LAST LAYER EMPIRICAL BAYES

Valentin Villecroze
Layer 6 AI
valentin.v@layer6.ai

Yixin Wang
University of Michigan
yixinw@umich.edu

Gabriel Loaiza-Ganem
Layer 6 AI
gabriel@layer6.ai

ABSTRACT

The task of quantifying the inherent uncertainty associated with neural network predictions is a key challenge in artificial intelligence. Bayesian neural networks (BNNs) and deep ensembles are among the most prominent approaches to tackle this task. Both approaches produce predictions by computing an expectation of neural network outputs over some distribution on the corresponding weights; this distribution is given by the posterior in the case of BNNs, and by a mixture of point masses for ensembles. Inspired by recent work showing that the distribution used by ensembles can be understood as a posterior corresponding to a learned data-dependent prior, we propose last layer empirical Bayes (LLEB). LLEB instantiates a learnable prior as a normalizing flow, which is then trained to maximize the evidence lower bound; to retain tractability we use the flow only on the last layer. We show why LLEB is well motivated, and how it interpolates between standard BNNs and ensembles in terms of the strength of the prior that they use. LLEB performs on par with existing approaches, highlighting that empirical Bayes is a promising direction for future research in uncertainty quantification.

1 INTRODUCTION

Uncertainty quantification (UQ) is a crucial task in scientific and safety-critical settings (Esteva et al., 2017; Bojarski et al., 2016; Litjens et al., 2017; Psaros et al., 2023), and every improvement in UQ within deep learning is a step towards a broader adoption of AI. The two most popular approaches for UQ are Bayesian neural networks (BNNs; Welling & Teh, 2011; Graves, 2011; Hernández-Lobato & Adams, 2015; Blundell et al., 2015; Gal & Ghahramani, 2016; Ritter et al., 2018) and deep ensembles (Lakshminarayanan et al., 2017). Both BNNs and ensembles produce a distribution q^* over neural network weights θ; q^* is the Bayesian posterior (or an approximation thereof) for BNNs, and a mixture of point masses obtained by independent training runs for ensembles. Computing the expectation of network outputs over $\theta \sim q^*$ produces predictions, and the corresponding variability of the outputs over θ can be used to quantify uncertainty.

Ensembles typically outperform BNNs at UQ, but they do so at increased computational cost (Abdar et al., 2021). In recent work, Loaiza-Ganem et al. (2025) pointed out that the distribution q^* used by ensembles can actually be interpreted as a Bayesian posterior corresponding to a learned data-dependent prior. In this sense ensembles are BNNs, except the prior is not fixed beforehand as in standard BNNs. Loaiza-Ganem et al. (2025) also argue that using priors which concentrate their mass around the set of maximum-likelihood weights is likely beneficial for UQ and a potential reason behind the good performance of ensembles. Inspired by this connection, we propose last layer empirical Bayes (LLEB) as an intermediate between standard BNNs and ensembles in terms of the strength of the used prior: LLEB is a BNN where the prior is instantiated as a normalizing flow (NFs; Dinh et al., 2015; Rezende & Mohamed, 2015; Durkan et al., 2019) and learned by maximizing the standard evidence lower bound (ELBO) from variational inference (VI; Wainwright & Jordan, 2008; Kingma & Welling, 2014; Rezende et al., 2014; Blei et al., 2017). One key motivation behind LLEB is that by learning the prior through a NF, q^* can place most of its mass around "good values of θ" while retaining diversity in θ, thus hopefully achieving comparable performance to ensembles without the need to train various models. To maintain tractability, we follow the recent trend in BNNs of being Bayesian only over a subset of parameters such as those in the last layer (Lázaro-Gredilla & Figueiras-Vidal, 2010; Kristiadi et al., 2020; Watson et al., 2020; 2021; Harrison et al., 2024; Yang et al., 2024).

Empirically, we find that LLEB performs on par but not significantly nor consistently better than existing UQ approaches of similar computational cost. Our results highlight the promise of empirical Bayes for UQ, and we hope that future work will be able to leverage the ideas behind LLEB to outperform existing UQ methods.

2 Background and Related Work

Setup Throughout this work we will consider a classification setup, where we have access to a dataset $\mathcal{D} = \{(x_i, y_i)\}_i$ of feature-label pairs (x_i, y_i). Here, the likelihood $p(\mathcal{D} \mid \theta)$ is given by $p(\mathcal{D} \mid \theta) = \prod_i p(y_i \mid x_i, \theta)$, where $p(y_i \mid x_i, \theta)$ is the probability assigned by the neural network parameterized by $\theta \in \Theta$ to the label y_i when given the input x_i. We will assume that the likelihood function achieves its maximum, and will denote the set of maximizers as $\Theta^* \subset \Theta$.

Bayesian neural networks and variational inference BNNs begin by specifying a prior π, often as a Gaussian with diagonal covariance. The main object of interest in BNNs is then the corresponding posterior distribution, which is given by $\pi(\theta \mid \mathcal{D}) \propto \pi(\theta) p(\mathcal{D} \mid \theta)$. Unfortunately, computing $\pi(\cdot \mid \mathcal{D})$ and sampling from it are intractable. Various lines of research, which we will shortly summarize, attempt to circumvent this problem by providing a distribution q^* whose goal is to approximate the posterior, i.e. $q^* \approx \pi(\cdot \mid \mathcal{D})$. Once q^* has been obtained, the epistemic uncertainty (Hüllermeier & Waegeman, 2021) associated with predicting the label of a query point x_{n+1} can be quantified through the variability of $p(\cdot \mid x_{n+1}, \theta)$ over $\theta \sim q^*$, and predictions can be made through the predictive distribution,

$$p(\cdot \mid x_{n+1}) \coloneqq \mathbb{E}_{\theta \sim q^*} \left[p(\cdot \mid x_{n+1}, \theta) \right]. \tag{1}$$

One class of methods uses a Laplace approximation, i.e. a second-order Taylor expansion of $\log \pi(\cdot \mid \mathcal{D})$, to obtain q^* (Ritter et al., 2018; Kristiadi et al., 2020; Daxberger et al., 2021; Yang et al., 2024); this results in q^* being a Gaussian approximation of the posterior. Another class of methods uses Markov chain Monte Carlo to approximately sample from $\pi(\cdot \mid \mathcal{D})$ (Welling & Teh, 2011; Chen et al., 2014; Zhang et al., 2020), here the distribution of the chain corresponds to q^*. Gal & Ghahramani (2016) obtain q^* by using dropout (Srivastava et al., 2014).

A final relevant class of procedures to obtain q^* do so through VI (Graves, 2011; Blundell et al., 2015; Louizos & Welling, 2016; 2017; Wu et al., 2019; Osawa et al., 2019; Harrison et al., 2024), i.e. by maximizing the ELBO,

$$\mathrm{ELBO}(q, \pi) \coloneqq \mathbb{E}_{\theta \sim q} \left[\log p(\mathcal{D} \mid \theta) \right] - \mathbb{KL} \left(q \| \pi \right), \tag{2}$$

over $q \in \mathcal{Q}$ for some family of distributions $\mathcal{Q}$. When $\mathcal{Q}$ is flexible enough in the sense that it contains the true posterior, this maximization is well known to yield $q^* = \pi(\cdot \mid \mathcal{D})$. However, most VI-based BNN methods use simple choices of $\mathcal{Q}$ (e.g. Gaussians) due to tractability. Many methods use NFs[1] to instantiate $\mathcal{Q}$ (Rezende & Mohamed, 2015; Kingma et al., 2016), resulting in increased flexibility while keeping the KL term in the ELBO tractable. However, these methods apply VI in the context of variational autoencoders (Kingma & Welling, 2014; Rezende et al., 2014) and not to BNNs because the size of NFs cannot be scaled to the number of parameters in a neural network. Indeed, the large number of parameters in neural networks results in BNNs having to deal with extremely high-dimensional distributions; some works have sought to circumvent this bottleneck by being Bayesian only over the last layer of the network (Kristiadi et al., 2020; Harrison et al., 2024).

Deep ensembles and their connection to Bayesian neural networks Like BNNs, ensembles find a distribution q^* which is also used to quantify uncertainty, and to produce predictions through Equation 1. Ensembles train M separate models through maximum-likelihood, i.e. maximizing $\log p(\mathcal{D} \mid \theta)$, to obtain $\theta^*_m \in \Theta^*$ for $m = 1, \ldots, M$; all these values are different due to the randomness of stochastic optimization – the resulting q^* is then given by $q^*(\theta) = \sum_m \delta_{\theta^*_m}(\theta)$, where $\delta_{\theta^*_m}$ denotes a point mass at θ^*_m. Although ensembles are not typically thought of as Bayesian, Loaiza-Ganem et al. (2025) recently argued they can be understood as performing empirical Bayes, i.e. learning the prior π from data. More specifically, assuming enough capacity, the ELBO in

[1]Recall that NFs define a density q as the density of $f(Z)$, where f is an invertible neural network and Z has a simple distribution such as an isotropic Gaussian.

Equation 2 is maximized over both q and π (rather than just q) by (q^*, π^*) if and only if q^* assigns probability 1 to Θ^* and $q^* = \pi^*$; in this case, the prior π^*, its corresponding posterior $\pi^*(\cdot \mid \mathcal{D})$, and q^* are all equal to each other (see Appendix A for more details). The particular q^* used by ensembles assigns probability 1 to Θ^*, and thus it follows that it can be interpreted as both a learned prior and its corresponding posterior. In short, the main difference between BNNs and ensembles is that BNNs use weak (e.g. Gaussian), fixed priors (or with at most the variance being learnable), whereas ensembles use strong and implicitly learned data-dependent priors.

3 Last Layer Empirical Bayes

There are three main motivations behind our work. First, deep ensembles tend to outperform BNNs at UQ (Abdar et al., 2021), and the empirical Bayes view of ensembles thus suggests that using stronger, data-dependent priors is preferable to using weak, fixed ones. Consequently, we aim to explore explicitly learning the prior. Second, although the empirical Bayes view of ensembles suggests that very strong priors are better than very weak ones, it does not guarantee that stronger is always better. In particular, the prior q^* used by ensembles is extremely strong, and part of our motivation is to use a slightly weaker learned prior which is still strong enough to concentrate mass around Θ^*. Third, ensembles are computationally expensive as they require training M models. Our final motivator is that by explicitly learning q^* once we can avoid training M models. We hope that a model satisfying these motivations might perform similarly to ensembles while being cheaper to train.

With these motivations in mind, we first consider simply maximizing $\mathbb{E}_{\theta \sim q}[\log p(\mathcal{D} \mid \theta)]$ over $q \in \mathcal{Q}$; this will produce the same optimal q^* as maximizing $\text{ELBO}(q, \pi)$ over q and π under a flexible enough π. Furthermore, if $\mathcal{Q}$ is flexible enough, the resulting q^* will assign probability 1 to Θ^*, and could thus be interpreted as both a prior and its corresponding posterior, just like in ensembles. Our goal here is then to choose a $\mathcal{Q}$ which (i) is flexible enough for q^* to concentrate mass around Θ^* while not collapsing onto a point mass (as this would just recover a maximum-likelihood solution), and (ii) results in q^* being more diverse than the mixture of point masses used by ensembles. Specifying $\mathcal{Q}$ as NFs is then very natural since NFs are very flexible, yet their invertibility acts as an implicit regularizer which prevents collapse onto a point mass and promotes some diversity. In summary, we would ideally like to instantiate q_η as a NF parameterized by η and maximize $\mathbb{E}_{\theta \sim q_\eta}[\log p(\mathcal{D} \mid \theta)]$ over η to then treat the resulting q_{η^*} as we would q^* in BNNs or ensembles.

Using q_η as described above would satisfy our first two motivations but would still result in a highly intractable procedure despite not requiring to train M models. The root cause of this intractability is the high dimensionality of the NF, and we thus propose to quantify uncertainty only over a subset of parameters. More precisely, let $\theta = (\theta_{QU}, \theta_{NU})$, where θ_{QU} and θ_{NU} are the parameters over which we do and do not quantify uncertainty, respectively. As a first attempt to address the tractability issues, we then instantiate q_η as a NF on θ_{QU} and maximize $\mathbb{E}_{\theta_{QU} \sim q_\eta}[\log p(\mathcal{D} \mid \theta_{QU}, \theta_{NU})]$ over θ_{NU} and η. We found this end-to-end objective performed well with small classifiers, but that it resulted in unstable and slow optimization when using larger classifiers. As a way to circumvent this issue, we train our model in two steps: we first perform maximum likelihood by maximizing $\log p(\mathcal{D} \mid \theta_{QU}, \theta_{NU})$ to obtain θ^*_{QU} and θ^*_{NU}, and we then discard θ^*_{QU} and maximize $\mathbb{E}_{\theta_{QU} \sim q_\eta}[\log p(\mathcal{D} \mid \theta_{QU}, \theta^*_{NU})]$ over η; we found this strategy to be faster and much more stable for larger classifiers. Note that here q^* is now formally given by $q^*(\theta_{QU}, \theta_{NU}) = \delta_{\theta^*_{NU}}(\theta_{NU}) q_{\eta^*}(\theta_{QU})$.

In practice we chose to set θ_{UQ} as the weights of the last layer of the classifier. The resulting methods (both end-to-end and two-step training), which we call *last layer empirical Bayes*, satisfy all our motivations: q^* is flexible and explicitly learned, the invertibility of the flow prevents collapse onto point masses and encourages some diversity, and since the NF is relatively low-dimensional, training it does not incur significant computational overhead as compared to just maximizing the likelihood. We highlight that LLEB is certainly not the only way to satisfy our starting motivations; we include in Appendix B various alternatives to LLEB which we considered but found to empirically underperform LLEB.

4 Experiments

Setup We conduct experiments on two pairs of datasets, MNIST (LeCun et al., 1998) & Fashion-MNIST (Xiao et al., 2017), and CIFAR-10 (Krizhevsky & Hinton, 2009) & SVHN (Netzer et al.,

Table 1: Results on MNIST & Fashion-MNIST. The top and bottom parts show single and ensembled ($M = 5$) models, respectively. For each metric, the best results within models of comparable computational cost are bolded (only best mean values are bolded).

	Train/Test: MNIST, OOD: Fashion-MNIST			Train/Test: Fashion-MNIST, OOD: MNIST		
Method	Acc. (↑)	ECE (↓)	AUC (↑)	Acc. (↑)	ECE (↓)	AUC (↑)
Default	98.02 ± 0.05	$\mathbf{0.00 \pm 0.00}$	-	88.02 ± 0.10	$\mathbf{0.01 \pm 0.00}$	-
LLL	98.02 ± 0.05	0.75 ± 0.00	$\mathbf{0.96 \pm 0.00}$	88.02 ± 0.10	0.66 ± 0.00	$\mathbf{0.82 \pm 0.01}$
MCD	$\mathbf{98.50 \pm 0.05}$	0.01 ± 0.00	0.91 ± 0.00	$\mathbf{88.47 \pm 0.05}$	0.02 ± 0.00	0.75 ± 0.03
LLEB (ours)	97.74 ± 0.24	$\mathbf{0.00 \pm 0.00}$	0.95 ± 0.01	87.83 ± 0.37	$\mathbf{0.01 \pm 0.00}$	0.72 ± 0.03
Default ($M = 5$)	98.26 ± 0.02	$\mathbf{0.01 \pm 0.00}$	$\mathbf{0.97 \pm 0.00}$	88.71 ± 0.09	$\mathbf{0.02 \pm 0.00}$	0.84 ± 0.01
LLL ($M = 5$)	98.26 ± 0.02	0.76 ± 0.00	0.96 ± 0.00	88.71 ± 0.09	0.67 ± 0.00	0.87 ± 0.01
MCD ($M = 5$)	$\mathbf{98.69} \pm 0.02$	0.02 ± 0.00	0.95 ± 0.00	89.34 ± 0.08	0.04 ± 0.00	$\mathbf{0.89 \pm 0.00}$
LLEB ($M = 5$, ours)	98.30 ± 0.08	$\mathbf{0.01 \pm 0.00}$	$\mathbf{0.97 \pm 0.00}$	$\mathbf{89.44 \pm 0.16}$	0.03 ± 0.00	$\mathbf{0.89 \pm 0.01}$

Table 2: Results on CIFAR-10 & SVHN, metrics and methods are identical to those in Table 1.

	Train/Test: CIFAR-10, OOD: SVHN			Train/Test: SVHN, OOD: CIFAR-10		
Method	Acc. (↑)	ECE (↓)	AUC (↑)	Acc. (↑)	ECE (↓)	AUC (↑)
Default	92.82 ± 0.09	$\mathbf{0.05 \pm 0.00}$	-	$\mathbf{95.26 \pm 0.03}$	0.03 ± 0.00	-
LLL	92.82 ± 0.09	0.70 ± 0.00	$\mathbf{0.94 \pm 0.01}$	$\mathbf{95.26 \pm 0.03}$	0.73 ± 0.00	$\mathbf{0.92 \pm 0.00}$
MCD	92.29 ± 0.09	0.10 ± 0.01	0.89 ± 0.02	95.11 ± 0.05	0.09 ± 0.01	0.89 ± 0.00
LLEB (ours)	$\mathbf{92.85 \pm 0.09}$	0.06 ± 0.00	$\mathbf{0.94 \pm 0.01}$	95.23 ± 0.03	$\mathbf{0.02 \pm 0.01}$	0.86 ± 0.01
Default ($M = 5$)	$\mathbf{94.82 \pm 0.01}$	$\mathbf{0.01 \pm 0.00}$	0.91 ± 0.01	$\mathbf{96.55 \pm 0.03}$	$\mathbf{0.01 \pm 0.00}$	0.97 ± 0.00
LLL ($M = 5$)	$\mathbf{94.82 \pm 0.01}$	0.73 ± 0.00	0.90 ± 0.01	$\mathbf{96.55 \pm 0.03}$	0.74 ± 0.00	0.97 ± 0.00
MCD ($M = 5$)	94.72 ± 0.04	0.12 ± 0.00	0.93 ± 0.01	96.54 ± 0.02	0.11 ± 0.00	$\mathbf{0.98 \pm 0.00}$
LLEB ($M = 5$, ours)	94.78 ± 0.01	$\mathbf{0.01 \pm 0.00}$	$\mathbf{0.95 \pm 0.01}$	96.52 ± 0.03	$\mathbf{0.01 \pm 0.00}$	$\mathbf{0.98 \pm 0.00}$

2011). For each pair, we use one dataset for the train and test sets, and the other as an out-of-distribution (OOD) set. For each pair we fix an architecture and compare LLEB against: the default network, last layer Laplace approximation (LLL; Daxberger et al., 2021), and Monte Carlo dropout (MCD; Gal & Ghahramani, 2016); all these baselines have comparable computational costs to LLEB. We also compare ensembles of LLEB models against standard ensembles (Lakshminarayanan et al., 2017) and ensembled versions of all the aforementioned baselines; once again all these comparisons are fair from a perspective of computational cost. See Appendix C for more information on the implementation details; our code is available at `https://github.com/layer6ai-labs/last_layer_empirical_bayes`.

Metrics We report the accuracy (Acc.) and the expected calibration error (ECE) over the test set; the latter measures how well models quantify aleatoric uncertainty (Hüllermeier & Waegeman, 2021). For every test and OOD point x we also compute $\sum_y \text{var}_{\theta \sim q^*}[p(y \mid x, \theta)]$ to quantify epistemic uncertainty; to evaluate how well models quantify epistemic uncertainty, we compute the area under the receiver operating characteristic curve (AUC) obtained when using this metric to classify between in- and out-of-distribution data (with large values corresponding to OOD). These metrics, along with standard errors across 5 random seeds, are shown in Table 1 and Table 2.

Results LLEB underperforms ensembles despite its ambitious motivation being to achieve similar results, nevertheless we stress that this comparison favours ensembles since they are much more costly to train. Although LLEB outperforms baselines of comparable computational cost in a few tasks and metrics, it does not do so consistently; this holds true both for single models and when ensembling. We see LLEB performing on par with the best existing baselines as highlighting the promise in its empirical Bayes motivation, yet we also believe that LLEB not outperforming these baselines is likely a consequence of the choices we made for tractability.

5 Conclusion

In this work we argued that maximizing $\mathbb{E}_{\theta \sim q}[\log p(\mathcal{D} \mid \theta)]$ over q (with potential regularization) is backed by empirical Bayes as a sensible approach towards UQ. We further proposed LLEB as a way to approximately maximize this objective while retaining tractability. Although LLEB has decent performance, it does not significantly outperform other UQ methods; we hypothesize that this is due to the concessions we made in LLEB for tractability. We hope that future research will manage to improve upon LLEB by better leveraging empirical Bayes to learn q^* in a way that remains tractable and outperforms existing UQ approaches which use simple priors.

ACKNOWLEDGMENTS

We thank Brendan Ross for insightful discussions and for having provided feedback on our manuscript.

REFERENCES

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul W. Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, 2015.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *arXiv:1604.07316*, 2016.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International Conference on Machine Learning*, 2014.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux-effortless Bayesian deep learning. In *Advances in Neural Information Processing Systems*, 2021.

Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation. In *ICLR Workshop Track*, 2015.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. In *Advances in Neural Information Processing Systems*, 2019.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. nflows: normalizing flows in PyTorch, November 2020. URL `https://doi.org/10.5281/zenodo.4296287`.

Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, 2016.

Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*, 2011.

James Harrison, John Willes, and Jasper Snoek. Variational Bayesian last layers. In *International Conference on Learning Representations*, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *International Conference on Machine Learning*, 2015.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, 2016.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being Bayesian, even just a bit, fixes overconfidence in relu networks. In *International Conference on Machine Learning*, 2020.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017.

Miguel Lázaro-Gredilla and Aníbal R Figueiras-Vidal. Marginalized neural network mixtures for large-scale regression. *IEEE transactions on neural networks*, 21(8):1345–1351, 2010.

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

Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42: 60–88, 2017.

Gabriel Loaiza-Ganem, Yuanjun Gao, and John P Cunningham. Maximum entropy flow networks. In *International Conference on Learning Representations*, 2017.

Gabriel Loaiza-Ganem, Valentin Villecroze, and Yixin Wang. Deep ensembles secretly perform empirical bayes. *arXiv:2501.17917*, 2025.

Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix Gaussian posteriors. In *International Conference on Machine Learning*, 2016.

Christos Louizos and Max Welling. Multiplicative normalizing flows for variational Bayesian neural networks. In *International Conference on Machine Learning*, 2017.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. Practical deep learning with Bayesian principles. In *Advances in Neural Information Processing Systems*, 2019.

Apostolos F Psaros, Xuhui Meng, Zongren Zou, Ling Guo, and George Em Karniadakis. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *Journal of Computational Physics*, 477:111902, 2023.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, 2015.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable Laplace approximation for neural networks. In *International Conference on Learning Representations*, 2018.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

Joe Watson, Jihao Andreas Lin, Pascal Klink, and Jan Peters. Neural linear models with functional Gaussian process priors. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2020.

Joe Watson, Jihao Andreas Lin, Pascal Klink, Joni Pajarinen, and Jan Peters. Latent derivative Bayesian last layer networks. In *International Conference on Artificial Intelligence and Statistics*, 2021.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *International Conference on Machine Learning*, 2011.

Anqi Wu, Sebastian Nowozin, Edward Meeds, Richard E Turner, José Miguel Hernández-Lobato, and Alexander L Gaunt. Deterministic variational inference for robust Bayesian neural networks. In *International Conference on Learning Representations*, 2019.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*, 2017.

Adam X Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. Bayesian low-rank adaptation for large language models. In *International Conference on Learning Representations*, 2024.

Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient MCMC for Bayesian deep learning. In *International Conference on Learning Representations*, 2020.

A How Deep Ensembles Secretly Perform Empirical Bayes

The result that optimizing the ELBO in Equation 2 with flexible enough π and q results in $q^* = \pi^* = \pi^*(\cdot \mid \mathcal{D})$ with q^* assigning probability 1 to Θ^* might seem rather surprising at a first glance, since it is not often that a prior matches its posterior in Bayesian inference. This simple result can nonetheless be understood by simply inspecting Equation 2: first, notice that π appears only in the KL term, so that if the learnable prior π is flexible enough, it must be the case that $\pi = q$ holds at optimality. It follows that q must only maximize the first term in the ELBO, $\mathbb{E}_{\theta\sim q}[\log p(\mathcal{D} \mid \theta)]$; we can see by inspection that, when q is flexible enough, q^* must thus assign probability 1 to Θ^*. Additionally, when q is flexible enough, it is well known from variational inference that maximizing the ELBO will result in the variational posterior matching the true posterior, i.e. $q = \pi(\cdot \mid \mathcal{D})$ should also hold at optimality. Combining these observations together, we get that $q^* = \pi^* = \pi^*(\cdot \mid \mathcal{D})$, where these distributions assign probability 1 to Θ^*. We refer the reader to Loaiza-Ganem et al. (2025) for a formal derivation of this result, along with a much more thorough discussion.

B Alternatives to Last Layer Empirical Bayes

As mentioned in Section 3, we tried a few alternatives to LLEB which we now describe. First, we attempted to explicitly regularize the objective to encourage diversity. Since NFs admit density evaluation, it is straightforward to estimate the entropy $\mathbb{H}(q_\eta)$ of q_η. We thus attempted adding a regularizer which encourages maximizing entropy (Loaiza-Ganem et al., 2017), resulting in the objective

$$\mathbb{E}_{\theta_{QU}\sim q_\eta}\left[\log p(\mathcal{D} \mid \theta_{QU}, \theta_{NU})\right] + \lambda \mathbb{H}(q_\eta), \tag{3}$$

which we maximized over θ_{NU} and η, where $\lambda > 0$ is a hyperparameter. We also tried the two-step solution we followed in LLEB, i.e. we first obtained θ^*_{QU} and θ^*_{NU} through maximum-likelihood, we discarded θ^*_{QU}, and we then maximized $\mathbb{E}_{\theta_{QU}\sim q_\eta}[\log p(\mathcal{D} \mid \theta_{QU}, \theta^*_{NU})] + \lambda \mathbb{H}(q_\eta)$ over η. Neither of these approaches improved upon LLEB as described in the main text.

After observing that encouraging higher entropy did not help, we hypothesized that maybe the NF was overly diverse to begin with and that it was not succeeding at placing most of its mass around

Θ^*. We thus tried using $\lambda < 0$ as a way of reducing diversity, but once again this did not improve upon LLEB.

Lastly, we also tried forgoing NFs entirely by replacing them with fully-connected architectures. We found that implementing this change resulted in q^* collapsing onto a point mass, highlighting that the NFs used in LLEB indeed provide implicit regularization against this collapse. Note that since using fully-connected architectures loses density evaluation, we cannot regularize entropy to discourage this collapse.

These failed attempts are the reason why we used NFs and no entropy regularization in LLEB. Although we believe that LLEB can be improved upon by using different distributions which are flexible enough to concentrate mass on Θ^* while remaining as diverse as possible within Θ^*, doing so is not trivial.

C Implementation Details

MNIST & Fashion-MNIST For these two datasets, we use a small convolutional network described in Table 3. For LLEB, we add to the weights of the last layer the output of a Neural Spline Flow (Durkan et al., 2019) implemented using the `nflow` library (Durkan et al., 2020) with parameters described in Table 4. For LLL, we use the implementation from Daxberger et al. (2021) on top of our network. For MCD, we keep the dropout layer active during evaluation. For LLEB, we use the end-to-end training objective with the reparameterization trick to maximize $\mathbb{E}_{\theta_{QU} \sim q_\eta}[\log p(\mathcal{D} \mid \theta_{QU}, \theta_{NU})]$, and use 10 samples from q_η per gradient step. At test time, for all three methods, we also sample 10 times from q^* and average the predictions to approximate the expectation in Equation 1. The other training hyperparameters are given in Table 5.

CIFAR-10 & SVHN For both datasets, we use a ResNet18 (He et al., 2016), where we replace the last linear layer with two linear layers with a same hidden dimension of 50, similarly to the network in Table 3. We use the hyperparameters described in Table 6 to train the default network and ensemble. For LLEB, we use the two-step training procedure and use the frozen weights from the default network and train the flow for 100 epochs and a learning rate of 10^{-5}. For MCD, since there are no dropout layers in ResNet18, we use the same frozen weights but use a new linear head preceded by a dropout layer, which we train until convergence (10 epochs, learning rate of 10^{-5}).

Table 3: Network layers for MNIST and Fashion-MNIST.

Layer	**Parameters**
Conv2d	input_channels: 1, output_channels: 10, kernel_size: 5
MaxPool2d	kernel_size: 2, stride: 2
ReLU	
Conv2d	input_channels: 10, output_channels: 20, kernel_size: 5
Dropout2d	
MaxPool2d	kernel_size: 2, stride: 2
ReLU	
Flatten	
Linear	input_features: 320, output_features: 50
Linear	input_features: 50, output_features: 10

Table 4: Parameters for the Neural Spline Flow.

Parameter	**Value**
Base distribution	Gaussian
Hidden features	100
Number of coupling layers	2
Number of residual blocks	2
Number of bins	11
Tail bound	10
Dropout probability	0
Activations	ReLU

Table 5: Hyperparameters for training on MNIST and Fashion-MNIST.

Parameter	**Value**
Epochs	100
Optimizer	Adam (Kingma & Ba, 2015)
Loss Function	Cross Entropy
Learning Rate	10^{-3}
Weight Decay	10^{-5}
Gradient Clipping	0.1
Batch Size	10^4

Table 6: Hyperparameters for training on CIFAR-10 and SVHN.

Parameter	**Value**
Epochs	100
Optimizer	Adam
Loss Function	Cross Entropy
Learning Rate	5.10^{-4}
Weight Decay	10^{-5}
Gradient Clipping	0.1
Batch Size	128