1. Introduction

Two types of uncertainty are often of interest in ML: *epistemic uncertainty*, which is inherent to the model, caused by a lack of training data, and hence reducible with more data[1], and *aleatoric uncertainty*, caused by inherent noise or ambiguity in data, and hence irreducible with more data (Der Kiureghian & Ditlevsen, 2009; Kendall & Gal, 2017). Disentangling these two and reasoning about each one independently is critical for applications such as active learning (Gal et al., 2017) or detection of out-of-distribution (OoD) samples (Hendrycks & Gimpel, 2016): in active learning, we wish to avoid inputs with high aleatoric but low epistemic uncertainty, and in OoD detection, we wish to avoid mistaking ambiguous in-distribution (iD) examples as OoD. This is particularly challenging for noisy and ambiguous datasets found in safety-critical applications like autonomous driving (Huang & Chen, 2020) and medical diagnosis (Esteva et al., 2017; Filos et al., 2019).

Most well-known methods of uncertainty quantification in deep learning (Blundell et al., 2015; Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017; Wen et al., 2019; Dusenberry et al., 2020) require multiple forward passes at test time. Amongst these, Deep Ensembles (Lakshminarayanan et al., 2017) have generally performed best in uncertainty prediction (Ovadia et al., 2019), but their significant memory and compute burden at training and test time hinders their adoption in real-life and mobile applications. Consequently, there has been an increased interest in uncertainty quantification using deterministic single forward-pass neural networks which have a smaller footprint and lower latency.

Two recent works in single forward-pass uncertainty, DUQ (van Amersfoort et al., 2020) and SNGP (Liu et al., 2020a), propose distance-aware output layers, in the form of RBFs (radial basis functions) or GPs (Gaussian processes), and introduce additional inductive biases in the feature extractor using a Jacobian penalty (Gulrajani et al., 2017) or spectral normalisation (Miyato et al., 2018), respectively. These methods perform well and are competitive with Deep Ensembles on common uncertainty benchmarks. However, they require training to be changed substantially, use additional hyper-parameters due to the specialised output layers used at training, and in the case of DUQ, do not differentiate between epistemic and aleatoric uncertainty, all of which hinders adoption.

In this paper we introduce a new model class, *Deep Deterministic Uncertainty (DDU)*, and show that by enforcing *smoothness* and *sensitivity* on the feature-space as inductive

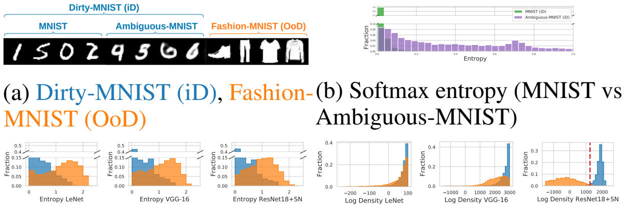
[*] Equal contribution [1] OATML, University of Oxford [2] Torr Vision Group, University of Oxford. Correspondence to: Jishnu Mukhoti <jishnu.mukhoti@eng.ox.ac.uk>, Andreas Kirsch <andreas.kirsch@cs.ox.ac.uk>.

Presented at the ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning., Copyright 2021 by the author(s).

[1] We follow the definition of epistemic uncertainty at input x as a quantity which is high for a previously unseen x, and decreases when x's label is added to the training set and the model is updated.

biases, *a single softmax neural network trained with maximum likelihood can reliably capture epistemic uncertainty through its feature-space density*, with no adaptations required for its training procedure. While sensitivity prevents feature collapse, which happens when feature extractors map OoD samples to iD regions in feature space (van Amersfoort et al., 2020), smoothness ensures that feature-space distances are meaningful (Liu et al., 2020a; Rosca et al., 2020), and as we show below, this is closely tied to feature-space density when using a distance-based density estimator. Given a *pre-trained* softmax neural network with such inductive biases, we employ Gaussian Discriminant Analysis (GDA), which fits a Gaussian Mixture Model (GMM) with one Gaussian per class on the feature space of the pre-trained network (Murphy, 2012), and use its density to estimate epistemic uncertainty. Our sensitivity and smoothness constraints are crucial, as without these the feature-space density alone might not separate in-distribution from OoD data, possibly explaining the limited empirical success of previous approaches which attempt to model feature-space density with no constraints over the feature space (Lee et al., 2018b; Postels et al., 2020). This can be clearly seen in Figure 1(d), further discussed below. Finally, by combining information from both the feature-space density and the softmax entropy, we can correctly disentangle epistemic and aleatoric uncertainty. We show that this is necessary since a single output layer (either softmax or GDA) cannot generally be optimal for capturing both epistemic and aleatoric uncertainty at the same time.

In Figure 1 we train a LeNet (LeCun et al., 1998), a VGG-16 (Simonyan & Zisserman, 2015) and a model with our inductive biases (a ResNet-18 with spectral normalisation, ResNet+SN (He et al., 2016; Miyato et al., 2018)) on *Dirty-MNIST*, a modified version of MNIST (LeCun et al., 1998) with additional ambiguous digits (Ambiguous-MNIST). *Ambiguous-MNIST* contains samples with multiple plausible labels and thus higher aleatoric uncertainty (see Figure 1(a)). We refer to Appendix D 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 (Filos et al., 2019; Krizhevsky et al., 2009). At the same time, Dirty-MNIST poses a challenge for recent uncertainty estimation methods, which often confound aleatoric and epistemic uncertainty. Figure 1(c) shows that the softmax entropy of a deterministic model is unable to distinguish between iD (Dirty-MNIST) and OoD (Fashion-MNIST (Xiao et al., 2017)) 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 Figure 1(d) captures epistemic uncertainty reliably and is able to distinguish iD from OoD samples. The same cannot be said for LeNet or VGG in Figure 1(d), 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, Figure 1(b) and Figure 1(d) 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 Figure 1(b).



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

(b) Softmax entropy (MNIST vs Ambiguous-MNIST)

(c) Softmax entropy

(d) Feature space density

Figure 1: *OoD separation for neural networks without and with appropriate inductive biases (*LeNet & VGG *vs* ResNet+SN *resp.), using* softmax entropy (left) *and* feature-space density (GMM, right). All trained on *Dirty-MNIST as in-distribution (iD)* and evaluated on *Fashion-MNIST as OoD*, see **(a)**. **(b)**: Softmax entropy of a model with appropriate inductive biases (ResNet+SN) trained on Dirty-MNIST is able to capture aleatoric uncertainty correctly. It separates MNIST from Ambiguous-MNIST, both iD in this case. **(c):** Softmax entropy has arbitrary values for OoD, indistinguishable from iD, thus cannot separate OoD from iD. **(d):** Without appropriate inductive bias (LeNet & VGG), feature density suffers from feature collapse, leading to OoD density overlapping with iD. However, with appropriate inductive biases (DDU with ResNet+SN), the densities do not overlap.

Our contributions are as follows. In Section 2, we show that **i)** we cannot estimate epistemic uncertainty from a deterministic softmax model's softmax entropy (and in fact, this is tied to the very reason why Deep Ensembles capture uncertainty well), hence we must resort to alternatives such as feature-space density; **ii)** feature collapse leads to a failure in capturing epistemic uncertainty via feature-space density of a deterministic softmax model, and enforcing smoothness and sensitivity on the feature space alleviates this problem; **iii)** that we need both softmax entropy and feature density to disentangle epistemic uncertainty from aleatoric uncertainty; and **iv)** we must use separate output layers for aleatoric and epistemic uncertainty because a single output layer cannot generally be optimal for both. In Section 3, we detail how to implement our approach, DDU, with no change to training procedure and minimal changes to softmax models' architecture. Finally, in Appendix C, we validate on relevant tasks like OoD detection and active learning that with DDU, using a single deterministic model with appropriate inductive biases, we are able to outperform Deep Ensemble uncertainty and other deterministic uncertainty baselines while maintaining high accuracy.

2. Uncertainty in Deterministic Softmax Models

In Appendix A and Appendix B, we provide a detailed description of related work and background concepts for this paper. In this section, we motivate DDU step by step. Additional formalisations and proofs for all statements in this section are provided in the Appendix I.

Softmax entropy cannot capture epistemic uncertainty because Deep Ensembles can. As mentioned in Appendix B, Equation (1) can be used with Deep Ensembles, as each ensemble member can be considered a sample from *some* distribution $\mathrm{p}(\omega \mid \mathcal{D})$ over model parameters $\omega \subset \Omega$ (e.g. a uniform distribution over K trained ensemble members $\omega_1, ..., \omega_K$). Note that the mutual information $\mathbb{I}[Y; \omega \mid x, \mathcal{D}]$ isolates epistemic from aleatoric uncertainty for Deep Ensembles as well, whereas the predictive entropy $\mathbb{H}[Y \mid x, \mathcal{D}]$ (often used with Deep Ensembles) measures predictive uncertainty, which will be high whenever either epistemic or aleatoric uncertainties are high. Furthermore, the mechanism underlying Deep Ensemble uncertainty that pushes epistemic uncertainty to be high on OoD data is the function disagreement between different ensemble components, i.e. arbitrary extrapolations of the softmax models composing the ensemble (leading the “aleatoric” term in Equation (1) to vanish (Smith & Gal, 2018)). From this and Equation (1), we can draw the following conclusion:

Proposition 2.1. *Let x_1 and x_2 be points such that x_1 has **higher** epistemic uncertainty than x_2 under the ensemble: $\mathbb{I}[Y_1; \omega \mid x_1, \mathcal{D}] > \mathbb{I}[Y_2; \omega \mid x_2, \mathcal{D}] + \delta$, with $\delta \geq 0$. Further assume both have similar predictive entropy $|\,\mathbb{H}[Y_1 | x_1, \mathcal{D}] - \mathbb{H}[Y_2 \mid x_2, \mathcal{D}]| \leq \epsilon$, for $\epsilon \geq 0$. 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: $\mathbb{H}[Y_1 \mid x_1, \omega] < \mathbb{H}[Y_2 \mid x_2, \omega] - (\delta - \epsilon)$.*

This shows that if a sample is assigned higher epistemic uncertainty (in the form of mutual information) by a Deep Ensemble, it will necessarily be assigned lower softmax entropy by at least one of the ensemble's members. As a result, *the empirical observation that the mutual information of an ensemble can quantify epistemic uncertainty well implies that the softmax entropy of a deterministic model cannot.* This claim is further supported by Figure 1(c) (and Appendix H and I.1.3) where we observe the softmax entropy for OoD samples to have values which can be high, low or anywhere in between. This failure of softmax entropy motivates us to study feature density as an alternative.

Capturing epistemic uncertainty in feature space requires sensitivity and smoothness. As discussed in Appendix B, feature extractors without sensitivity and smoothness constraints can suffer from feature collapse: they might map OoD samples to iD regions of the feature space. van Amersfoort et al. (2020) argue that sensitivity and smoothness are necessary for capturing epistemic uncertainty and show this empirically. Liu et al. (2020a) also show that a bi-Lipschitz constraint ensures both smoothness and sensitivity. This motivates our choice to focus on bi-Lipschitz constraints for DDU. The effects of these inductive biases on feature collapse are visible in Figure 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[2]. We see no overlapping densities in Figure 1(c, right) when we use an appropriate inductive bias, therefore we indeed have no feature collapse.

Disentangling epistemic and aleatoric uncertainty is not trivial. A feature-density estimator fit to iD data cannot generally estimate aleatoric uncertainty well: an iD sample ought to have a high density regardless of whether it is ambiguous (with high aleatoric uncertainty) or not, as is the case with Dirty-MNIST (see Section 1). Likewise, softmax entropy cannot capture epistemic uncertainty (see Figure 1(c) where entropy cannot distinguish OoD from ambiguous iD samples).

Observation 2.2. *Neither the softmax entropy of a deterministic model, nor the feature density alone, even with sensitivity and smoothness, can be used to disentangle epistemic from aleatoric uncertainty.*

Furthermore, as we prove next, the predictive probability induced by the feature-density estimator will generally not be well-calibrated as there is an objective mismatch. This was overlooked in previous research on uncertainty quantification for deterministic models (Lee et al., 2018b; Liu et al., 2020a; van Amersfoort et al., 2020; He et al., 2016) as they do not consider ambiguous samples which can be found in many real world applications.

Capturing aleatoric and epistemic uncertainty requires multiple generative classifiers. A single output layer, whether a softmax layer or GMM, cannot be optimal for both feature-space density and predictive distribution estimation as there is an *objective mismatch*:

Proposition 2.3. *For an input x, let $z = f_\theta(x)$ denote its*

[2] 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.

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* $\log \mathrm{q}(z)$*;*
3. *A mixture model* $\mathrm{q}(y, z)$ *cannot generally maximise both objectives, conditional log-likelihood and marginalised log-likelihood, at the same time. In the specific instance that it does maximise both, the resulting model must be a GDA (but the opposite does not hold).*

Note that if the two objectives coincide, then the resulting model must be a GDA. But the opposite is not true: given a GDA, the two objectives might not coincide. Hence, we use both a discriminative classifier and a feature-density estimator on a model trained using conditional log-likelihood (cross-entropy objective): we fit a density estimator on the feature representations of the training data to capture epistemic uncertainty, and use softmax entropy to capture aleatoric uncertainty.

From this, we conclude that *we can use softmax layer to capture aleatoric uncertainty for iD samples and a feature-space density estimator to capture epistemic uncertainty.* This combination allows us to disentangle aleatoric and epistemic uncertainty in deterministic models.

3. Algorithm

Using the insights developed above, we propose a deterministic neural network with appropriate inductive biases to satisfy our conditions. Additionally, we show how to disentangle its uncertainties, which enables us to outperform Deep Ensemble Uncertainty on standard uncertainty benchmarks.

Ensuring sensitivity & smoothness: We introduce inductive biases into the model to learn representations that avoid feature collapse. Specifically, we encourage bi-Lipschitzness by using residual connections together with spectral normalisation following (Liu et al., 2020a), who found empirically that this leads to the avoidance of feature collapse, which is further supported by our Figure 1 and experiments below. Comparing to the Jacobian gradient penalty used in (van Amersfoort et al., 2020), spectral normalisation is significantly faster and has more stable training. Additionally, using gradient penalty with residual connection leads to some difficulties, as discussed in (Liu et al., 2020a). We make minor changes to the standard ResNet model architecture to further encourage bi-Lipschitzness without sacrificing accuracy (detailed in Appendix E). Unlike other approaches (van Amersfoort et al., 2020; Liu et al., 2020a), DDU is trained using a standard softmax output layer with a maximum likelihood (cross-entropy) objective.

Disentangling epistemic & aleatoric uncertainty: To quantify epistemic uncertainty, a feature-space density model is fitted after training the softmax model. For this, DDU uses GDA: a GMM $q(y, z)$ with a single Gaussian mixture component per class. We fit each class component of the GDA by computing the empirical mean and covariance, per class, of the feature vectors $z = f_\theta(x)$, which are the outputs of the last convolutional layer of the model computed on the training samples x (*note that we do not require OoD data to fit these*). Unlike fitting a GMM using the expectation maximization algorithm, this requires just a single pass through the training set given a trained model. Moreover, as the softmax layer as an implicit GMM leads to clustering (Hess et al., 2020), we found that a single Gaussian per class is sufficient for state-of-the-art performance. At test time, we compute the epistemic uncertainty by evaluating the marginal likelihood of the feature representation under our density $q(z) = \sum_y q(z|y)q(y)$. To quantify aleatoric uncertainty for in-distribution samples, we use the entropy $\mathbb{H}[Y|x, \theta]$ of the softmax distribution $p(y|x, \theta)$ following proposition 2.3. Note that the softmax distribution thus obtained, can be further calibrated using temperature scaling (Guo et al., 2017). Finally, we disentangle aleatoric and epistemic uncertainty in DDU using the following steps:

1. For a given input, a high feature-space density indicates low epistemic uncertainty, i.e. the input is iD, and we can trust the aleatoric uncertainty estimated by the softmax entropy It can either be unambiguous (low softmax entropy) or ambiguous (high softmax entropy)
2. A low feature-space density indicates high epistemic uncertainty, i.e., the input is OoD, and the softmax entropy for the input can take any arbitrary value.

In Appendix C, we provide detailed experimental results to study DDU's performance in disentangling epistemic and aleatoric uncertainty, in Active Learning using both clean and Dirty-MNIST and on OoD detection using CIFAR-10 vs SVHN/CIFAR-100 and CIFAR-100 vs SVHN datasets.

4. Conclusion

Deep Deterministic Uncertainty (DDU) can outperform the state-of-the-art on uncertainty quantification including Deep Ensembles on active learning and OoD detection. We have shown that, with appropriate inductive biases, a neural network with a regular off-the-shelf architecture can quantify epistemic uncertainty through its feature-space density and can also disentangle aleatoric from epistemic uncertainty using its softmax entropy. Hence, DDU provides a very simple method to produce excellent epistemic and aleatoric uncertainty estimates without requiring the complexities or computational cost of the current state-of-the-art.

References

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pp. 1613–1622. PMLR, 2015.

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.

Cover, T. M. *Elements of information theory*. John Wiley & Sons, 1999.

Der Kiureghian, A. and Ditlevsen, O. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

DeVries, T. and Taylor, G. W. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

Dusenberry, M., Jerfel, G., Wen, Y., Ma, Y., Snoek, J., Heller, K., Lakshminarayanan, B., and Tran, D. Efficient and scalable bayesian neural nets with rank-1 factors. In *International conference on machine learning*, pp. 2782–2792. PMLR, 2020.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

Filos, A., Farquhar, S., Gomez, A. N., Rudner, T. G., Kenton, Z., Smith, L., Alizadeh, M., de Kroon, A., and Gal, Y. A systematic comparison of bayesian deep learning robustness in diabetic retinopathy tasks. *arXiv preprint arXiv:1912.10481*, 2019.

Gal, Y. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Gal, Y., Islam, R., and Ghahramani, Z. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pp. 1183–1192. PMLR, 2017.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Gouk, H., Frank, E., Pfahringer, B., and Cree, M. J. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *NeurIPS*, 2017.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*, 2017.

He, B., Lakshminarayanan, B., and Teh, Y. W. Bayesian Deep Ensembles via the Neural Tangent Kernel. In *Advances in neural information processing systems*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Hendrycks, D., Mazeika, M., and Dietterich, T. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2018.

Hernández-Lobato, J. M. and Adams, R. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, pp. 1861–1869. PMLR, 2015.

Hess, S., Duivesteijn, W., and Mocanu, D. Softmax-based classification is k-means clustering: Formal proof, consequences for adversarial attacks, and improvement through centroid based tailoring. *arXiv preprint arXiv:2001.01987*, 2020.

Hinton, G. E. and Van Camp, D. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pp. 5–13, 1993.

Huang, Y. and Chen, Y. Autonomous driving with deep learning: A survey of state-of-art technologies. *arXiv preprint arXiv:2006.06091*, 2020.

Kendall, A. and Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pp. 5574–5584, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *ICLR*, 2014.

Kirsch, A., Lyle, C., and Gal, Y. Unpacking information bottlenecks: Unifying information-theoretic objectives in deep learning. *arXiv preprint arXiv:2003.12537*, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pp. 6402–6413, 2017.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lee, K., Lee, H., Lee, K., and Shin, J. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018a.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018b.

Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

Liu, J. Z., Lin, Z., Padhy, S., Tran, D., Bedrax-Weiss, T., and Lakshminarayanan, B. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In *NeurIPS*, 2020a.

Liu, W., Wang, X., Owens, J., and Li, Y. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33, 2020b.

MacKay, D. J. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1992.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Murphy, K. P. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. Hybrid models with deep and invertible features. In *International Conference on Machine Learning*, pp. 4723–4732. PMLR, 2019.

Neal, R. M. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pp. 13991–14002, 2019.

Postels, J., Blum, H., Cadena, C., Siegwart, R., Van Gool, L., and Tombari, F. Quantifying aleatoric and epistemic uncertainty using density estimation in latent space. *arXiv preprint arXiv:2012.03082*, 2020.

Rosca, M., Weber, T., Gretton, A., and Mohamed, S. A case for new neural network smoothness constraints. *arXiv preprint arXiv:2012.07969*, 2020.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Smith, L. and Gal, Y. Understanding Measures of Uncertainty for Adversarial Example Detection. In *UAI*, 2018.

van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, pp. 9690–9700. PMLR, 2020.

van Amersfoort, J., Smith, L., Jesson, A., Key, O., and Gal, Y. Improving deterministic uncertainty estimation in deep learning for classification and regression. *arXiv preprint arXiv:2102.11409*, 2021.

Wen, Y., Tran, D., and Ba, J. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations*, 2019.

Wilson, A. G. and Izmailov, P. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*, 2020.

Winkens, J., Bunel, R., Roy, A. G., Stanforth, R., Natarajan, V., Ledsam, J. R., MacWilliams, P., Kohli, P., Karthikesalingam, A., Kohl, S., et al. Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566*, 2020.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

A. Related Work

Several methods to model uncertainty using feature-space density exist, but unfortunately under-perform compared to standard approaches for uncertainty. In this work we identify possible reasons for this failure, namely feature collapse and objective mismatch, and propose appropriate inductive biases to alleviate this failure. We note that our method also improves on the performance of Deep Ensembles, the state-of-the-art uncertainty method that requires significantly more compute.

Among the approaches that model feature-space density, Lee et al. (2018b) 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 Figure 1(d), for example, the feature density of a LeNet and a VGG are unable to distinguish OoD from iD samples. Postels et al. (2020) also propose a density-based estimation of aleatoric and epistemic uncertainty. Similar to (Lee et al., 2018b), 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 2.3 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 Appendix C. Likewise, Liu et al. (2020b) 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.

On the other hand, Winkens et al. (2020) use contrastive training on the feature extractor before estimating the feature-space density. Our method is separate from this work as we restrict ourselves to the supervised setting and show that the inductive biases that result in bi-Lipschitzness (van Amersfoort et al., 2020; Liu et al., 2020a) are a sufficient condition for the feature-space density to reliably capture epistemic uncertainty.

Lastly, our method differs from van Amersfoort et al. (2020) and Liu et al. (2020a) by alleviating the need for additional hyperparameters: DDU only needs minimal changes from the standard softmax setup to obtain excellent results on uncertainty benchmarks, and our GMM parameters are optimised for the already trained model using the training set. DDU does not require training or fine-tuning with OoD data. Note that Liu et al. (2020a) examine using the *softmax entropy and not the feature-space density* of a deterministic network with inductive biases (bi-Lipschitz) as a baseline to their proposed method, SNGP, and find it to underperform. We provide insights into why their baseline fails in Section 2.

B. Background

In this section, we provide a brief overview of approaches for quantifying uncertainty in deep learning and discuss related work to motivate our approach.

Bayesian Models (Neal, 2012; MacKay, 1992) provide a principled way of measuring uncertainty. Starting with a prior distribution $p(\omega)$ over model parameters ω, they infer a posterior $p(\omega|\mathcal{D})$, given the training data $\mathcal{D}$. The predictive distribution $p(y|x, \mathcal{D})$ for a given input x is computed via marginalisation over the posterior: $p(y|x, \mathcal{D}) = \mathbb{E}_{\omega \sim p(\omega|\mathcal{D})}[p(y|x, \omega)]$. As mentioned in Gal (2016) and Smith & Gal (2018), the predictive entropy $\mathbb{H}[Y|x, \mathcal{D}]$ of $p(y|x, \mathcal{D})$ upper-bounds the epistemic uncertainty, where epistemic uncertainty is quantified as the mutual information $\mathbb{I}[Y; \omega|x, \mathcal{D}]$ (*expected information gain*) between parameters ω and output y, following the equation:

$$\underbrace{\mathbb{H}[Y|x, \mathcal{D}]}_{\text{predictive}} = \underbrace{\mathbb{I}[Y; \omega|x, \mathcal{D}]}_{\text{epistemic}} + \underbrace{\mathbb{E}_{\mathrm{p}(\omega|\mathcal{D})}[\mathbb{H}[Y|x, \omega]]}_{\text{aleatoric (for iD } x)}. \tag{1}$$

Note that aleatoric uncertainty is only meaningful in-distribution, as, by definition, it quantifies the level of ambiguity between the different classes which might be observed for input x (i.e. if multiple labels were to be observed at x, aleatoric will be high[3]). Predictive uncertainty will be high whenever either epistemic uncertainty is high (for "far away inputs" such as OoD), or when aleatoric uncertainty is high (for near-by ambiguous inputs). The intractability of exact Bayesian inference in deep learning has led to the development of methods for approximate inference (Hinton & Van Camp, 1993; Hernández-Lobato & Adams, 2015; Blundell et al., 2015; Gal & Ghahramani, 2016). In practice, however, these methods

[3] More so, if the probability of observing x under the data generating distribution is zero, the probability $p(y|x) = \frac{p(x,y)}{p(x)}$, hence entropy as a measure of aleatoric uncertainty, is not defined.

are either unable to scale to large datasets and model architectures, suffer from low uncertainty quality, or require expensive Monte-Carlo sampling.

Deep Ensembles (Lakshminarayanan et al., 2017) are an ensemble of neural networks which average the models' softmax outputs. Uncertainty is then estimated as the entropy of this averaged softmax vector. Despite incurring a high computational overhead at training and test time, Deep Ensembles, along with recent extensions (Smith & Gal, 2018; Wen et al., 2019; Dusenberry et al., 2020) form the state-of-the-art in uncertainty quantification in deep learning. It is interesting to note that ensembling might also be seen as performing Bayesian Model Averaging (He et al., 2020; Wilson & Izmailov, 2020), as each ensemble member, producing a softmax output $p(y|x, \omega)$, can be considered to be drawn from some distribution $p(\omega|\mathcal{D})$ over the trained model parameters ω, which is induced by the pushforward of the weight initialization under stochastic optimization. As a result, Equation (1) can also be applied to Deep Ensembles to disentangle epistemic from predictive uncertainty.

In practice, both mutual information $\mathbb{I}[Y; \omega|x, \mathcal{D}]$ and predictive entropy $\mathbb{H}[Y|x, \mathcal{D}]$ are used in the literature to detect OoD samples. As seen in Equation (1), predictive entropy can be high for both iD ambiguous samples (high aleatoric uncertainty) as well as for OoD samples (high epistemic uncertainty). Hence, predictive entropy is only an effective measure for OoD detection when used with curated datasets that do not contain ambiguous samples, unlike Dirty-MNIST in Figure 1.

Deterministic Models produce a softmax distribution $p(y|x, \omega)$ and use either the maximum softmax probability $\max_c p(y = c|x, \omega)$ (*confidence*) or the *softmax entropy* $\mathbb{H}[Y|x, \omega]$ as a measure of uncertainty. It is well-known that these measures are often not indicative of OoD data (Hendrycks & Gimpel, 2016; Guo et al., 2017). Popular approaches to tackle this problem include pre-processing of inputs and post-hoc calibration methods (Liang et al., 2018; Guo et al., 2017), alternative objective functions (Lee et al., 2018a; DeVries & Taylor, 2018), and exposure to outliers (Hendrycks et al., 2018). However, these methods suffer from several shortcomings including failing to perform under distribution shift (Ovadia et al., 2019), requiring significant changes to the training setup, and assuming the availability of OoD samples during training (which many applications do not have access to). In the next section, we demonstrate that the softmax entropy is inherently inappropriate to capture epistemic uncertainty.

Feature-Space Distances and Density offer a different approach for estimating uncertainty in deterministic models (Lee et al., 2018b; van Amersfoort et al., 2020; Liu et al., 2020b;a; Postels et al., 2020). By definition, epistemic uncertainty must decrease when previously unseen samples are added to the training set. Feature space distances and density methods realise this by estimating distance (density) to training data in the feature space, where a previously unseen point with high distance (low density), once added to the training data, will have low distance (high density). Hence, they can be used as a proxy for epistemic uncertainty (under important assumptions about the feature space as detailed below). None of these methods, however, is competitive with the state-of-the-art (i.e., Deep Ensembles) in uncertainty quantification, potentially for the reasons discussed next.

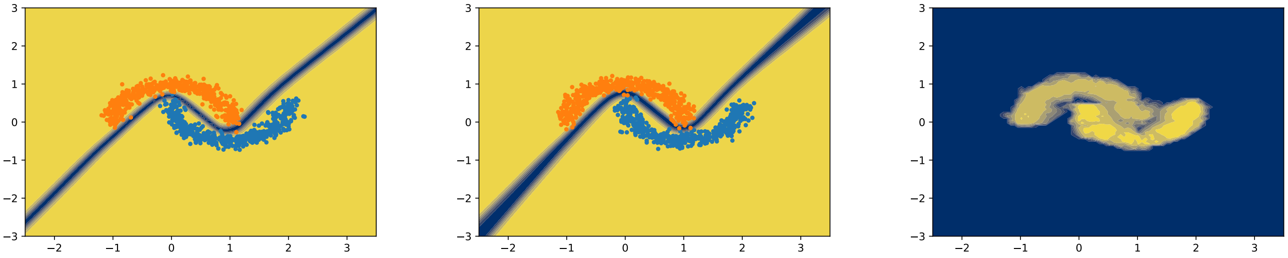
Feature Collapse (van Amersfoort et al., 2020) is a reason as to why distance and density estimation in the feature space may fail to capture epistemic uncertainty out of the box: feature extractors might map the features of OoD inputs to iD regions in the feature space (van Amersfoort et al., 2021, c.f. Figure 2). To fix this issue, a feature extractor f_θ, with parameters θ, can be subjected to a bi-Lipschitz constraint as follows:

$$K_1||\mathrm{x}_1 - \mathrm{x}_2||_I \leq ||f_\theta(\mathrm{x}_1) - f_\theta(\mathrm{x}_2)||_F \leq K_2||\mathrm{x}_1 - \mathrm{x}_2||_I, \tag{2}$$

for all inputs, x_1 and x_2, where $||.||_I$ and $||.||_F$ denote metrics in the input and feature space respectively, and K_1 and K_2 denote the lower and upper Lipschitz constants (Liu et al., 2020a). These are named *sensitivity* and *smoothness* conditions (van Amersfoort et al., 2020), where the lower Lipschitz bound ensures *sensitivity* to distances in the input space, and the upper Lipschitz bound ensures *smoothness* in the features, preventing them from becoming too sensitive to input variations, which, otherwise, can lead to poor generalisation and loss of robustness (Hess et al., 2020; Rosca et al., 2020). Contemporaneous methods of ensuring the bi-Lipschitz condition include: **i)** gradient penalty, by applying a two-sided penalty to the L2 norm of the Jacobian (Gulrajani et al., 2017), and **ii)** spectral normalisation (Miyato et al., 2018) in models with residual connections, like ResNets (He et al., 2016).

C. Experiments

In this section, we first show DDU's performance on the well-known Two Moons toy dataset. We then show that DDU can disentangle epistemic and aleatoric uncertainty. We evaluate DDU's quality of epistemic uncertainty estimation in active learning (Cohn et al., 1996) using both clean and ambiguous versions of MNIST and in OoD detection on the challenging

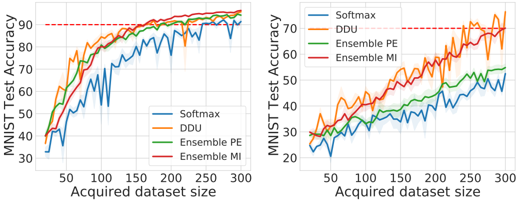


(a) Softmax Entropy (b) Ensemble Predictive Entropy (c) DDU Feature-space density

Figure 2: *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).

Table 1: *ECE for Dirty-MNIST test set and AUROC for Dirty-MNIST vs Fashion-MNIST.* ECE on D-MNIST is a measure for aleatoric uncertainty and D-MNIST vs F-MNIST requires detecting OoD samples, thereby epistemic uncertainty.

Model	ECE	AUROC (Softmax Entropy)	AUROC (Feature Density)
LeNet	2.22	84.23	71.41
VGG-16	2.11	84.04	89.01
ResNet18+SN (DDU)	2.34	83.01	**99.91**



(a) MNIST (b) Dirty-MNIST

Figure 3: *Active Learning* showing test accuracy vs acquired training set size with MNIST and Dirty-MNIST.

CIFAR-10 vs SVHN/CIFAR-100 and CIFAR-100 vs SVHN dataset pairings, where we set new state-of-the-art performance.

C.1. Toy-Setup: 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 (Liu et al., 2020a). The input is 2-dimensional and is projected into the 128 dimensional space using a fully connected layer. Using this 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 Figure 2, 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. Thus, we note that DDU captures uncertainty reliably in a small 2D setup like Two Moons.

C.2. Disentangling Epistemic and Aleatoric Uncertainty

We used a simple example in Section 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 (LeCun et al., 1998) as an iD dataset of unambiguous samples, Ambiguous-MNIST as an iD dataset of ambiguous samples and

Table 2: *OoD detection performance of different baselines using a Wide-ResNet-28-10 architecture with the CIFAR-10 vs SVHN/CIFAR-100 and CIFAR-100 vs SVHN dataset pairs averaged over 25 runs.* Note: SN stands for Spectral Normalisation, JP stands for Jacobian Penalty.

Train Dataset	**Method**	**Penalty**	**Aleatoric Uncertainty**	**Epistemic Uncertainty**	**Test Accuracy**	**Test ECE**	**AUROC SVHN**	**AUROC CIFAR100**
CIFAR-10	Softmax	-	Softmax Entropy	Softmax Entropy	95.98 ± 0.02	0.85 ± 0.02	94.44 ± 0.43	89.39 ± 0.06
	Energy-based (Liu et al., 2020b)	-	Softmax Entropy	Softmax Density	95.98 ± 0.02	0.85 ± 0.02	94.56 ± 0.51	88.89 ± 0.07
	DUQ (van Amersfoort et al., 2020)	JP	Kernel Distance	Kernel Distance	94.6 ± 0.16	1.55 ± 0.08	93.71 ± 0.61	85.92 ± 0.35
	SNGP (Liu et al., 2020a)	SN	Predictive Entropy	Predictive Entropy	96.04 ± 0.09	1.8 ± 0.1	94.0 ± 1.3	91.13 ± 0.15
	5-Ensemble (Lakshminarayanan et al., 2017)	-	Predictive Entropy	Predictive Entropy	96.59 ± 0.02	0.76 ± 0.03	97.73 ± 0.31	$\mathbf{92.13 \pm 0.02}$
				Mutual Information	96.59 ± 0.02	0.76 ± 0.03	97.18 ± 0.19	91.33 ± 0.03
	DDU (ours)	SN	Softmax Entropy	GMM Density	95.97 ± 0.03	0.85 ± 0.04	$\mathbf{97.86 \pm 0.19}$	91.34 ± 0.04
CIFAR-100					**Test Accuracy**	**Test ECE**	**AUROC SVHN**	
	Softmax	-	Softmax Entropy	Softmax Entropy	80.26 ± 0.06	4.62 ± 0.06	77.42 ± 0.57	
	Energy-based (Liu et al., 2020b)	-	Softmax Entropy	Softmax Density	80.26 ± 0.06	4.62 ± 0.06	78 ± 0.63	
	SNGP (Liu et al., 2020a)	SN	Predictive Entropy	Predictive Entropy	80.00 ± 0.11	4.33 ± 0.01	85.71 ± 0.81	
	5-Ensemble (Lakshminarayanan et al., 2017)	-	Predictive Entropy	Predictive Entropy	82.79 ± 0.10	3.32 ± 0.09	79.54 ± 0.91	
				Mutual Information	82.79 ± 0.10	3.32 ± 0.09	77.00 ± 1.54	
	DDU (ours)	SN	Softmax Entropy	GMM Density	80.98 ± 0.06	4.10 ± 0.08	$\mathbf{87.53 \pm 0.62}$	

Fashion-MNIST (Xiao et al., 2017) as an OoD dataset (see Figure 1(a)), with more details in Appendix D. We train a LeNet (LeCun et al., 1998), a VGG-16 (Simonyan & Zisserman, 2015) and a ResNet-18 (He et al., 2016) with spectral normalisation (SN) on Dirty-MNIST (a mix of Ambiguous- and standard MNIST) with the training setup detailed in Appendix F.1.

Table 1 gives a quantitative evaluation of the qualitative results in Section 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 Figure 1(c), 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 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 (Lee et al., 2018b)) 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 Table 1, also noticeable in Figure 1(d)).* 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 Section 2). 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.

C.3. Active Learning

We further demonstrate the quality of our uncertainty disentanglement in active learning (AL) (Cohn et al., 1996). AL aims to train models in a data-efficient manner. The goal is to label as few samples as possible while reaching a satisfactory model accuracy. Additional training samples are iteratively acquired from a large pool of unlabelled data and labelled with the help of an expert. After each acquisition step, the model is retrained on the newly expanded training set. This is repeated until the model achieves a desirable accuracy or when a maximum number of samples have been acquired.

Quickly obtaining a high-accuracy model with AL relies on acquiring the most informative samples during each acquisition step. This can be achieved by selecting points with high epistemic uncertainty (Gal et al., 2017). On the other hand, repeated acquisition of points with high aleatoric uncertainty is not informative for the model and such acquisitions lead to data inefficiency. AL, therefore, makes an excellent application for evaluating epistemic uncertainty and the ability of models to separate different sources of uncertainty. We evaluate DDU on two different setups: **i)** with clean MNIST samples in the pool set, and **ii)** with Dirty-MNIST, having a 1:60 ratio of MNIST to Ambiguous-MNIST samples, in the pool set.

Clean MNIST in the Pool set: In this setup, we use normal MNIST samples in the pool set and compare 3 baselines:

i) a ResNet-18 with softmax entropy as the acquisition function (samples with the top-k softmax entropies are acquired at each step), **ii)** DDU trained using a ResNet-18 with the feature density as the acquisition function (samples with the bottom-k densities are acquired at each step) , and **iii)** a Deep Ensemble of 3 ResNet-18s with the predictive entropy (PE) and mutual information (MI) of the ensemble as the acquisition functions (samples with the top-k predictive entropies or mutual information values are acquired at each step). We start with an initial training-set size of 20 randomly chosen MNIST points, and in each iteration, acquire 5 new samples. For each step, we train the models using Adam (Kingma & Ba, 2015) for 100 epochs and choose the one with the best validation set accuracy. We stop the process when the training set size reaches 300. In Figure 3(a), DDU clearly outperforms the deterministic softmax baseline and is competitive with Deep Ensembles. The softmax baseline reaches 90% test-set accuracy at a training-set size of 245. DDU reaches 90% accuracy at a training-set size of 160, whereas Deep Ensemble reaches the same at 185 and 155 training samples with PE and MI as the acquisition functions respectively. Note that DDU is three times faster than a Deep Ensemble, which needs to train three models independently after every acquisition.

Dirty-MNIST in the Pool set: The MNIST samples in the pool set of the previous setup lack ambiguity. However, real-life datasets often contain observation noise and ambiguous samples. What happens when the pool set contains a lot of such noisy samples having high aleatoric uncertainty? In such cases, it becomes important for models to identify unseen and informative samples with high epistemic uncertainty and not with high aleatoric uncertainty. In order to study this scenario in greater detail, we construct a pool set with samples from Dirty-MNIST (see Appendix C.2). We significantly increase the proportion of ambiguous samples by using a 1:60 split of MNIST to Ambiguous-MNIST (a total of 1K MNIST and 60K Ambiguous-MNIST samples). With this new pool set, we compare DDU with the deterministic softmax baseline as well as the Deep Ensembles. In Figure 3(b), the difference in the performance of DDU and the deterministic softmax model is stark. While DDU achieves a test set accuracy of 70% at a training set size of 240 samples, the accuracy of the softmax baseline peaks at a mere 50%. Note that DDU also performs better than Deep Ensembles with the PE acquisition function. The difference gets larger as the training set size grows: DDU's feature density and Deep Ensemble's MI solely capture epistemic uncertainty and hence, do not get confounded by iD ambiguous samples. This leads to the acquisition of unseen points from the pool set which have high epistemic uncertainty and low aleatoric uncertainty, thereby leading to better performance.

C.4. CIFAR-10 (iD) vs SVHN/CIFAR-100 (OoD) & CIFAR-100 (iD) vs SVHN (OoD)

OoD detection is an application of epistemic uncertainty quantification: we expect OoD data points to have higher epistemic uncertainty than iD data. We use CIFAR-10 (Krizhevsky et al., 2009) vs SVHN (Netzer et al., 2011)/CIFAR-100 and CIFAR-100 vs SVHN as iD vs OoD dataset pairs for this experiment. These pairings are known to be challenging (Nalisnick et al., 2019). The training setup is described in Appendix F.2. In addition to using softmax entropy of a deterministic model (*Softmax*) for both aleatoric and epistemic uncertainty, we also compare with the following baselines that do not require training or fine-tuning on OoD data:

- *Energy-based model* (Liu et al., 2020b): We use the softmax entropy of a deterministic model as aleatoric uncertainty and the unnormalized softmax density (using the logsumexp of the logits of the model) as epistemic uncertainty *without* regularisation to avoid feature collapse. We only compare with the version that does not train with OoD data.
- *DUQ (van Amersfoort et al., 2020), SNGP (Liu et al., 2020a)*: We compare with the state-of-the-art deterministic methods for uncertainty quantification including DUQ and SNGP. For SNGP, we use the entropy of the average of the MC softmax samples as uncertainty. For DUQ, we use the closest kernel distance. Note that for CIFAR-100, we noticed DUQ's centroids to not converge during training and hence, we do not include the DUQ baseline for CIFAR-100.
- *5-Ensemble*: We use an ensemble of 5 networks and compute the predictive entropy of the ensemble as both epistemic and aleatoric uncertainty and mutual information as epistemic uncertainty.

DDU beats Deep Ensembles on CIFAR-100 vs SVHN by a large margin. Table 2 shows the AUROC scores on CIFAR-10 vs SVHN/CIFAR-100 and CIFAR-100 vs SVHN along with their respective test set accuracy and test set ECE after temperature scaling. For DDU, post-hoc calibration, e.g. in the form of temperature scaling (Guo et al., 2017), is straightforward as it does not affect the GMM density. For OoD detection, *DDU outperforms all other methods including the state-of-the-art Deep Ensembles as well as DUQ, SNGP and the energy-based model approach from Liu et al. (2020b) on CIFAR-10 vs SVHN/CIFAR-100 and CIFAR-100 vs SVHN*. Importantly, the superior performance in OoD detection comes without compromising on the single-model test set accuracy (we do not claim to improve accuracy over Deep Ensembles).

Additional ablations for the CIFAR-10/100 experiments are detailed in Appendix G: Table 3 and Table 4. These tables show that *the feature density of a VGG-16 (i.e., without the appropriate inductive biases) is unable to beat a VGG-16* *ensemble, whereas a Wide-ResNet-28-10 (with appropriate inductive biases) outperforms its corresponding ensemble in almost all the cases*. This result further validates the importance of having the bi-Lipschitz constraint on the model to obtain smoothness and sensitivity. Finally, even without spectral normalisation, a Wide-ResNet-28 has the inductive bias of residual connections built into its model architecture, which can be a contributing factor towards good performance in general as residual connections already make the model sensitive to changes in the input space.

D. 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 (Kingma & Welling, 2014). 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).

E. Additional Architectural Changes

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 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.

F. Experimental Details

F.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 (Liu et al., 2020a), 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 Gouk et al. (2021). The coefficient for SN is a hyper-parameter which we set to 3 using cross-validation.

F.2. CIFAR-10 vs SVHN/CIFAR-100 & CIFAR-100 vs SVHN

We use Wide-ResNet-28-10 (Zagoruyko & Komodakis, 2016) as the model architecture for all the baselines. We train the softmax baselines 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 (Liu et al., 2020a; van Amersfoort et al., 2020).

F.3. Compute Resources

Each model (ResNet-18, Wide-ResNet-28-10 or VGG-16) used for the large scale active learning, CIFAR-10 vs SVHN/CIFAR-100 and CIFAR-100 vs SVHN tasks was trained on a single Nvidia Quadro RTX 6000 GPU. Each model

Ablations					Aleatoric Uncertainty	Epistemic Uncertainty	Test Accuracy (↑)	Test ECE (↓)	AUROC SVHN (↑)	AUROC CIFAR-100 (↑)
Architecture	Ensemble	Residual Connections	SN	GMM						
Wide-ResNet-28-10	✗	✓	✗	✗	Softmax Entropy	Softmax Entropy	95.98 ± 0.02	0.85 ± 0.02	94.44 ± 0.43	89.39 ± 0.06
						Softmax Density	95.98 ± 0.02	0.85 ± 0.02	94.56 ± 0.51	88.89 ± 0.07
				✓	Softmax Entropy	GMM Density	95.98 ± 0.02	0.85 ± 0.02	96.08 ± 0.25	90.94 ± 0.03
			✓	✗	Softmax Entropy	Softmax Entropy	95.97 ± 0.03	0.85 ± 0.04	94.05 ± 0.26	90.02 ± 0.07
						Softmax Density	95.97 ± 0.03	0.85 ± 0.04	94.31 ± 0.33	89.78 ± 0.08
				✓	**Softmax Entropy**	**GMM Density**	$\mathbf{95.97 \pm 0.03}$	$\mathbf{0.85 \pm 0.04}$	$\mathbf{97.86 \pm 0.19}$	$\mathbf{91.34 \pm 0.04}$
	✓	✓	✗	✗	Predictive Entropy	Predictive Entropy	96.59 ± 0.02	0.76 ± 0.03	97.73 ± 0.31	92.13 ± 0.02
						Mutual Information	96.59 ± 0.02	0.76 ± 0.03	97.18 ± 0.19	91.33 ± 0.03
VGG-16	✗	✗	✗	✗	Softmax Entropy	Softmax Entropy	93.63 ± 0.04	1.64 ± 0.03	85.76 ± 0.84	82.48 ± 0.14
						Softmax Density	93.63 ± 0.04	1.64 ± 0.03	84.24 ± 1.04	81.91 ± 0.17
				✓	Softmax Entropy	GMM Density	93.63 ± 0.04	1.64 ± 0.03	89.25 ± 0.36	86.55 ± 0.10
	✓	✗	✗	✗	Predictive Entropy	Predictive Entropy	94.9 ± 0.05	2.03 ± 0.03	92.80 ± 0.18	89.01 ± 0.08
						Mutual Information	94.9 ± 0.05	2.03 ± 0.03	91 ± 0.22	88.43 ± 0.08

Table 3: *OoD detection performance of different ablations trained on CIFAR-10 using Wide-ResNet-28-10 and VGG-16 architectures with SVHN and CIFAR-100 as OoD datasets averaged over 25 runs.* Note: SN stands for Spectral Normalisation.

Ablations					Aleatoric Uncertainty	Epistemic Uncertainty	Test Accuracy (↑)	Test ECE (↓)	AUROC SVHN (↑)
Architecture	Ensemble	Residual Connections	SN	GMM					
Wide-ResNet-28-10	✗	✓	✗	✗	Softmax Entropy	Softmax Entropy	80.26 ± 0.06	4.62 ± 0.06	77.42 ± 0.57
						Softmax Density	80.26 ± 0.06	4.62 ± 0.06	78.00 ± 0.63
				✓	Softmax Entropy	GMM Density	80.26 ± 0.06	4.62 ± 0.06	87.54 ± 0.61
			✓	✗	Softmax Entropy	Softmax Entropy	80.98 ± 0.06	4.10 ± 0.08	85.37 ± 0.36
						Softmax Density	80.98 ± 0.06	4.10 ± 0.08	86.41 ± 0.38
				✓	**Softmax Entropy**	**GMM Density**	$\mathbf{80.98 \pm 0.06}$	$\mathbf{4.10 \pm 0.08}$	$\mathbf{87.53 \pm 0.62}$
	✓	✓	✗	✗	Predictive Entropy	Predictive Entropy	82.79 ± 0.10	3.32 ± 0.09	79.54 ± 0.91
						Mutual Information	82.79 ± 0.10	3.32 ± 0.09	79.54 ± 0.91
VGG-16	✗	✗	✗	✗	Softmax Entropy	Softmax Entropy	73.48 ± 0.05	4.46 ± 0.05	76.73 ± 0.72
						Softmax Density	73.48 ± 0.05	4.46 ± 0.05	77.70 ± 0.86
				✓	Softmax Entropy	GMM Density	73.48 ± 0.05	4.46 ± 0.05	75.65 ± 0.95
	✓	✗	✗	✗	Predictive Entropy	Predictive Entropy	77.84 ± 0.11	5.32 ± 0.10	79.62 ± 0.73
						Mutual Information	77.84 ± 0.11	5.32 ± 0.10	72.07 ± 0.48

Table 4: *OoD detection performance of different ablations trained on CIFAR-100 using Wide-ResNet-28-10 and VGG-16 architectures with SVHN as the OoD dataset averaged over 25 runs.* Note: SN stands for Spectral Normalisation.

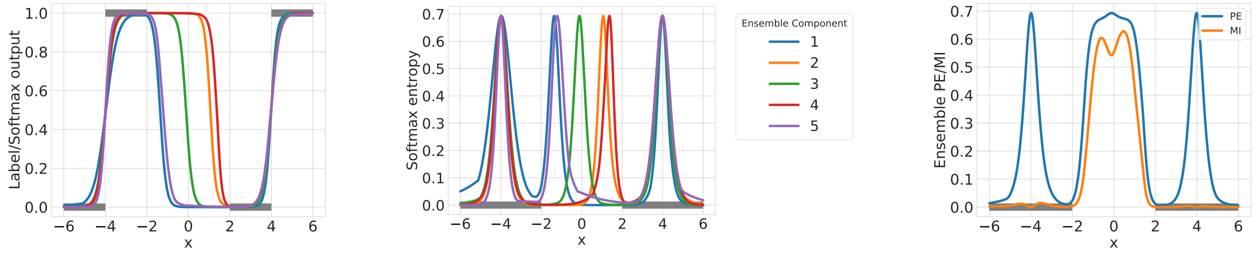
(LeNet, VGG-16 and ResNet-18) used to get the results in Figure 1 and Table 1 was trained on a single Nvidia GeForce RTX 2060 GPU.

G. Additional Results (CIFAR-10 vs SVHN/CIFAR-100 & CIFAR-100 vs SVHN)

In this section, we provide details of additional results on the OoD detection task using CIFAR-10 vs SVHN/CIFAR-100 and CIFAR-100 vs SVHN for various ablations on DDU. As mentioned in Section 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 without spectral normalisation. Note that VGG-16 does not have residual connections and hence, a VGG-16 trained without spectral normalisation 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 entropy of the model. On the other hand, to quantify the epistemic uncertainty, we use **i)** the softmax entropy, **ii)** the softmax density (Liu et al., 2020b) or **iii)** the GMM feature density (as described in Section 3).

Finally, for the purposes of comparison, we also present scores obtained by a 5-Ensemble of the respective architectures



(a) Softmax output

(b) Softmax entropy

(c) PE and MI of ensemble

Figure 4: *Visualisation of softmax outputs/entropies along with the Predictive Entropy (PE) and Mutual Information of a 5-Ensemble.* Figures (a) and (b) show that the softmax entropy is high at points of ambiguity, i.e., where the label changes from 0 to 1 for the data, thereby capturing aleatoric uncertainty, whereas softmax entropy can be low or high for OoD (between -2 to 2). At the same time, figure (c) shows that the MI of the ensemble is only high for OoD, thereby solely capturing epistemic uncertainty, whereas the PE of the ensemble is high for both OoD and for regions of ambiguity, thereby capturing both epistemic and aleatoric uncertainty.

(i.e., Wide-ResNet-28-10 and VGG-16) in Table 3 for CIFAR-10 vs SVHN/CIFAR-100 and in Table 4 for CIFAR-100 vs SVHN. Based on these results, we can make the following observations (in addition to the ones we make in Appendix C.4):

Inductive biases are important for feature density. From the AUROC scores in Table 3, 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 Figure 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 Section 2) on the feature space of the model to obtain feature densities that 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, the best of both worlds. 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.

H. 5-Ensemble Visualisation

In Figure 4, 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 Section 2). We train the networks on 1-dimensional data with binary labels 0 and 1. The data is shown in Figure 4(b). From Figure 4(a) and Figure 4(b), we find that softmax entropy is high in regions of ambiguity where the label changes from 0 to 1 (i.e., at x value -4 and 4). 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 Figure 4(a)) and have softmax entropies which can be high, low or anywhere in between (see Figure 4(b)) following our claim in Section 2. 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.

I. Theoretical Results

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

I.1.1. Qualitative Statement

We start with a proof of Proposition 2.1, 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 2.1. *Let x_1 and x_2 be points such that x_1 has* ***higher*** *epistemic uncertainty than x_2 under the ensemble: $\mathbb{I}[Y_1; \omega \mid x_1, \mathcal{D}] > \mathbb{I}[Y_2; \omega \mid x_2, \mathcal{D}] + \delta$, with $\delta \geq 0$. Further assume both have similar predictive entropy $|\,\mathbb{H}[Y_1 \mid x_1, \mathcal{D}] - \mathbb{H}[Y_2 \mid x_2, \mathcal{D}]| \leq \epsilon$, for $\epsilon \geq 0$. 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: $\mathbb{H}[Y_1 \mid x_1, \omega] < \mathbb{H}[Y_2 \mid x_2, \omega] - (\delta - \epsilon)$.*

Proof. From Equation (1), we obtain

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

and hence we have

$$\begin{aligned} &\mathbb{E}_{\mathrm{p}(\omega|\mathcal{D})}\left[\mathbb{H}[Y_1 \mid x_1, \omega]\right] - \mathbb{E}_{\mathrm{p}(\omega|\mathcal{D})}\left[\mathbb{H}[Y_2 \mid x_2, \omega]\right] \\ &+ \underbrace{(\mathbb{I}[Y_1; \omega \mid x_1, \mathcal{D}] - \mathbb{I}[Y_2; \omega \mid x_2, \mathcal{D}])}_{>\delta} \leq \epsilon. \end{aligned} \tag{4}$$

We rearrange the terms:

$$\mathbb{E}_{\mathrm{p}(\omega|\mathcal{D})}\left[\mathbb{H}[Y_1 \mid x_1, \omega]\right] < \mathbb{E}_{\mathrm{p}(\omega|\mathcal{D})}\left[\mathbb{H}[Y_2 \mid x_2, \omega]\right] - (\delta - \epsilon). \tag{5}$$

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

$$\mathbb{H}[Y_1 \mid x_1, \omega] < \mathbb{H}[Y_2 \mid x_2, \omega] - (\delta - \epsilon), \tag{6}$$

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.

I.1.2. Infinite Deep Ensemble

There are two interpretations of the ensemble parameter distribution $\mathrm{p}(\omega \mid \mathcal{D})$: we can view it as an empirical distribution given a specific ensemble with members $\omega_{i \in \{1,\ldots,K\}}$, 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 $\mathbb{I}[Y; \omega \mid x, \mathcal{D}]$.

I.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 a lower bound for the variance. Empirically, we find the 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 $(Y_1, \ldots, Y_K) \sim \mathrm{Dir}(\alpha)$ with non-negative concentration parameters $\alpha = (\alpha_1, \ldots, \alpha_K)$ 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 I.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:*

$$\mathbb{E}\left[\log \mathbf{p}_i\right] = \psi(\alpha_i) - \psi(\alpha_0). \tag{7}$$

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

$$\mathrm{Cov}[\log \mathbf{p}_i, \log \mathbf{p}_j] = \psi'(\alpha_i)\, \delta_{ij} - \psi'(\alpha_0). \tag{8}$$

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

$$\begin{aligned} &\mathbb{E}\left[\mathbf{p}_i^n \mathbf{p}_j^m \log \mathbf{p}_i\right] \\ &= \frac{\alpha_i^{\overline{n}}\, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n+m}}} \left(\psi(\alpha_i + n) - \psi(\alpha_0 + n + m)\right), \end{aligned} \tag{9}$$

where $i \neq j$*, and* $n^{\overline{k}} = n\,(n+1)\,\ldots\,(n+k-1)$ *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

$$A(\alpha) = \sum_{i=1}^{k} \log \Gamma\left(\alpha_i\right) - \log \Gamma\left(\alpha_0\right), \tag{10}$$

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

$$\mathbb{E}\left[T_i\right] = \frac{\partial A(\alpha)}{\partial \alpha_i} \tag{11}$$

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

$$\mathbb{E}\left[\log \mathbf{p}_i\right] = \frac{\partial}{\partial \alpha_i}\left[\sum_{i=1}^{k} \log \Gamma\left(\alpha_i\right) - \log \Gamma\left(\sum_{i=1}^{k} \alpha_i\right)\right] \tag{12}$$

$$= \psi\left(\alpha_i\right) - \psi\left(\alpha_0\right) \frac{\partial}{\partial \alpha_i} \alpha_0, \tag{13}$$

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 (7) as $\frac{\partial}{\partial \alpha_i} \alpha_0 = 1$.
2. Similarly, the covariance is obtained using a second-order partial derivative:

$$\operatorname{Cov}[T_i, T_j] = \frac{\partial^2 A(\alpha)}{\partial \alpha_i \, \partial \alpha_i}. \tag{14}$$

Again, substituting yields

$$\begin{aligned} \operatorname{Cov}[\log \mathbf{p}_i, \log \mathbf{p}_j] &= \frac{\partial}{\partial \alpha_j} \left[\psi\left(\alpha_i\right) - \psi\left(\alpha_0\right) \right] &(15)\\ &= \psi'\left(\alpha_i\right) \delta_{ij} - \psi'\left(\alpha_0\right). &(16) \end{aligned}$$

3. We will make use of a simple reparameterization to prove the statement using Equation (7). Expanding the expectation and substituting the density $\operatorname{Dir}(\mathbf{p}; \alpha)$, we obtain

$$\begin{aligned} \mathbb{E}\left[\mathbf{p}_i^n \mathbf{p}_j^m \log \mathbf{p}_i\right] &= \int \operatorname{Dir}(\mathbf{p}; \alpha)\, \mathbf{p}_i^n\, \mathbf{p}_j^m\, \log \mathbf{p}_i \, d\mathbf{p} &(17)\\ &= \int \frac{\Gamma\left(\alpha_0\right)}{\prod_{i=1}^K \Gamma\left(\alpha_i\right)} \prod_{k=1}^K \mathbf{p}_k^{\alpha_k - 1}\, \mathbf{p}_i^n\, \mathbf{p}_j^m\, \log \mathbf{p}_i \, d\mathbf{p} &(18)\\ &= \frac{\Gamma(\alpha_i + n)\Gamma(\alpha_j + m)\Gamma(\alpha_0 + n + m)}{\Gamma(\alpha_i)\Gamma(\alpha_j)\Gamma(\alpha_0)} \\ &\quad \int \operatorname{Dir}(\hat{\mathbf{p}}; \hat{\alpha})\, \hat{\mathbf{p}}_i^n\, \hat{\mathbf{p}}_j^m\, \log \hat{\mathbf{p}}_i \, d\hat{\mathbf{p}} &(19)\\ &= \frac{\alpha_i^{\overline{n}}\, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n+m}}} \mathbb{E}\left[\log \hat{\mathbf{p}}_i\right], &(20) \end{aligned}$$

where $\hat{\mathbf{p}} \sim \operatorname{Dir}(\hat{\alpha})$ 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 Equation (7) on $\hat{\mathbf{p}} \sim \operatorname{Dir}(\hat{\alpha})$ to show

$$= \frac{\alpha_i^{\overline{n}}\, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n+m}}} \left(\psi(\alpha_i + n) - \psi(\alpha_0 + n + m)\right). \tag{21}$$

□

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

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

$$\mathbb{E}_{\mathrm{p}(\mathbf{p} \mid \alpha)} \mathbb{H}[Y \mid \mathbf{p}] = \psi(\alpha_0 + 1) - \sum_{y=1}^K \frac{\alpha_i}{\alpha_0} \psi(\alpha_i + 1). \tag{22}$$

Proof. Applying the sum rule of expectations and Equation (9) from Lemma I.1, we can write

$$\begin{aligned} \mathbb{E}\, \mathbb{H}[Y \mid \mathbf{p}] = \mathbb{E}\left[-\sum_{i=1}^K \mathbf{p}_i \log \mathbf{p}_i \right] &= -\sum_i \mathbb{E}\left[\mathbf{p}_i \log \mathbf{p}_i\right] &(23)\\ &= -\sum_i \frac{\alpha_i}{\alpha_0} \left(\psi(\alpha_i + 1) - \psi(\alpha_0 + 1)\right). &(24) \end{aligned}$$

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 I.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*

$$\mathrm{Cov}[\mathbf{p}_i^n \log \mathbf{p}_i, \mathbf{p}_j^m \log \mathbf{p}_j] \tag{25}$$

$$\begin{aligned} &= \frac{\alpha_i^{\overline{n}}\,\alpha_j^{\overline{m}}}{\alpha_0^{\overline{n+m}}}\,((\psi(\alpha_i+n)-\psi(\alpha_0+n+m)) \\ &\quad (\psi(\alpha_j+m)-\psi(\alpha_0+n+m)) \\ &\quad -\psi'(\alpha_0+n+m)) \\ &\quad + \frac{\alpha_i^{\overline{n}}\,\alpha_j^{\overline{m}}}{\alpha_0^{\overline{n}}\,\alpha_0^{\overline{m}}}(\psi(\alpha_i+n)-\psi(\alpha_0+n)) \\ &\quad (\psi(\alpha_j+m)-\psi(\alpha_0+n)), \end{aligned} \tag{26}$$

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*

$$\mathrm{Cov}[\mathbf{p}_i^n \log \mathbf{p}_i, \mathbf{p}_i^m \log \mathbf{p}_i] \tag{27}$$

$$\begin{aligned} &= \frac{\alpha_i^{\overline{n+m}}}{\alpha_0^{\overline{n+m}}}\,\big((\psi(\alpha_i+n+m)-\psi(\alpha_0+n+m))^2 \\ &\quad + \psi'(\alpha_i+n+m)-\psi'(\alpha_0+n+m)\big) \\ &\quad + \frac{\alpha_i^{\overline{n}}\,\alpha_i^{\overline{m}}}{\alpha_0^{\overline{n}}\,\alpha_0^{\overline{m}}}(\psi(\alpha_i+n)-\psi(\alpha_0+n)) \\ &\qquad (\psi(\alpha_i+m)-\psi(\alpha_0+n)). \end{aligned} \tag{28}$$

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

Definition I.4. *We will denote by*

$$\overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^{n,m}\right]} = \psi(\alpha_i+n)-\psi(\alpha_0+n+m), \tag{29}$$

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,*

$$\overline{\mathrm{Cov}[\log \hat{\mathbf{p}}_i^{n,m}, \log \hat{\mathbf{p}}_j^{n,m}]} = \psi'(\alpha_i+n)\delta_{ij}-\psi'(\alpha_0+n+m). \tag{30}$$

This notation agrees with the proof of Equation (7) and (8) in Lemma I.1. With this, we can significantly simplify the previous statements:

Corollary I.4.1. *Given a Dirichlet distribution and random variable* $\mathbf{p} \sim \mathrm{Dir}(\alpha)$*,*

$$\mathbb{E}\left[\mathbf{p}_i^n \mathbf{p}_j^m \log \mathbf{p}_i\right] = \frac{\alpha_i^{\overline{n}}\,\alpha_j^{\overline{m}}}{\alpha_0^{\overline{n+m}}}\overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^{n,m}\right]}, \tag{31}$$

$$\begin{aligned}
&\mathrm{Cov}[\mathbf{p}_i^n \log \mathbf{p}_i, \mathbf{p}_j^m \log \mathbf{p}_j] && (32)\\
&\quad = \frac{\alpha_i^{\overline{n}}\, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n+m}}} \Big(\overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^{n,m}\right]}\, \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_j^{m,n}\right]} \\
&\qquad\qquad \overline{\mathrm{Cov}[\log \hat{\mathbf{p}}_i^{n,m}, \log \hat{\mathbf{p}}_j^{n,m}]} \Big) && (33)\\
&\qquad + \frac{\alpha_i^{\overline{n}}\, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n}}\, \alpha_0^{\overline{m}}} \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^{n}\right]}\, \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_j^{m}\right]} \quad \textit{for } i \neq j\textit{, and}\\
&\mathrm{Cov}[\mathbf{p}_i^n \log \mathbf{p}_i, \mathbf{p}_i^m \log \mathbf{p}_i] && (34)\\
&\quad = \frac{\alpha_i^{\overline{n+m}}}{\alpha_0^{\overline{n+m}}} \Big(\overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^{n+m}\right]}^2 \\
&\qquad + \overline{\mathrm{Cov}[\log \hat{\mathbf{p}}_i^{n+m}, \log \hat{\mathbf{p}}_i^{n+m}]} \Big) && (35)\\
&\qquad + \frac{\alpha_i^{\overline{n}}\, \alpha_i^{\overline{m}}}{\alpha_0^{\overline{n}}\, \alpha_0^{\overline{m}}} \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^{n}\right]}\, \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_j^{m}\right]}.
\end{aligned}$$

Proof of Lemma I.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 Equation (9) several times:

$$\begin{aligned}
&\mathrm{Cov}[\mathbf{p}_i^n \log \mathbf{p}_i, \mathbf{p}_j^m \log \mathbf{p}_j] && (36)\\
&\overset{\textbf{cov}}{=} \mathbb{E}\left[\mathbf{p}_i^n \log(\mathbf{p}_i)\, \mathbf{p}_j^m \log(\mathbf{p}_j)\right] \\
&\quad - \mathbb{E}\left[\mathbf{p}_i^n \log \mathbf{p}_i\right] \mathbb{E}\left[\mathbf{p}_j^m \log \mathbf{p}_j\right] && (37)\\
&= \frac{\alpha_i^{\overline{n}}\, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n+m}}}\, \mathbb{E}\left[\log(\hat{\mathbf{p}}_i^{i,j}) \log(\hat{\mathbf{p}}_j^{i,j})\right] \\
&\quad - \mathbb{E}\left[\log \hat{\mathbf{p}}_i^{i}\right] \mathbb{E}\left[\log \mathbf{p}_j^{j}\right] && (38)\\
&\overset{\textbf{(rcov)}}{=} \frac{\alpha_i^{\overline{n}}\, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n+m}}} \Big(\mathrm{Cov}[\log \hat{\mathbf{p}}_i^{i,j}, \log \hat{\mathbf{p}}_j^{i,j}] \\
&\qquad + \mathbb{E}\left[\log \hat{\mathbf{p}}_i^{i,j}\right] \mathbb{E}\left[\log \hat{\mathbf{p}}_j^{i,j}\right] \Big) && (39)\\
&\quad - \frac{\alpha_i^{\overline{n}}\, \alpha_j^{\overline{m}}}{\alpha_0^{\overline{n}}\, \alpha_0^{\overline{m}}}\, \mathbb{E}\left[\log \hat{\mathbf{p}}_i^{i}\right] \mathbb{E}\left[\log \mathbf{p}_j^{j}\right],
\end{aligned}$$

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 Equation (8) and Equation (7) from Lemma I.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 I.5. *Given a Dirichlet distribution and a random variable* $\mathbf{p} \sim$ *Dir*(α)*, 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*

$$\begin{aligned}
&\mathrm{Var}[\mathbb{H}[Y \mid \mathbf{p}]] && (40)\\
&= \sum_i \frac{\alpha_i^{\overline{2}}}{\alpha_0^{\overline{2}}} \left(\overline{\mathrm{Cov}[\log \hat{\mathbf{p}}_i^2, \log \hat{\mathbf{p}}_i^2]} + \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^2\right]}^2 \right) \\
&+ \sum_{i \neq j} \frac{\alpha_i\, \alpha_j}{\alpha_0^{\overline{2}}} \left(\overline{\mathrm{Cov}[\log \hat{\mathbf{p}}_i^1, \log \hat{\mathbf{p}}_j^1]} + \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^{1,1}\right]}\; \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_j^{1,1}\right]} \right) && (41)\\
&- \sum_{i,j} \frac{\alpha_i\, \alpha_j}{\alpha_0^2} \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_i^1\right]}\, \overline{\mathbb{E}\left[\log \hat{\mathbf{p}}_j^1\right]}.
\end{aligned}$$

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 I.3 repeatedly. □

Main Result

All the above allows us to formulate our main result for an ensemble with a given predictive entropy $\mathrm{p}(y \mid x)$ and mutual information $\mathbb{I}[Y; \omega \mid x]$:

Theorem I.6. *Fix* $\mathrm{p}(y \mid x)$, $\mathbb{I}[Y; \omega \mid x]$, *a distribution over models* $\mathrm{p}(\omega \mid \mathcal{D})$, *and a sample* x. *The maximum-entropy estimate of the distribution over probability vectors* $\mathrm{p}(y \mid x, \omega)$ *for models* $\omega \sim \mathrm{p}(\omega \mid \mathcal{D})$ *given* $\mathrm{p}(y \mid x)$ *and* $\mathbb{I}[Y; \omega \mid x]$ *is a Dirichlet distribution* $\mathbf{p} \sim \mathrm{Dir}(\alpha)$ *that satisfies:*

$$\mathrm{p}(y \mid x) = \frac{\alpha_i}{\alpha_0} \tag{42}$$

$$\mathbb{H}[Y \mid x] - \mathbb{I}[Y; \omega \mid x] = \psi(\alpha_0 + 1) - \sum_{y=1}^{K} \mathrm{p}(y \mid x) \psi(\alpha_0\ \mathrm{p}(y \mid x) + 1). \tag{43}$$

The variance $\mathrm{Var}[\mathbb{H}[Y \mid x, \omega]]$ *of the softmax entropy over models* $\omega \sim \mathrm{p}(\omega \mid \mathcal{D})$ *is bounded by* $\mathrm{Var}[\mathbb{H}[Y \mid \mathbf{p}]]$*:*

$$\mathrm{Var}_{\omega}[\mathbb{H}[Y \mid x, \omega]] \geq \mathrm{Var}_{\mathbf{p}}[\mathbb{H}[Y \mid \mathbf{p}]] \tag{44}$$

with the latter term given in eq. (41).

Proof. We can use moment matching to fix the distribution. Equation (42) is just the mean $\mathbb{E}\,[Y]$, and Equation (43) follows from Equation (1) and Lemma I.2 when we substitute $\frac{\alpha_i}{\alpha_0} = \mathbf{p}_i$. It is easy to check that Equation (43) is monotonously increasing in α_0, which is thus uniquely determined. Furthermore, we can use the variance of the entropy of the maximum entropy estimate as a lower bound because the Dirichlet distribution is a maximum entropy distribution. □

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.

Corollary I.6.1. *Assuming that* $p(y|x, \omega)$ *only depends on* $\mathrm{p}(y \mid x)$ *and* $\mathbb{I}[Y; \omega \mid x]$, *the maximum-entropy estimate for the distribution* $p(y|x, \omega)$ *for a given* ω *and different OoD* x *is given by a Dirichlet distribution* $\mathrm{Dir}(\alpha)$ *that satisfies:*

$$\mathrm{p}(y \mid x) = \frac{\alpha_i}{\alpha_0} \tag{45}$$

$$\mathbb{H}[Y \mid x] - \mathbb{I}[Y; \omega \mid x] = \psi(\alpha_0 + 1) \tag{46}$$

$$- \sum_{y=1}^{K} \mathrm{p}(y \mid x) \psi(\alpha_0\ \mathrm{p}(y \mid x) + 1).. \tag{47}$$

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

$$\mathrm{p}(y \mid x, \omega) \approxeq \mathrm{Cat}(\mathbf{p}). \tag{48}$$

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 bounded by* $\mathrm{Var}[\mathbb{H}[Y \mid \mathbf{p}]]$*:*

$$\mathrm{Var}_{\omega}[\mathbb{H}[Y \mid x, \omega]] \geq \mathrm{Var}_{\mathbf{p}}[\mathbb{H}[Y \mid \mathbf{p}]] \tag{49}$$

with the latter term given in eq. (41).

We empirically find this to be true.

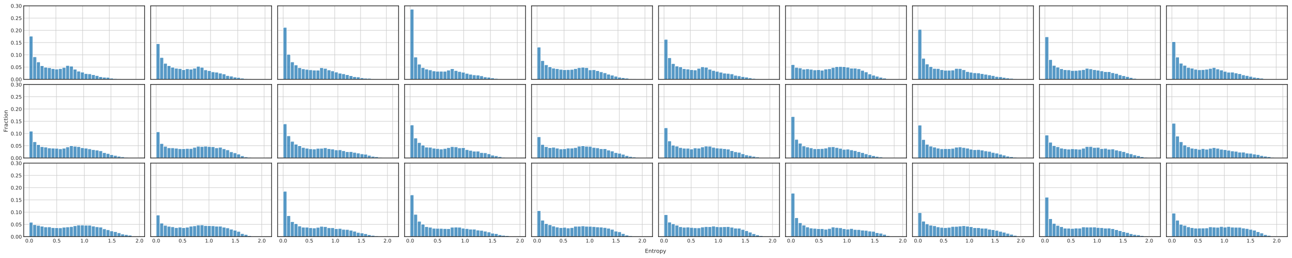
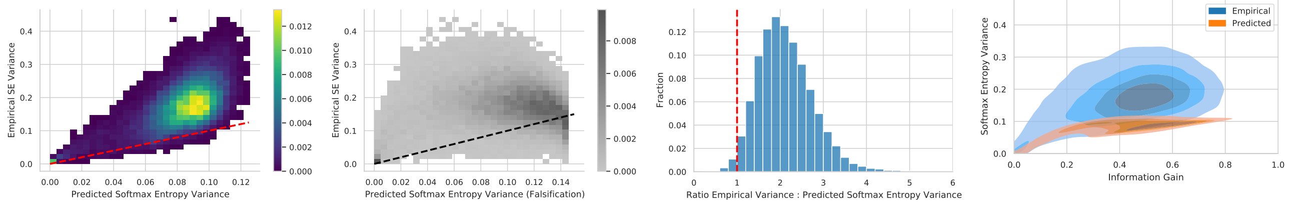


Figure 5: *Softmax entropy histograms of 30* Wide-ResNet-28-10+SN *models trained on CIFAR-10, evaluated on SVHN (OoD).* The softmax entropy distribution of the different models varies considerably.



(a) Verification

(b) Falsification

(c) Ratio

(d) Density (Expected Information Gain)

Figure 6: *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 the prediction using Theorem I.6. The red dashed line depicts equality. **(b)** Using uniform predictions in Equation (44) leads to a violation of the predicted lower bounds (black dashed line). **(c)** The ratio histogram shows that there are only few violations due to precision issues ($< 2\%$ (a) vs $< 22\%$ for (b), not depicted). **(d)** The variance of the softmax entropy is not linearly correlated to the epistemic uncertainty. For both high and low epistemic uncertainty, the variance decreases. It is still lower-bounded by Equation (44).

Empirical Results

We empirically verify that softmax entropies vary considerably in Figure 5. In Figure 6, we verify the validity of Equation (44) empirically. Morever, Figure 6(d) 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 Figure 7.

I.2. Capturing aleatoric and epistemic uncertainty requires multiple mixture models

In Section 2, 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.

I.2.1. Preliminaries

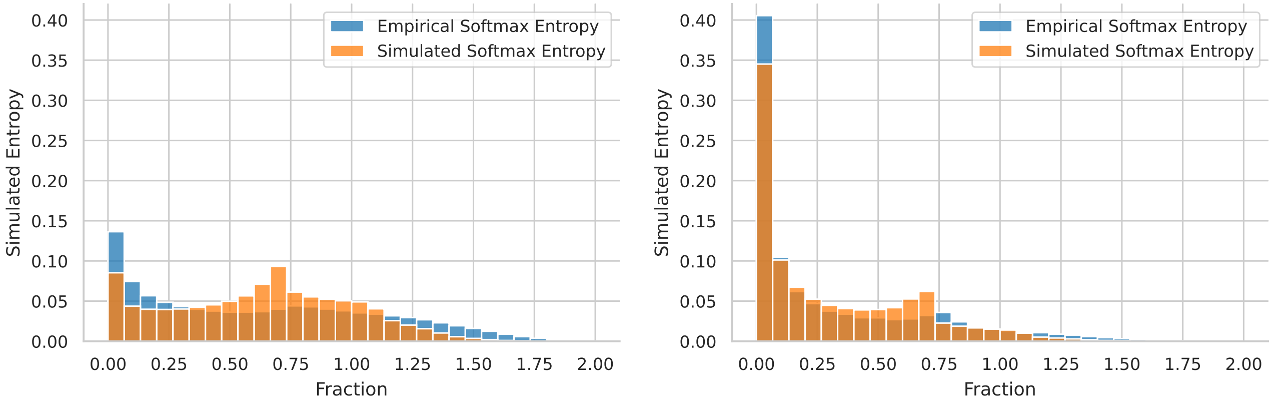
Before we prove Proposition 2.3, we will introduce some additional notation following Kirsch et al. (2020).

Definition I.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 $\hat{\mathrm{p}}(\cdot)$*:*

$$\mathbb{H}[\cdot] = \mathbb{H}\left(\hat{\mathrm{p}}(\cdot)\right) = \mathbb{E}_{\hat{\mathrm{p}}(\cdot)}\left[-\log \hat{\mathrm{p}}(\cdot)\right]. \tag{50}$$

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:*

$$\mathbb{H}[Y \mid Z] = \mathbb{E}_{\hat{\mathrm{p}}(z)}\, \mathbb{H}[Y \mid z]. \tag{51}$$



(a) WideResNet-28-10+SN

(b) VGG16

Figure 7: *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.

5. *$\mathbb{H}(\mathrm{p}(y,z) \,||\, \mathrm{q}(y|z|))$ is the cross-entropy of $\mathrm{q}(y \mid z)$ 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 $\mathrm{q}(y \mid z)$ under $\mathrm{p}(y \mid z)$ 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 I.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 I.9.

$$\begin{aligned}\mathbb{H}_\theta[Y,Z] &= \mathbb{H}_\theta[Y \mid Z] + \mathbb{H}_\theta[Z] &(52)\\ &= \mathbb{H}_\theta[Z \mid Y] + \mathbb{H}_\theta[Y]. &(53)\end{aligned}$$

Proof. We substitute the definitions and obtain:

$$\mathbb{H}_\theta[Y, Z] = \mathbb{E}_{\mathrm{p}(y,z)}\left[-\log \mathrm{q}(y, z)\right] \tag{54}$$

$$= \mathbb{E}_{\mathrm{p}(y,z)}\left[-\log \mathrm{q}(y \mid z)\right] + \mathbb{E}_{\mathrm{p}(y,z)}\left[-\log \mathrm{q}(z)\right] \tag{55}$$

$$= \mathbb{H}_\theta[Y \mid Z] + \mathbb{H}_\theta[Z]. \tag{56}$$

□

The same holds for entropies: $\mathbb{H}[Y, Z] = \mathbb{H}[Y \mid Z] + \mathbb{H}[Z] = \mathbb{H}[Y \mid Z] + \mathbb{H}[Y]$ (Cover, 1999).

I.2.2. Proof

We can now prove the observation.

Proposition 2.3. *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 $\log \mathrm{q}(z)$;*
3. *A mixture model $\mathrm{q}(y, z)$ cannot generally maximise both objectives, conditional log-likelihood and marginalised log-likelihood, at the same time. In the specific instance that it 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 (Gneiting & Raftery, 2007). The optimization objective can be rewritten as

$$\max_\theta \mathbb{E}_{\log \mathrm{p}_\theta(y|z)} = \min_\theta \mathbb{H}_\theta[Y \mid Z] \geq \mathbb{H}[Y \mid Z]. \tag{57}$$

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. $\mathbb{H}[Y \mid z]$ is the irreducible residual conditional entropy, the aleatoric uncertainty.
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

$$\max_\theta \mathbb{E}_{\hat{\mathrm{p}}(y,z)} \log \mathrm{p}_\theta(z) = \min_\theta \mathbb{H}_\theta[Z] \geq \mathbb{H}[Z]. \tag{58}$$

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 $\mathbb{H}_\theta[Y, Z] = \mathbb{H}_\theta[Z \mid Y] + \mathbb{H}_\theta[Y]$, we can relate the objectives from Equation (57) and (58) 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 $\nabla\, \mathbb{H}_\theta[Z \mid Y] = 0$ and $\nabla\, \mathbb{H}_\theta[Z] = 0$, and we obtain

$$\nabla\, \mathbb{H}_\theta[Y, Z] = \nabla\, \mathbb{H}_\theta[Z \mid Y] + \nabla\, \mathbb{H}_\theta[Y] = 0. \tag{59}$$

From this we conclude that for 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 $\mathbb{H}_\theta[Y, Z]$:

$$\nabla\, \mathbb{H}_\theta[Z \mid Y] = -\nabla\, \mathbb{H}_\theta[Y] \quad \text{when } \nabla\, \mathbb{H}_\theta[Y, Z] = 0 \tag{60}$$

As can easily be verified, a trivial 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:

$$\mathrm{q}^*(y, z) = \sum_y \hat{\mathrm{p}}(y)\, \mathbb{E}_{\hat{z} \sim \hat{\mathrm{p}}(z|y)}\, \mathcal{N}(z;\, \hat{z},\, \sigma^2 \mathbf{I}), \tag{61}$$

for small enough σ. However, generally, there is no guaranteed shared optimum for other generative classifiers.

Specifially, we will examine GMMs 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 Equation (58). The difference is that using a GMM as a density estimator does not constrain the component assignment, unlike in GDA. Consequently, we can intuitively see that *all 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 example that shows this: if we have two classes whose features lie in well-separated clusters, GDA will minimize all objectives. The opposite does not hold: it will fail for example if some of the class labels are flipped. An optimal density estimator will still fit to the original clusters, while GDA will not.

Given that perfect separation is impossible with ambiguous data, a shared optimum is rare, but only then is GDA optimal. In all other cases, GDA does not optimize both objectives, and neither can any other GMM (with one component per class). Moreover, Equation (60) 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.

□