ELBO, regularized maximum likelihood, and their common one-sample approximation for training stochastic neural networks

Sina Däubener[1] **Simon Damm**[1] **Asja Fischer**[1]

[1]Department of Computer Science, Ruhr University Bochum, Germany.

Abstract

Monte Carlo approximations are central to the training of stochastic neural networks in general, and Bayesian neural networks (BNNs) in particular. We observe that the common one-sample approximation of the standard training objective can be viewed both as maximizing the Evidence Lower Bound (ELBO) *and* as maximizing a regularized log-likelihood of a compound distribution. This latter approach differs from the ELBO only in the order of the logarithm and expectation, and is theoretically grounded in PAC-Bayes theory. We argue theoretically and demonstrate empirically that training with the regularized maximum likelihood increases prediction variance, enhancing performance in misspecified settings, adversarial robustness, and strengthening out-of-distribution (OOD) detection. Our findings help reconcile previous contradictions in the literature by providing a detailed analysis of how training objectives and Monte Carlo sample sizes affect uncertainty quantification in stochastic neural networks.

1 INTRODUCTION

With rapid advances in model performance over the past decade, the deep learning community has increasingly focused on developing methods to quantify model uncertainty—critical for ensuring reliable predictions, particularly in high-stakes applications like healthcare, autonomous systems, and scientific research.

Bayesian neural networks [Neal, 1993, MacKay, 1992] are regularly utilized for this purpose, where the derivation of a posterior distribution over parameters is a main challenge. A central approach for deriving an approximate posterior is through variational inference, where a parametric distribution is fitted to match the true unknown distribution by minimizing the Kullback-Leibler (KL) divergence between the true posterior $p(\theta|\mathcal{D})$ and an approximate parametric posterior distribution $q(\theta)$, i.e.,

$$D_{\mathrm{KL}}[q(\theta) \parallel p(\theta|\mathcal{D})] \ . \tag{1}$$

This divergence is not directly tractable due to the unknown true posterior $p(\theta|\mathcal{D})$, but can be decomposed into the logarithm of the data likelihood $p(\mathcal{D})$ (also called evidence) minus a second term called the *evidence lower bound* (ELBO). Since the evidence $p(\mathcal{D})$ does not depend on the model parameters, one can minimize Eq. (1) by maximizing the ELBO. For a data set consisting of N input-output pairs $\{(x_n, y_n)\}_{n=1}^N$, this objective is given by[1]

$$\mathcal{L}_{\mathrm{VI}} = \sum_{n=1}^{N} \mathbb{E}_{q(\theta)}[\ln p(y_n|x_n,\theta)] - \lambda\, D_{\mathrm{KL}}[q(\theta) \parallel p(\theta)] \ , \tag{VI}$$

where $p(\theta)$ is a prior distribution over the parameters. In general, however, there is no closed-form solution for the expectation term in Eq. (VI), such that Monte Carlo (MC) approximations are applied in practice.[2] That is, for a given input-output pair (x_n, y_n) the expectation is approximated by $\frac{1}{S}\sum_{s=1}^{S} \ln p(y_n|x_n,\theta_s)$, with $\theta_1, \ldots, \theta_S$ being i.i.d. draws from $q(\theta)$. This estimate is known to converge with a convergence rate of $1/\sqrt{S}$ and hence needs a large amount of samples to have a small approximation error.

However, during training the expectation term is usually approximated with *just a single MC sample* $\theta_1 \sim q(\theta)$, resulting in

$$\sum_{n=1}^{N} \ln p(y_n|x_n,\theta_1) - \lambda\, D_{\mathrm{KL}}[q(\theta) \parallel p(\theta)] \ . \tag{baseline}$$

[1]The theory proposes to set $\lambda = 1$. In practice, tempering is often applied, i.e., choosing some $\lambda > 0$. Values $\lambda < 1$, however, are the reason for the "cold posterior" discussion in BNNs as they can increase test set accuracy [Wenzel et al., 2020] but alter training assumptions.

[2]Exceptions exists for simple edge-cases, local linearization [Goulet et al., 2021] or at stationary points [Damm et al., 2023, Velychko et al., 2024, Lücke and Warnken, 2024].

This frequently utilized one-sample MC estimate[3] of Eq. (VI) is *also* the one-sample approximation of a regularized maximum likelihood objective

$$\mathcal{L}_{\mathrm{ML}} = \sum_{n=1}^{N} \ln\left(\mathbb{E}_{q(\theta)}[p(y_n|x_n,\theta)]\right) - \lambda\, D_{\mathrm{KL}}[q(\theta) \parallel p(\theta)] \ . \tag{ML}$$

This objective, $\mathcal{L}_{\mathrm{ML}}$, differs from $\mathcal{L}_{\mathrm{VI}}$, Eq. (VI), only in the first term in the order of expectation and logarithm: $\mathbb{E}_{q(\theta)}[\ln(\cdot)]$ is replaced by $\ln(\mathbb{E}_{q(\theta)}[\cdot])$. Maximizing $\mathcal{L}_{\mathrm{ML}}$ no longer provides a guarantee to reduce the KL divergence between approximate and true posterior distribution.[4] The first term in $\mathcal{L}_{\mathrm{ML}}$ corresponds to the log-likelihood under a compound distribution, where the likelihood is averaged over the mixing distribution $q(\theta)$:

$$p(y|x) = \int p(y|x,\theta)\ q(\theta)\,\mathrm{d}\theta \ . \tag{2}$$

It thus corresponds to the predictive log-loss, which is also used for test-time predictions or evaluation. The second term acts as a regularizer, encouraging the mixing distribution $q(\theta)$ to remain close to a pre-specified distribution $p(\theta)$, as measured by the Kullback–Leibler divergence. In contrast to the ELBO, $p(\theta)$ does not need to be a prior distribution in the Bayesian sense, but can be chosen freely. To summarize, $\mathcal{L}_{\mathrm{ML}}$ minimizes the (regularized) predictive risk (log-loss) of a compound distribution, while $\mathcal{L}_{\mathrm{VI}}$ minimizes the KL divergence to the true model.

The latter objective in Eq. (ML) is no unknown objective. It has been shown to enable tighter generalization bounds following the PAC-Bayesian theory and is known under various names, e.g., as *direct loss minimization* [Sheth and Khardon, 2020, Wei et al., 2021, Wei and Khardon, 2022], PAC^m [Morningstar et al., 2022], or *predictive variational Bayesian inference* [Futami et al., 2022]. Besides the theoretically grounded advantages, $\mathcal{L}_{\mathrm{ML}}$ was shown to behave favorably in practice, especially in the misspecified setting [Morningstar et al., 2022], for (sparse) Gaussian processes [Sheikh et al., 2017, Jankowiak et al., 2020, Wei et al., 2021], and in capturing aleatoric uncertainty [Masegosa, 2020]. On the contrary, for BNNs there exist findings indicating that $\mathcal{L}_{\mathrm{VI}}$ performs favorably [Wei and Khardon, 2022].

However, a thorough understanding of the effects of training stochastic neural networks with $\mathcal{L}_{\mathrm{VI}}$ or $\mathcal{L}_{\mathrm{ML}}$, especially in comparison to their common one-sample approximation is missing so far. We close this gap, by conducting an in-depth analysis of the implications of the changed training objective for the multi-class classification setting. We pay particular attention to the diversity of predictions as these are key for performance and generalization [e.g., Masegosa, 2020, Futami et al., 2022, Ortega et al., 2022]. Besides standard performance measures (NLL, accuracy, ECE) we also investigate the effect of increased prediction variance on adversarial robustness and the capability of detecting out-of-distribution samples.

The presented variance insights also clarify conflicting findings in the literature and resolve their ambiguity, thereby bridging different research branches.

Main Contributions

- We observe that the ELBO ($\mathcal{L}_{\mathrm{VI}}$) and the regularized maximum likelihood objective ($\mathcal{L}_{\mathrm{ML}}$) are indistinguishable when approximating them with only a single Monte Carlo sample, i.e., when $S = 1$, raising the question how the losses and the resulting models differ for $S > 1$, and whether models trained with $S = 1$ are better understood as optimizing $\mathcal{L}_{\mathrm{VI}}$ or $\mathcal{L}_{\mathrm{ML}}$.
- We investigate both losses theoretically and empirically in the multi-class classification setting and demonstrate that training with $\mathcal{L}_{\mathrm{ML}}$ leads to significantly higher diversity in predictions.
- We find that the performance of $\mathcal{L}_{\mathrm{ML}}$ relative to $\mathcal{L}_{\mathrm{VI}}$ and the common one-sample approximation depends on the 'hardness' of the task: for 'hard' tasks and tasks with model-misspecification $\mathcal{L}_{\mathrm{ML}}$ typically outperforms $\mathcal{L}_{\mathrm{VI}}$ and the baseline, while NLL and ECE are typically worse on 'easy' tasks. In addition, $\mathcal{L}_{\mathrm{ML}}$ yields models more robust to OOD inputs and adversarial attacks.
- Finally, we confirm that the commonly used one-sample approximation closely resembles the standard training with $\mathcal{L}_{\mathrm{VI}}$ (which justifies its use for training BNNs).

2 AN ANALYSIS OF THE VARIANCE

We theoretically investigate the difference between the two losses of interest and find the diversity of the prediction to be the key differentiating factor. Consequently, we empirically validate these findings.

2.1 THEORETICAL CONSIDERATIONS

By Jensen's inequality, we see that the first term in Eq. (ML) is at least as large as that of Eq. (VI). That is, ceteris paribus, the KL divergence has a relatively lower influence for Eq. (ML), compared to Eq. (VI), allowing for stronger deviations from the prior. Further analysis shows that we

[3]The one-sample approximation is a standard choice in Bayesian neural network training, e.g., in foundational works such as Auto-Encoding Variational Bayes [Kingma and Welling, 2014], as well as dedicated libraries like Bayesian Torch [Krishnan et al., 2022] and BayesDLL [Kim and Hospedales, 2023].

[4]An exception is the edge case where the Jensen inequality between $\mathbb{E}_{q(\theta)}[\ln(\cdot)]$ and $\ln(\mathbb{E}_{q(\theta)}[\cdot])$ becomes an equality.

can characterize the Jensen gap

$$J(q(\theta)) := \mathcal{L}_{\mathrm{ML}}(q(\theta)) - \mathcal{L}_{\mathrm{VI}}(q(\theta)) \tag{3}$$

by variations in the predictions:

Proposition 1 (Bounds on the Jensen Gap)**.** *Consider a parametrized distribution $p : (\mathcal{X} \times \mathcal{Y}) \times \Theta$, a posterior $q(\theta)$ over the parameter space Θ, and input pairs $(x_n, y_n) \in (\mathcal{X} \times \mathcal{Y})$ for $i \in \{1, \dots, N\}$. Assume that for each n, $p(y_n|x_n, \theta)$ satisfies $p(y_n|x_n, \theta) \in [a_n, 1]$ for $a_n > 0$ with mean $\mu_n = \mathbb{E}_\theta[p(y_n|x_n, \theta)]$, mean absolute deviation $\overline{m}_n = \mathbb{E}_\theta[|p(y_n|x_n, \theta) - \mu_n|]$ and variance $\sigma_n^2 = \mathbb{E}_\theta[(p(y_n|x_n, \theta) - \mu_n)^2]$. Then, the Jensen gap $J(q(\theta))$ between the objectives is bounded by*

$$\sum_{n=1}^{N} \max\left\{\frac{\sigma_n^2}{2}, \delta_{p,n}\right\} \leq J(q(\theta)) \leq \sum_{n=1}^{N} \min\left\{\frac{\sigma_n^2}{2a_n^2}, \frac{\overline{m}_n}{a_n}\right\} \tag{4}$$

where for $p > 1$ and $n \in \{1, \dots, N\}$

$$\delta_{p,n} := \ln\left(\frac{\mathbb{E}_{q(\theta)}[p(y_n|x_n, \theta)]}{\left(\mathbb{E}_{q(\theta)}\left[p(y_n|x_n, \theta)^{\frac{1}{p}}\right]\right)^p}\right) \geq 0 \ . \tag{5}$$

The quantity δ_p, which we refer to as p-compressed expectation spread, is, like the variance, a measure of variability of $p(y_n|x_n, \theta)$. Thus, the *Jensen gap can be characterized by variations in the predictions*: variance or absolute deviation for the upper bound, and variance or δ_p for the lower bound. The difference between $\mathcal{L}_{\mathrm{VI}}$ and $\mathcal{L}_{\mathrm{ML}}$ grows linearly with larger variations but similarly shrinks linearly to zero with smaller variations. Equality between the two objectives is reached if and only if $\forall n : \sigma_n^2 = \mathrm{Var}_{q(\theta)}[p(y_n|x_n, \theta)] = 0$.

The proof is deferred to Appendix A, alongside further explanations on the p-compressed expectation spread δ_p derived from the self-improving AM-GM inequality [Aldaz, 2009]. Note, that the Jensen gap is also investigated in other works, e.g., by Masegosa [2020], which present the lower bound in terms of the prediction variance to the Jensen gap, and by Futami et al. [2021]. A discussion on existing results is given in Appendix A, and an empirical comparison of the different bounds in Figure 5.

Proposition 1 suggests that diversity in the predictions may be the key factor in analyzing the effects of the above described 'log $\mathbb{E}$xchange'. A further inspection of the gradients adds to these findings. The gradients in their S-sample approximation read:

$$\nabla_\theta \mathbb{E} \ln : \quad \frac{1}{S}\sum_{s=1}^{S} \frac{\nabla_\theta \, p(y_n|x_n, \theta_s)}{p(y_n|x_n, \theta_s)} \ , \tag{6}$$

$$\nabla_\theta \ln \mathbb{E} : \quad \frac{1}{S}\sum_{s=1}^{S} \frac{\nabla_\theta \, p(y_n|x_n, \theta_s)}{\frac{1}{S}\sum_{r=1}^{S} p(y_n|x_n, \theta_r)} \ . \tag{7}$$

The main difference between these gradients lies in how $\nabla_\theta \, p(y_n|x_n, \theta_s)$ is scaled. For $\mathcal{L}_{\mathrm{VI}}$, by the likelihood of the observation for each θ_s individually (Eq. (6)); for $\mathcal{L}_{\mathrm{ML}}$, by the *average* likelihood of the observation over all S draws from $q(\theta)$ (Eq. (7)). Suppose that a model θ_s has low confidence for a given sample (x_n, y_n). Regarding $\mathcal{L}_{\mathrm{VI}}$, this strongly impacts the gradient (weighting is inversely proportional to the confidence). On the contrary, because of the averaged predictions in the denominator of the $\mathcal{L}_{\mathrm{ML}}$ gradient, the gradient magnitude from a single model with low-confidence are in comparison down-weighted whenever the overall likelihood of the mixture $\sum_{s=1}^{S} p(y_n|x_n, \theta_s)$ is sufficiently high. This effect is expected to reduce the diversity between individual posterior samples for $\mathcal{L}_{\mathrm{VI}}$, while allowing for more diversity for $\mathcal{L}_{\mathrm{ML}}$ (and the possibility to learn multiple modes in the posterior, as seen in the toy examples in Morningstar et al. [2022]). For the one-sample approximation ($S = 1$), this gradient down-weighting effect is not present and we therefore expect the one-sample approximation to behave more similar to $\mathcal{L}_{\mathrm{VI}}$. Motivated by the theoretical considerations above we proceed to investigate the manifested differences resulting from training with $\mathcal{L}_{\mathrm{VI}}$ vs. $\mathcal{L}_{\mathrm{ML}}$ in practice.

2.2 EXPERIMENTAL ANALYSIS

Before diving into the empirical part, we want to highlight that the change of objectives is in practice done by a simple one-line change of code compared to regular Bayesian neural network training with the ELBO as shown in Listing 1.

Experimental set-up We set the number of MC samples to $S = 5$ for approximating the expectation during training and $\lambda = 1$ (weighting of the KL divergence). We always compare to the 'baseline' ($S = 1$) for which $\mathcal{L}_{\mathrm{VI}}$ and $\mathcal{L}_{\mathrm{ML}}$ are equivalent. At test time all predictions are made based on 100 samples drawn from $q(\theta)$ to approximate $\mathbb{E}_{q(\theta)}[p(y|x, \theta)]$. We validate the findings for different model architectures and hyperparameters (see below). We report means and standard deviation for each experimental setting over 10 random seeds.

```
# multiple forward passes through model (S times)
for s in range(S):
    logit_s, kl = model(x)
    log_p_y_s = dists.Categorical(logit_s).log_prob(
     target)
    log_p_y.append(log_p_y_s)
if args.objective == 'logE': # Eq.(ML)
    E_term = torch.mean(torch.logsumexp(torch.stack(
     log_p_y), 0) - math.log(S))
elif args.objective == 'Elog': # Eq.(VI)
    E_term = torch.mean(torch.stack(log_p_y))
```

Listing 1: **Example implementation** of the 'log $\mathbb{E}$xchange' in the objectives (PyTorch).

Models and Datasets We conduct experiments on five different datasets. Next to the classics in computer vision, i.e.,

MNIST [Deng, 2012] FashionMNIST [Xiao et al., 2017], and CIFAR10 [Krizhevsky et al., 2009], we also use two medical datasets, namely PathMNIST [Kather et al., 2019][5] and DermaMNIST [Tschandl et al., 2018, Codella et al., 2019][6] in the highest resolution from the MedMNIST benchmark dataset [Yang et al., 2021, 2023]. Furthermore, we used four different architectural designs for our stochastic models: A small feedforward network, denoted 'FF', with two hidden layers of size 256 and 128 with ReLu activation functions, and a multivariate normal distribution over the weights with standard normal distributions as our prior. In addition, we use a feedforward network with two hidden layers (width 128) where we model the weight distribution as a matrix variate normal distribution as proposed by Louizos and Welling [2016], denoted 'FF-MVN'. This model type assumes that the learned variance factorizes and therefore reduces the amount of variance parameters from $d_{\text{in}} \times d_{\text{out}}$ to $d_{\text{in}} + d_{\text{out}}$. For CIFAR10 we additionally train a ResNet20 architecture utilizing the code, hyperparameters and training procedure from Krishnan et al. [2022]. Lastly, with 'DINOTopping' we denote a model that uses the above-described 'FF' model on top of the features extracted by DINOv2 [Oquab et al., 2023],[7] where we extract the [CLS] token from the final transformer layer as a global representation of each image. For the experiments, we used AdamW [Loshchilov and Hutter, 2019] with a batch size of 128 and an initial learning rate of 0.001. For more details please see Appendix B.

Analysing the prediction variance As outlined in Section 2.1, the gap between the objectives for the same weight distribution $q(\theta)$ is characterized by the prediction variance. However, because $q(\theta)$ is continuously changing during training, the behavior of the models trained with the different objectives are not directly relatable and hence it is not clear how much and if the prediction variance of the trained model differs. Therefore, we estimate the variance empirically

$$\max_c \left\{ \frac{1}{S} \sum_{s=1}^{S} p(y_c|x_n, \theta_s)^2 - \left(\frac{1}{S} \sum_{s=1}^{S} p(y_c|x_n, \theta_s) \right)^2 \right\}$$

and visualize the results in Figure 1.

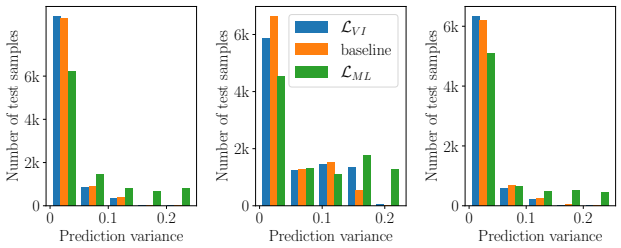
As expected, we observe significantly higher prediction variances for the models trained with $\mathcal{L}_{\text{ML}}$ throughout all datasets and all network designs (full results presented in Table 1). Models trained with baseline and $\mathcal{L}_{\text{ML}}$ show comparable variance.[8]

[5]Released under CC BY 4.0 license.

[6]Released under CC BY-NC 4.0 license.

[7]Released under Apache License 2.0.

[8]Regarding the KL divergence, we observed throughout all experiments that it is lower for $\mathcal{L}_{\text{VI}}$ during training than for the baseline or $\mathcal{L}_{\text{ML}}$ (see argument in Sec. 2.1 and Fig. 6).



(a) MNIST (b) CIFAR10 (c) PathMNIST

Figure 1: **Histogram of test samples binned by the prediction variance.** a) Uses the FF-MVN, b) the ResNet20, and c) the DINOTopping model.

However, high(er) prediction variance per se is not informative about the behavior of ensemble members: Ensemble members can behave similarly for a given input, i.e., giving the same ordering of class labels, or predicting entirely different labels (see Figure 8 in Appendix D for an illustrative examples). To further analyze the variability in prediction, we propose to investigate the dissimilarity score between predictions of single drawn networks as done by Fort et al. [2020]. That is, we measure the dissimilarity between two networks, corresponding to parameters θ_i and θ_j drawn from the learned posterior $q(\theta)$, as the fraction of disagreeing predictions, given by

$$\frac{1}{N} \sum_{n=1}^{N} \mathbb{1}[\arg\max_c p(y_c|x_n, \theta_i) \neq \arg\max_c p(y_c|x_n, \theta_j)] \ .$$

To generate the plot shown in in Figure 2 we draw ten samples θ_i (for each learned posterior $q(\theta)$), i.e., each pixel represents the dissimilarity between the predictions of two distinct parameter draws.

For models trained with $\mathcal{L}_{\text{ML}}$ we observed notably higher function space diversity compared to those from the models trained with $\mathcal{L}_{\text{VI}}$ or the baseline. Regarding MNIST and PathMNIST, the results for $\mathcal{L}_{\text{VI}}$ and the one-sample approximation appear similar, while $\mathcal{L}_{\text{VI}}$ demonstrates higher dissimilarity for CIFAR10. This is in line with Figure 1, where the models trained with $\mathcal{L}_{\text{VI}}$ show slightly higher variance than the baseline.

The higher function space diversity is an interesting property of $\mathcal{L}_{\text{ML}}$-trained models, as it has been found to improve ensemble predictions in many tasks. Amongst others, it has been argued to be the reason for good uncertainty estimates of ensembles [Fort et al., 2020], found to be relevant to bound the PAC-Bayes error under misspecification [Masegosa, 2020], improving uncertainty and OOD detection performance of ensembles [Pagliardini et al., 2023], and the motivation for function space variational inference [Sun et al., 2019, Wang et al., 2019].

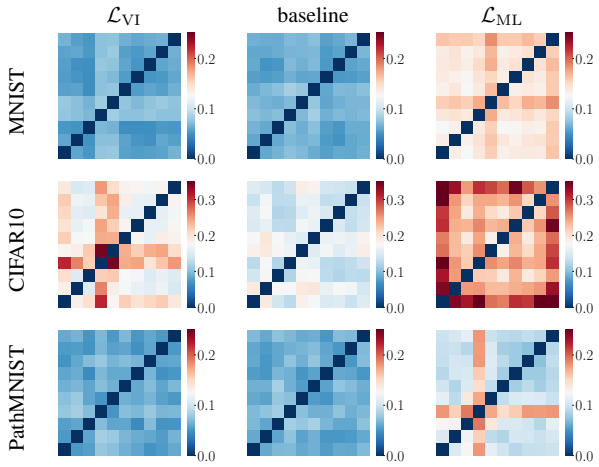


Figure 2: **Dissimilarity in predictions** measured as a fraction of disagreement between predictions of networks based on single draws from $q(\theta)$. Darker red resembles higher amount of disagreement, indicating that more diverse functions are learned.

Analyzing the weight distributions To determine the origins of the differing prediction variance behaviors, we inspect the learned distribution $q(\theta)$ for the models trained on PathMNIST, as the architecture and weight distribution design allow for straightforward analysis. We calculated the Kullback-Leibler divergence between each weights' univariate normal distributions and standard normal distributions, see Figure 3, and observe, that the model trained with $\mathcal{L}_{\text{VI}}$ has the highest amount of 'collapsed' weights, i.e., weights following the prior distribution. Naturally, this finding also translates when comparing the weights' variances[9] resulting from the different objectives. Interestingly, we find that the weight distribution of the baseline and models trained with the $\mathcal{L}_{\text{ML}}$ seem to behave more similarly. Another finding is that the $\mathcal{L}_{\text{ML}}$ trained model has relatively more weights for which the variance is essentially zero (i.e., they behave almost deterministically).

Thus, higher learned variances over the weights seem to correlate with lower prediction variance. We hypothesize that the models trained with $\mathcal{L}_{\text{VI}}$ partly learn to 'disable' high variance connections from contributing to the final prediction, effectively learning a sparser network to better comply with the KL divergence.

3 AN ANALYSIS OF THE EFFECTS OF THE PREDICTION VARIANCE...

Given our finding that the training objectives lead to substantial differences in the prediction variance, this section analyses the effects and consequences of these differences, starting with the classical performance metrics such as accuracy, negative log-likelihood, and expected calibration error.

[9]The mean distribution does not show interesting differences, results are therefore shown in Figure 7 in the Appendix.

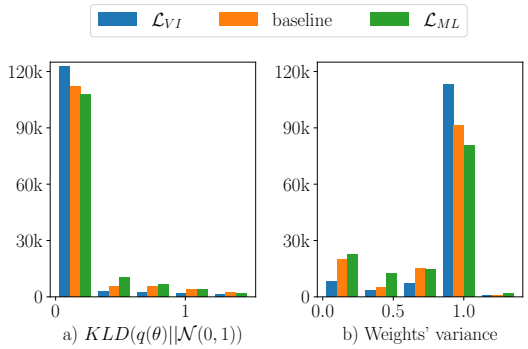


Figure 3: **Histogram of $D_{\text{KL}}[q(\theta)||\mathcal{N}(0,1)]$ and σ^2 for each weight in the trained network.** $\mathcal{L}_{\text{VI}}$ has the most weights which are essentially equal to the prior distribution. Weight distributions for baseline and $\mathcal{L}_{\text{ML}}$ seem to be more similar compared to the $\mathcal{L}_{\text{VI}}$ trained model.

3.1 ...ON ACCURACY, NLL, CALIBRATION ERROR AND PREDICTION CONFIDENCE

The relevant statistics for all objectives, model types, and datasets are presented in Table 1, showing that the overall performance of all inspected models is decent (with FF-architecture on CIFAR10 as an intended exception). For the DINOTopping models, we even reach state-of-the-art results on DermaMNIST and PathMNIST (which justify our setup).

$\mathcal{L}_{\text{ML}}$ is better on 'hard' tasks While accuracy is mostly comparable, we observe a significant increase in accuracy and log-likelihood for models trained with $\mathcal{L}_{\text{ML}}$ for the FF architecture on CIFAR10 and DINOTopping on DermaMNIST. This increase in accuracy can be explained by the difficulty of the task: The small fully connected feedforward network (FF) is clearly unsuited for CIFAR10 (misspecified), while for DermaMNIST only few training samples are available (cf. Table 3) and overall performance is quite low (baseline achieves accuracies below 80%).

As found by Ortega et al. [2022] for general ensembles, we suspect that these accuracy advantages stem from the combination of diverse weak learners (as found in Section 2.2) which lead to better accuracies through error diversification. This finding resonates with that of Morningstar et al. [2022], who found that the $\mathcal{L}_{\text{ML}}$ objective (termed PAC^m in their work) performs better in case of misspecification, i.e., when the true data generating distribution cannot be matched by any model in the single parameter setting ($\nexists\, \theta \in \Theta : p(y|x,\theta) = p_{\text{data}}(y|x)$). Experimen-

Dataset	Arch	Obj.	Accuracy in % ↑	NLL ↓	Avg. pred conf in %	Avg. variance	ECE ↓
MNIST	FF	$\mathcal{L}_{\mathrm{VI}}$	$97.94_{\pm 0.04}$	$0.081_{\pm 0.001}$	$95.41_{\pm 0.06}$	$0.012_{\pm 0.000}$	$0.025_{\pm 0.001}$
		baseline	$98.12_{\pm 0.06}$	$\mathbf{0.072}_{\pm \mathbf{0.001}}$	$95.97_{\pm 0.09}$	$0.013_{\pm 0.000}$	$\mathbf{0.022}_{\pm 0.001}$
		$\mathcal{L}_{\mathrm{ML}}$	$98.19_{\pm 0.07}$	$0.075_{\pm 0.001}$	$95.49_{\pm 0.06}$	$0.030_{\pm 0.000}$	$0.027_{\pm 0.000}$
	FF-MVN	$\mathcal{L}_{\mathrm{VI}}$	$97.65_{\pm 0.06}$	$\mathbf{0.099}_{\pm \mathbf{0.001}}$	$94.22_{\pm 0.04}$	$0.014_{\pm 0.000}$	$0.034_{\pm 0.001}$
		baseline	$97.42_{\pm 0.04}$	$0.106_{\pm 0.003}$	$93.88_{\pm 0.08}$	$0.015_{\pm 0.000}$	$0.035_{\pm 0.001}$
		$\mathcal{L}_{\mathrm{ML}}$	$97.46_{\pm 0.09}$	$0.118_{\pm 0.001}$	$92.41_{\pm 0.13}$	$0.046_{\pm 0.001}$	$0.051_{\pm 0.002}$
FashionMNIST	FF	$\mathcal{L}_{\mathrm{VI}}$	$87.18_{\pm 0.21}$	$0.358_{\pm 0.002}$	$83.66_{\pm 0.21}$	$0.016_{\pm 0.000}$	$0.035_{\pm 0.002}$
		baseline	$87.87_{\pm 0.09}$	$0.340_{\pm 0.002}$	$84.69_{\pm 0.20}$	$0.016_{\pm 0.000}$	$0.032_{\pm 0.002}$
		$\mathcal{L}_{\mathrm{ML}}$	$\mathbf{88.33}_{\pm \mathbf{0.10}}$	$\mathbf{0.328}_{\pm \mathbf{0.001}}$	$84.97_{\pm 0.07}$	$0.049_{\pm 0.001}$	$0.034_{\pm 0.000}$
	FF-MVN	$\mathcal{L}_{\mathrm{VI}}$	$85.94_{\pm 0.28}$	$0.393_{\pm 0.003}$	$82.81_{\pm 0.11}$	$0.014_{\pm 0.000}$	$0.032_{\pm 0.004}$
		baseline	$85.73_{\pm 0.17}$	$0.398_{\pm 0.002}$	$82.55_{\pm 0.25}$	$0.015_{\pm 0.000}$	$0.032_{\pm 0.003}$
		$\mathcal{L}_{\mathrm{ML}}$	$\mathbf{86.53}_{\pm \mathbf{0.11}}$	$\mathbf{0.382}_{\pm \mathbf{0.003}}$	$82.41_{\pm 0.26}$	$0.050_{\pm 0.001}$	$0.041_{\pm 0.002}$
CIFAR10	ResNet	$\mathcal{L}_{\mathrm{VI}}$	$89.95_{\pm 0.37}$	$0.314_{\pm 0.009}$	$85.88_{\pm 0.40}$	$0.049_{\pm 0.002}$	$0.041_{\pm 0.002}$
		baseline	$89.59_{\pm 0.24}$	$0.312_{\pm 0.005}$	$87.63_{\pm 0.16}$	$0.036_{\pm 0.001}$	$\mathbf{0.021}_{\pm 0.002}$
		$\mathcal{L}_{\mathrm{ML}}$	$89.48_{\pm 0.44}$	$0.347_{\pm 0.009}$	$82.94_{\pm 0.39}$	$0.077_{\pm 0.002}$	$0.065_{\pm 0.005}$
	FF	$\mathcal{L}_{\mathrm{VI}}$	$39.92_{\pm 0.65}$	$1.683_{\pm 0.008}$	$33.30_{\pm 0.47}$	$0.013_{\pm 0.000}$	$0.066_{\pm 0.002}$
		baseline	$40.58_{\pm 1.00}$	$1.655_{\pm 0.014}$	$34.80_{\pm 0.64}$	$0.013_{\pm 0.001}$	$0.058_{\pm 0.005}$
		$\mathcal{L}_{\mathrm{ML}}$	$\mathbf{45.37}_{\pm \mathbf{0.20}}$	$\mathbf{1.550}_{\pm \mathbf{0.004}}$	$39.15_{\pm 0.22}$	$0.079_{\pm 0.002}$	$0.062_{\pm 0.002}$
DermaMNIST	DINOTopping	$\mathcal{L}_{\mathrm{VI}}$	$77.58_{\pm 1.02}$	$0.617_{\pm 0.015}$	$72.68_{\pm 2.02}$	$0.021_{\pm 0.001}$	$0.053_{\pm 0.018}$
		baseline	$79.11_{\pm 0.89}$	$0.575_{\pm 0.014}$	$74.02_{\pm 1.01}$	$0.024_{\pm 0.002}$	$0.052_{\pm 0.011}$
		$\mathcal{L}_{\mathrm{ML}}$	$\mathbf{81.77}_{\pm \mathbf{0.43}}$	$\mathbf{0.515}_{\pm \mathbf{0.007}}$	$76.92_{\pm 0.70}$	$0.089_{\pm 0.004}$	$0.050_{\pm 0.011}$
PathMNIST	DINOTopping	$\mathcal{L}_{\mathrm{VI}}$	$94.48_{\pm 0.38}$	$0.151_{\pm 0.009}$	$93.76_{\pm 0.38}$	$0.015_{\pm 0.001}$	$0.007_{\pm 0.002}$
		baseline	$94.43_{\pm 0.11}$	$0.152_{\pm 0.004}$	$93.92_{\pm 0.29}$	$0.016_{\pm 0.001}$	$0.007_{\pm 0.002}$
		$\mathcal{L}_{\mathrm{ML}}$	$94.44_{\pm 0.32}$	$0.166_{\pm 0.006}$	$92.88_{\pm 0.19}$	$0.043_{\pm 0.001}$	$0.016_{\pm 0.002}$

Table 1: **Accuracy, negative log-likelihood (NLL), average prediction confidence, average prediction variance, and expected calibration error (ECE)** for different datasets and model types on the respective test sets. Previous SOTA accuracy for DermaMNIST was 76.8% (with Google AutoML Vision), and 91.1% for PathMNIST (with ResNet-50 (28)), see Table 3 in Yang et al. [2023]. Bold indicates the best performance in terms of accuracy, NLL or ECE whenever the effect size exceeds two standard deviations ($\geq 2\sigma_{\max}$).

tally they demonstrate some benefits of the $\mathcal{L}_{\mathrm{ML}}$ objective for neural networks when using an explicitly ill-defined regression problem[10] and reached comparable accuracies to $\mathcal{L}_{\mathrm{VI}}$ in classification tasks, where the prior was named as the source of misspecification. With our experiments on FF on CIFAR10 we contribute to their finding by adding an instance to the list of misspecifications, namely a misspecification in form an unsuitable network architecture, where the accuracy benefits from using $\mathcal{L}_{\mathrm{ML}}$. Furthermore, the $\mathcal{L}_{\mathrm{ML}}$ objective seems to be beneficial for difficult classification tasks (thinking of DermaMNIST), which can also be regarded as another form of misspecification.

High prediction variance can also hurt Throughout all experiments, we observe—in line with the ideas outlined in Section 2.1—that the average prediction variance is highest for models trained with $\mathcal{L}_{\mathrm{ML}}$. Models trained with the one sample approximation and $\mathcal{L}_{\mathrm{VI}}$ typically show similar prediction variances. In setups where the increased prediction variance is not beneficial, especially when the overall accuracy is already high, it reduces the average prediction confidence. In turn, this negatively impacts the negative log-likelihood as well as the expected calibration error, see the results for MNIST, CIFAR10 with ResNet20 or PathMNIST. This resonates well with the findings of Wei and Khardon [2022], who found that models trained with $\mathcal{L}_{\mathrm{ML}}$ usually get worse negative log-likelihood scores—which at first glance contradict the positive findings reported for example by Morningstar et al. [2022], Futami et al. [2022], and Masegosa [2020].

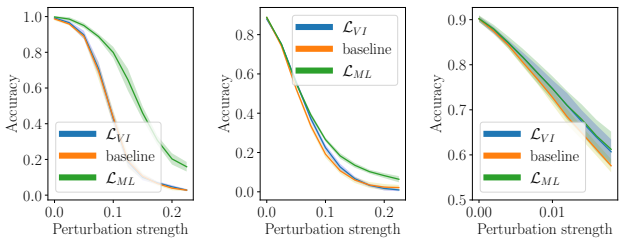
While Wei and Khardon [2022] try to explain these finding with learning dynamics, i.e. by $\mathcal{L}_{\mathrm{ML}}$ getting stuck in bad local minima (a hypothesis they falsified themselves), this behavior is expected as the increased prediction variance naturally reduces the negative log-likelihood in settings where already highly accurate and confident predictions are made. This is because the higher diversity between single ensemble members reduces the model confidence and therefore also the negative log-likelihood—in line with the findings from Jeffares et al. [2023] and Abe et al. [2023] that argue that artificially increasing prediction diversity during

[10] They used the upper half of images as inputs and tried to predict independently the pixel values for the lower half of the images. Because the predictions happen independently but pixel values in images are certainly correlated, it is in the misspecified regime.

training of ensembles can in fact be counterproductive. Interestingly though, we a) get comparable test accuracies (negative effects seem to be limited to NLL and ECE), b) do not directly optimize for increased prediction variance, and c) have the very same setup and only uni-modal normal distributions over the weights and still observe higher function space diversity with $\mathcal{L}_{\text{ML}}$. In addition, we found that training with the baseline typically leads to the best calibrated models.

3.2 ...ON ADVERSARIAL ROBUSTNESS

Recent work by Däubener and Fischer [2022] suggests, that higher prediction variance can have a positive effect on the adversarial robustness of models, which we test in this subsection. For this we attacked the FF-MVN network on MNIST, the FF network on FashionMNIST and the ResNet20 architecture on CIFAR10 with strong attacks, namely with the projected gradient descent method [Madry et al., 2018], which iteratively conducts fast gradient sign method [FGSM, Goodfellow et al., 2015] updates with a smaller step size than the allowed maximal perturbation size. We used 10 iterations and 10 samples per approximation of the gradient. This leads to 100 sampled θ in total per data point. We used the l_∞-norm to quantify the maximal allowed perturbation which we gradually increased from 0 to 0.25. For the models trained on CIFAR10, we calculated adversarial examples with FGSM where we estimated each gradient based on 10 samples of θ for computational reasons. Figure 4 shows the accuracies under adversarial attacks for the models optimized with $\mathcal{L}_{\text{VI}}$, $\mathcal{L}_{\text{ML}}$, and the baseline.



(a) MNIST (b) FMNIST (c) CIFAR10

Figure 4: **Accuracy under adversarial attack with an increasing amount of allowed perturbation.** We report the mean and standard deviation (shaded area) calculated on 10 independently trained and attacked models. a) uses the FF-MVN, b) the FF, and c) the ResNet architecture.

We see, that the adversarial accuracies of the baseline and the $\mathcal{L}_{\text{VI}}$ trained models are lower than for the $\mathcal{L}_{\text{ML}}$ trained model on MNIST and FashionMNIST. This effect is not directly observable for the models trained on CIFAR10.

3.3 ...ON OUT-OF-DISTRIBUTION DETECTION

Lastly, this subsection investigates how capable the models trained with $\mathcal{L}_{\text{ML}}, \mathcal{L}_{\text{VI}}$ and the baseline are to detect out-of-distribution data. To create realistic OOD samples we utilize the benchmark image corruptions by Michaelis et al. [2019]. We take the CIFAR10 test set and generate 75 OOD data sets: for each of the 15 different corruption styles we generated 5 corrupted data sets with increasing severity (see Figure 9 in the Appendix for example images). Next, we let all models predict all samples in these test sets and also for the benign test data set. In addition, we compute the entropy of the predictive distribution (resulting from 100 draws from $q(\theta)$) for each example, as it quantifies the uncertainty in the model's output distribution over the classes:

$$\mathcal{H}(p(y|x)) = -\sum_c p(y_c|x) \ln \left(p(y_c|x)\right)$$

High entropy reflects uncertainty or lack of confidence, which is ideally elevated for OOD inputs, while entropy should be comparably lower on in-distribution data. Thus, it can serve as an effective score for OOD detection. Based on the computed entropy values, the AUROC for distinguishing test from OOD data is calculated, yielding 75 AUROC scores. Based on these values the AUROC for discriminating between test and OOD data set is calculated, which results in 75 AUROC values. Each experiment is repeated 10 times. Because of the same initialization, we conducted a pairwise Wilcoxon rank-sum test with significance level $\alpha = 0.05$ to compare the AUROC values against each other in Table 2.

	$\mathcal{L}_{\text{VI}}$	baseline	$\mathcal{L}_{\text{ML}}$	Avg. Acc	Avg. AUROC
$\mathcal{L}_{\text{VI}}$	×	18	0	0.3942	0.8806
baseline	0	×	0	0.3923	0.8723
$\mathcal{L}_{\text{ML}}$	27	31	×	0.3945	0.8970

Table 2: **The $\mathcal{L}_{\text{ML}}$ objective leads to models more capable of detecting corrupted test instances**. The first block reports the number of successful pairwise Wilcoxon rank-sum tests based on the AUROC values for discriminating between test and OOD samples. The pairwise tests compare if the row objective leads to a significantly higher AUROC than the column objective with entropy as the score function. The total number of tests is 75. Example interpretation for the bottom left entry: In 27 out of all 75 cases (i.e., 36%) the $\mathcal{L}_{\text{ML}}$ objective significantly outperforms the $\mathcal{L}_{\text{VI}}$ objective (while $\mathcal{L}_{\text{VI}}$ never outperformed $\mathcal{L}_{\text{ML}}$). The last two columns display the average accuracy over all seeds and corruptions, and the average AUROC.

Table 2 shows that models trained with the $\mathcal{L}_{\text{ML}}$ objective lead to significantly higher AUROC values in 36% of the OOD detection tasks when compared to models trained with the other objectives. The average accuracy over all OOD datasets is similar for all models, while the average AUROC

mirrors the results of the hypotheses tests, where the $\mathcal{L}_{\mathrm{ML}}$ trained models lead to the highest average value. In this context, we see that $\mathcal{L}_{\mathrm{VI}}$ performs better than the baseline (which is not the case in our other experiments).

4 RELATED WORK

$\mathcal{L}_{\mathrm{ML}}$ loss for neural networks Several works derive $\mathcal{L}_{\mathrm{ML}}$ from different backgrounds. For example, Morningstar et al. [2022] motivate the derivation from the distinction between the predictive risk $\mathcal{P}(q) = -\mathbb{E}_{\nu(X)}\left[\ln \mathbb{E}_{q(\Theta)}[p(x|\Theta)]\right]$ (in this work termed $\mathcal{L}_{\mathrm{ML}}$) and the inferential risk $\mathcal{R}(q) = -\mathbb{E}_{\nu(X)}\left[\mathbb{E}_{q(\Theta)}[\ln p(x|\Theta)]\right]$ (here denoted $\mathcal{L}_{\mathrm{VI}}$). Building on Masegosa [2020], who found that in the case of model misspecifications minimizing the latter is not a tight bound for the predictive risk, Morningstar et al. [2022] leverage an expectation approximation trick following Burda et al. [2016] to derive PAC-Bayesian like guarantees for their PAC^m-bound, which is identical to $\mathcal{L}_{\mathrm{ML}}$ in their numerical approximation. However, their derived bound is vacuous for any fixed number of samples [Morningstar et al., 2022, Appendix B.2] and, therefore, can only serve as a theoretical motivation for the $\mathcal{L}_{\mathrm{ML}}$ objective.

Wei and Khardon [2022] examine $\mathcal{L}_{\mathrm{ML}}$ (*direct loss minimization* with an additional regularization term as they term it) empirically for BNNs and found that models trained with $\mathcal{L}_{\mathrm{ML}}$ perform and generalize worse than their counterparts trained with $\mathcal{L}_{\mathrm{VI}}$. That is, models trained with $\mathcal{L}_{\mathrm{VI}}$ get better negative log-likelihoods across all classification tasks and models they tested, and they hypothesize that this is due to optimization difficulties or overfitting. We confirm this finding on 'easy' classification tasks with sufficiently much training data (although test accuracy seems not to be affected). In contrast, we find that $\mathcal{L}_{\mathrm{ML}}$ is indeed outperforming the ELBO $\mathcal{L}_{\mathrm{VI}}$ on misspecified and challenging tasks (FF-CIFAR10 and DermaMNIST, respectively).

Dusenberry et al. [2020] briefly empirically evaluate the impact of exchanging $\ln$ and $\mathbb{E}$ during training without KL regularization (here termed negative log-marginal-likelihood or mixture NLL). They find that models trained with $\mathcal{L}_{\mathrm{ML}}$ on CIFAR10 result in the worst test set performance in terms of expected calibration error, log-likelihood, and accuracy. They hypothesize that for "misspecified models such as overparametrized neural networks, training a looser bound on the log-likelihood leads to improved predictive performance."

Maximizing variances during neural network training Other works explicitly use methods for enhancing variances to boost prediction performance. For example, based on the derivation of a second-order PAC-Bayes bound, Masegosa [2020] propose to include the prediction variance into the ensemble learning objective, whereas Futami et al. [2021] boost the variances between losses in their approach. Another interesting work was conducted by Ortega et al. [2022] who investigated the interplay between generalization performance and diversity for neural network ensembles. Their main theorem gives insights into what drives ensemble diversity which is a) uncorrelated ensemble members and b) different predictions across models and data samples. Interestingly, they find that the relation between generalization and diversity is not present when operating in the "interpolation regime" for ResNet architectures on CIFAR10, where empirical errors are close to zero. On the contrary, it has been shown that artificially inflating diversity of neural ensembles does not generalize well and actually degrades performance Jeffares et al. [2023], Abe et al. [2023].

5 DISCUSSION AND CONCLUSION

This work takes a closer look at the ELBO used in the variational training of Bayesian neural networks and at how this objective is approximated in practice. The commonly used one-sample approximation of the expectation term in the ELBO, Eq. (VI), can be reinterpreted as the log-likelihood of a compound density model (a fact that only a subgroup of researchers in the PAC-Bayesian domain seem to be aware of).

This implies, that for $S = 1$ the trained stochastic model can *either* be seen as a Bayesian neural network where we try to approximate the true but unknown posterior, *or* as a compound density model where we maximize the (regularized) log-likelihood of the parameters of the mixing distribution. In practice, the difference between these two objectives becomes evident when optimizing with more than one Monte Carlo sample ($S > 1$) which has theoretical and practical implications, specifically with regards to the variance of the model predictions. More precisely, we present a simple proposition indicating that maximizing the ELBO leads to models with lower prediction variance than training with the likelihood-based $\mathcal{L}_{\mathrm{ML}}$ objective. This is verified throughout extensive experiments, where we find that models trained with $\mathcal{L}_{\mathrm{ML}}$ lead to comparable accuracy, increased prediction variance, and increased function space diversity compared to identically initialized models trained with the ELBO or the one-sample approximation (baseline).

The aforementioned properties are linked to model robustness concerning adversarial examples and OOD detection performance, which we also empirically find in our paper. However, encouraging function space diversity for networks that are capable of making highly confident correct predictions for a given task naturally leads to a degradation of prediction confidence and therefore also of the negative log-likelihood. Hence, the findings of Wei and Khardon [2022] are not discrediting the performance of the $\mathcal{L}_{\mathrm{ML}}$, but give credit to Jeffares et al. [2023], Abe et al. [2023], who state that in this particular setting enhancing diversity between ensemble members is not advantageous.

In contrast, we see that enhancing diversity helps when using a poorly suited model architecture—such as in our experiments with the FF on CIFAR10 experiments—or when tackling a challenging task, as with DermaMNIST. In such cases, $\mathcal{L}_{\mathrm{ML}}$ performs favorably compared to the baseline and $\mathcal{L}_{\mathrm{VI}}$. This behavior is resembling the idea of boosting where combing weak learners can yield stronger overall performance when their errors are sufficiently diverse. In this context, we add another misspecification setting (in which the $\mathcal{L}_{\mathrm{ML}}$ objective is beneficial) to the findings of Morningstar et al. [2022], which is a (poorly) suited model architecture itself (next to a suitable prior and likelihood definition). Another contribution of this paper is that our investigation of the gradients in Section 2.2 can explain Morningstar et al. [2022]'s toy regression findings, where $\mathcal{L}_{\mathrm{ML}}$ trained models can reproduce multi-modal predictive distributions and are more robust to outliers: Because of the implicitly encouraged high function space diversity, the averaged likelihood for an outlier is enhanced and thereby reduces the impact of its gradient direction to improve the model on this particular sample. At the same time, the same inner workings allow the model to explore and learn multimodal distributions, as the impact of some wrong predictions is not dominating the gradient direction when training with the $\mathcal{L}_{\mathrm{ML}}$ objective.

Lastly, our analysis verifies the common practice of a one-sample-approximation (baseline) as a good approximation to training with the ELBO ($\mathcal{L}_{\mathrm{VI}}$), as it reproduces similar train and test behavior as the several-sample-approximation (though the weight distributions are more alike to models trained with $\mathcal{L}_{\mathrm{ML}}$).

The presented results show that the way to train stochastic neural networks should depend on the characteristics of the problem itself: When large prediction variance is advantageous to remedy misspecifications, tackling a hard classifcation task with little available data or increase robustness against adversarial or OOD inputs (and no Bayesian interpretation is needed), training a compound density model with the regularized maximum-likelihood objective $\mathcal{L}_{\mathrm{ML}}$ with $S > 1$ can indeed be a capable alternative to the ELBO.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2092 CASA – 390781972 and under project 464104047, 'On the Convergence of Variational Deep Learning to Sums of Entropies', within the priority program 'Theoretical Foundations of Deep Learning' (SPP 2298). Moreover, we acknowledge funding by the Ministry of Culture and Science of Northrhine-Westphalia as part of the Lamarr Fellow Network.

References

Taiga Abe, E Kelly Buchanan, Geoff Pleiss, and John P Cunningham. Pathologies of predictive diversity in deep ensembles. *arXiv preprint arXiv:2302.00704*, 2023.

J. M. Aldaz. Self-improvement of the inequality between arithmetic and geometric means. *Journal of Mathematical Inequalities*, 3, 2009.

Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance Weighted Autoencoders. In *4th International Conference on Learning Representations*, 2016. URL http://arxiv.org/abs/1509.00519.

Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019.

Simon Damm, Dennis Forster, Dmytro Velychko, Zhenwen Dai, Asja Fischer, and Jörg Lücke. The ELBO of Variational Autoencoders Converges to a Sum of Entropies. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, 2023. URL https://proceedings.mlr.press/v206/damm23a.html.

Sina Däubener and Asja Fischer. How sampling impacts the robustness of stochastic neural networks. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=-zBN5sBzdvr.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Michael W. Dusenberry, Ghassen Jerfel, Yeming Wen, Yi an Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. Efficient and scalable bayesian neural nets with rank-1 factors. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2020.

Futoshi Futami, Tomoharu Iwata, Naonori Ueda, Issei Sato, and Masashi Sugiyama. Loss function based second-order jensen inequality and its application to particle variational inference. *Advances in Neural Information Processing Systems*, 2021.

Futoshi Futami, Tomoharu Iwata, Naonori Ueda, Issei Sato, and Masashi Sugiyama. Predictive variational bayesian inference as risk-seeking optimization. *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, 2022.

Xiang Gao, Meera Sitharam, and Adrian E Roitberg. Bounds on the jensen gap, and implications for mean-concentrated distributions. *The Australian Journal of Mathematical Analysis and Applications (AJMAA)*, 16.2 (14):1–18, 2019.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations*, 2015.

James-A Goulet, Luong Ha Nguyen, and Saeid Amiri. Tractable approximate gaussian inference for bayesian neural networks. *Journal of Machine Learning Research*, 22(251):1–23, 2021.

Martin Jankowiak, Geoff Pleiss, and Jacob Gardner. Parametric gaussian process regressors. In *International conference on machine learning*, pages 4702–4712. PMLR, 2020.

Alan Jeffares, Tennison Liu, Jonathan Crabbé, and Mihaela van der Schaar. Joint training of deep ensembles fails due to learner collusion. *Advances in Neural Information Processing Systems*, 36:13559–13589, 2023.

Jakob Nikolas Kather, Johannes Krisam, et al. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLOS Medicine*, 16(1):1–22, 01 2019.

Minyoung Kim and Timothy Hospedales. BayesDLL: Bayesian Deep Learning Library. In *arXiv preprint arXiv:2309.12928*, 2023. URL http://arxiv.org/abs/2309.12928.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the Second International Conference on Learning Representations*, April 2014.

Ranganath Krishnan, Pi Esposito, and Mahesh Subedar. Bayesian-torch: Bayesian neural network layers for uncertainty estimation, January 2022. URL https://github.com/IntelLabs/bayesian-torch.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Toronto, ON, Canada*, 2009.

JG Liao and Arthur Berg. Sharpening jensen's inequality. *The American Statistician*, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix gaussian posteriors. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1708–1716, 2016.

Jörg Lücke and Jan Warnken. On the convergence of the elbo to entropy sums. *arXiv preprint arXiv:2209.03077*, 2024. URL https://arxiv.org/abs/2209.03077.

David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3): 448–472, 1992.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Andres Masegosa. Learning under model misspecification: Applications to variational and ensemble methods. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/3ac48664b7886cf4e4ab4aba7e6b6bc9-Paper.pdf.

Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019.

Warren R Morningstar, Alex Alemi, and Joshua V Dillon. Pacm-bayes: Narrowing the empirical risk gap in the misspecified bayesian regime. *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, 2022.

Radford M Neal. Bayesian learning via stochastic dynamics. In *Advances in neural information processing systems*, pages 475–482, 1993.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

Luis A. Ortega, Rafael Cabañas, and Andrés R. Masegosa. Diversity and Generalization in Neural Network Ensembles. In *International Conference on Artificial Intelligence and Statistics*, 2022.

Matteo Pagliardini, Martin Jaggi, François Fleuret, and Sai Praneeth Karimireddy. Agree to disagree: Diversity through disagreement for better transferability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=K7CbYQbyYhY.

Abdul-Saboor Sheikh, Kashif Rasul, Andreas Merentitis, and Urs Bergmann. Stochastic maximum likelihood optimization via hypernetworks. In *Advances in Neural Information Processing Systems*, 2017.

Rishit Sheth and Roni Khardon. Pseudo-bayesian learning via direct loss minimization with applications to sparse gaussian process models. In *Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference*, volume 118 of *Proceedings of Machine Learning Research*, 2020. URL https://proceedings.mlr.press/v118/sheth20a.html.

Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger Grosse. Functional variational bayesian neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rkxacs0qY7.

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, page 180161, 2018.

Dmytro Velychko, Simon Damm, Asja Fischer, and Jörg Lücke. Learning sparse codes with entropy-based elbos. In *International Conference on Artificial Intelligence and Statistics*, pages 2089–2097. PMLR, 2024.

Ziyu Wang, Tongzheng Ren, Jun Zhu, and Bo Zhang. Function space particle optimization for bayesian neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BkgtDsCcKQ.

Yadi Wei and Roni Khardon. On the performance of Direct Loss Minimization for Bayesian Neural Networks. *I Can't Believe It's Not Better Workshop: Understanding Deep Learning Through Empirical Falsification at NeurIPS*, 2022.

Yadi Wei, Rishit Sheth, and Roni Khardon. Direct loss minimization for sparse gaussian processes. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, 2021. URL https://proceedings.mlr.press/v130/wei21b.html.

Florian Wenzel, Kevin Roth, Bastiaan S. Veeling, Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

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

Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 191–195, 2021.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

ELBO, regularized maximum likelihood, and their common one-sample approximation for training stochastic neural networks (Supplementary Material)

Sina Däubener[1] **Simon Damm**[1] **Asja Fischer**[1]

[1]Department of Computer Science, Ruhr University Bochum, Germany.

A PREDICTION VARIANCE AND PROOF OF PROPOSITION 1

Before proving the theoretical results in the main paper, we briefly summarize the theoretical findings from related works. Futami et al. [2021] focus on particle variational inference and derive a second-order Jensen inequality (their Theorem 3), which for fixed x reads

$$J(q(\theta)) \geq \mathbb{E}_{q(\theta)} \left(\frac{\ln(p(x|\theta) - \mathbb{E}_{q(\theta)} \ln(p(x|\theta)}{2h(x|\theta)} \right)^2$$

with

$$h(x,\theta)^{-2} = \exp\Big(\ln p(x|\theta) + E_{q(\theta)}[\ln p(x|\theta)] - 2 \max_{\theta} \ln p(x|\theta) \Big) \ .$$

The gap is upper bounded by the weighted variance of the loss function (in contrast to ours, where we focus on the variance in predictions), and utilized as a 'repulsion' loss term. Masegosa [2020] presents a lower bound on the Jensen gap in terms of the prediction variance (their Theorem 2) which is a special case of the results by Liao and Berg [2019], which for fixed x reads

$$J(q(\theta)) \geq \frac{1}{2 \max_{\theta} p(x|\theta)^2} \underbrace{\mathbb{E}_{\theta}[(p(x|\theta) - \mathbb{E}_{\theta} p(x|\theta))^2]}_{\sigma^2} \ .$$

We will make use of the latter inequality in our Proposition 1 . The bounds in Gao et al. [2019] relate the Jensen gap to the (centered) moments of a random variable, but are not directly applicable in our setting.

Let us now restate Proposition 1 and prove it.

Proposition 1 (Bounds on the Jensen Gap)**.** *Consider a parametrized distribution $p : (\mathcal{X} \times \mathcal{Y}) \times \Theta$, a posterior $q(\theta)$ over the parameter space Θ, and input pairs $(x_n, y_n) \in (\mathcal{X} \times \mathcal{Y})$ for $i \in \{1, \ldots, N\}$. Assume that for each n, $p(y_n|x_n, \theta)$ satisfies $p(y_n|x_n, \theta) \in [a_n, 1]$ for some $a_n > 0$ with*

- $\mu_n = \mathbb{E}_{\theta}[p(y_n|x_n, \theta)]$ *(mean),*
- $\overline{m}_n = \mathbb{E}_{\theta}[|p(y_n|x_n, \theta) - \mu_n|]$*, (absolute deviation)*
- $\sigma_n^2 = \mathbb{E}_{\theta}[(p(y_n|x_n, \theta) - \mu_n)^2]$ *(variance).*

Then, the Jensen gap $J(\theta)$ between the objectives is bounded by:

$$\sum_{n=1}^{N} \max \left\{ \frac{\sigma_n^2}{2}, \delta_{p,n} \right\} \leq J(q(\theta)) \leq \sum_{n=1}^{N} \min \left\{ \frac{\sigma_n^2}{2a_n^2}, \frac{\overline{m}_n}{a_n} \right\} \ ,$$

where for $p > 1$ and $n \in \{1, \ldots, N\}$

$$\delta_{p,n} := \ln \left(\frac{\mathbb{E}_{q(\theta)}[p(y_n|x_n, \theta)]}{\left(\mathbb{E}_{q(\theta)} \left[p(y_n|x_n, \theta)^{\frac{1}{p}} \right] \right)^p} \right) \geq 0 \ .$$

Proof. We are interested in deriving upper and lower bounds for some random variable X on $[a, b]$. Let X be a shorthand for the model prediction $p(y_n|x_n, \theta)$ (in dependence of the posterior $q(\theta)$) for some fixed data point (x_n, y_n). Further, the support of X satisfies $a > 0$ by assumption and $b = 1$ as we are analyzing a classification problem.

We start with an *upper bound* on the Jensen gap, involving the expected absolute deviation $\overline{m}$. The Jensen gap itself reads

$$J(X) = \ln(\mathbb{E}[X]) - \mathbb{E}[\ln(X)] \tag{8}$$

$$= \int \ln(\mu) - \ln(x) \, \mathrm{d}P(x) \tag{9}$$

where $P(x)$ denotes the density function of the random variable X. We continue with

$$\leq \int |\ln(\mu) - \ln(x)| \, \mathrm{d}P(x) \tag{10}$$

and by the Lipschitz-continuity of the logarithm on $[a, b]$ with Lipschitz constant $\frac{1}{a}$, i.e., $\forall x, y \in [a, b] : |\ln(x) - \ln(y)| \leq |x - y|/a$, we conclude

$$\leq \frac{1}{a} \int |\mu - x| \, \mathrm{d}P(x) = \frac{1}{a}\overline{m} \tag{11}$$

in which $\overline{m}$ denotes the first absolute centered moment of X.

We now turn to the *lower bound* based on the p-compressed expectation spread δ_p. This bound is inspired by the self-improvement version of the AM-GM inequality, see, e.g., Aldaz [2009]. We consider the random variable $X^{\frac{1}{p}}$ for some $p > 1$ (a 'compressed' version of X) and start with the classical Jensen inequality

$$\mathbb{E}_{q(\theta)}\left[\ln\left(X^{\frac{1}{p}}\right)\right] \leq \ln\left(\mathbb{E}_{q(\theta)}\left[X^{\frac{1}{p}}\right]\right) \tag{12}$$

which implies

$$\mathbb{E}_{q(\theta)}\left[\ln\left(X\right)\right] \leq p \ln\left(\mathbb{E}_{q(\theta)}\left[X^{\frac{1}{p}}\right]\right) \tag{13}$$

$$\leq \ln\left(\left(\mathbb{E}_{q(\theta)}\left[X^{\frac{1}{p}}\right]\right)^p\right) \tag{14}$$

$$\leq \ln\left(\left(\mathbb{E}_{q(\theta)}\left[X^{\frac{1}{p}}\right]\right)^p\right) + \ln\left(\mathbb{E}_{q(\theta)}[X]\right) - \ln\left(\mathbb{E}_{q(\theta)}[X]\right) \tag{15}$$

$$\leq \ln\left(\mathbb{E}_{q(\theta)}[X]\right) - \ln\left(\frac{\mathbb{E}_{q(\theta)}[X]}{\left(\mathbb{E}_{q(\theta)}\left[X^{\frac{1}{p}}\right]\right)^p}\right) \tag{16}$$

Rearranging and invoking the definition of X gives

$$\underbrace{\ln\left(\mathbb{E}_{q(\theta)}[p(y_n|x_n,\theta)]\right)}_{\text{from } \mathcal{L}_{\mathrm{ML}}} - \underbrace{\mathbb{E}_{q(\theta)}\left[\ln\left(p(y_n|x_n,\theta)\right)\right]}_{\text{from } \mathcal{L}_{\mathrm{VI}}} \geq \underbrace{\ln\left(\frac{\mathbb{E}_{q(\theta)}[p(y_n|x_n,\theta)]}{\left(\mathbb{E}_{q(\theta)}\left[p(y_n|x_n,\theta)^{\frac{1}{p}}\right]\right)^p}\right)}_{=:\,\delta_{p,n}} . \tag{17}$$

The Jensen inequality guarantees the non-negativity of $\delta_{p,n}$ (by concavity of the pth root).

The *variance-based bounds* follow from a slight modification from the argumentation in Masegosa [2020] (see statement above), which in turn is just a special case of the results by Liao and Berg [2019]. We include the proof for completeness: We utilize the Taylor series representation of the logarithm up to the second degree about $\mu = \mathbb{E}[\ln(X)]$ with the Lagrange form of the remainder, which reads

$$\ln(X) = \ln(\mu) + \frac{1}{\mu}(X - \mu) - \frac{1}{2\xi^2}(X - \mu)^2 \tag{18}$$

for some ξ between X and μ. Taking the expectation

$$\mathbb{E}\ln(X) = \underbrace{\mathbb{E}\ln(\mu)}_{=\ln(\mu)} + \frac{1}{\mu}\underbrace{(\mathbb{E}X - \mu)}_{=0} - \mathbb{E}\left[\frac{1}{2\xi^2}(X - \mu)^2\right]$$

and noting that $a \leq \xi \leq b$ implies

$$\mathbb{E}\left[\frac{1}{2\xi^2}(X-\mu)^2\right] \leq \mathbb{E}\left[\frac{1}{2\max \xi^2}(X-\mu)^2\right] = \frac{1}{2a^2}\sigma^2 \tag{19}$$

$$\mathbb{E}\left[\frac{1}{2\xi^2}(X-\mu)^2\right] \geq \mathbb{E}\left[\frac{1}{2\min \xi^2}(X-\mu)^2\right] = \frac{1}{2b^2}\sigma^2 \tag{20}$$

directly gives

$$J(X) = \ln(\mu) - \mathbb{E}\ln(X) = \mathbb{E}\left[\frac{1}{2\xi^2}(X-\mu)^2\right] \tag{21}$$

$$\implies J(X) \in \left[\frac{1}{2b^2}\sigma^2, \frac{1}{2a^2}\sigma^2\right] \quad . \tag{22}$$

All presented bounds hold for any $X_n = p(y_n|x_n,\theta)$ on $[a_n, 1]$, with mean μ_n, first absolute centered moment $\overline{m}_n$, variance σ_n^2 and p-compressed expectation spread $\delta_{p,n}$. We can thus combine Eqs. (11), (17) and (22) such that the Jensen gap satisfies

$$\sum_{n=1}^{N} \max\left\{\frac{\sigma_n^2}{2}, \delta_{p,n}\right\} \leq J(q(\theta)) \leq \sum_{n=1}^{N} \min\left\{\frac{\sigma_n^2}{2a_n^2}, \frac{\overline{m}_n}{a_n}\right\}, \tag{23}$$

which concludes the proof. □

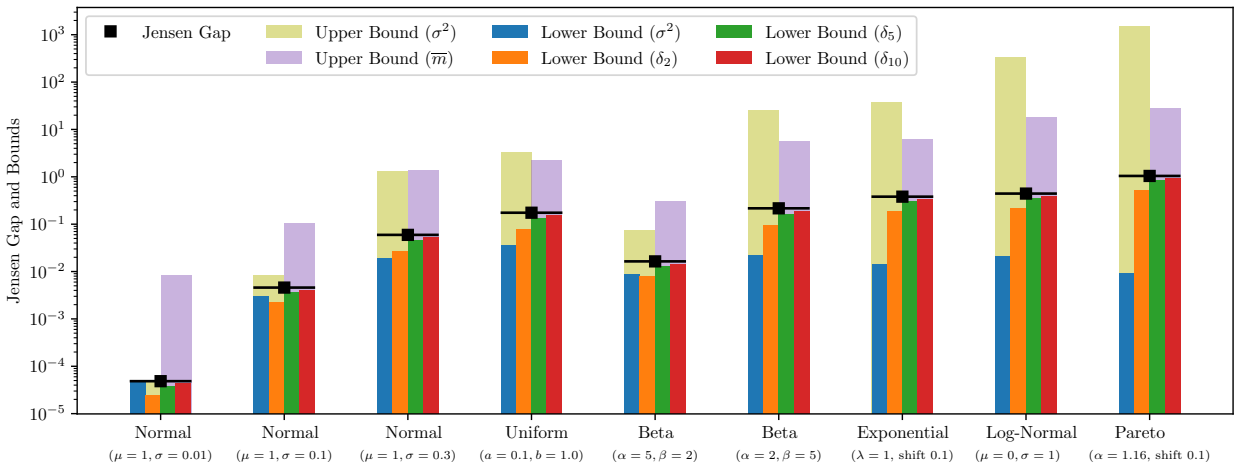


Figure 5: **The Jensen gap and a comparison of the presented bounds for various distributions**. Depicted are the Jensen gap for the logarithm $J(X) = \ln(\mathbb{E}[X]) - \mathbb{E}\ln(X)$ and upper and lower bounds from Proposition 1 for X following the specified distribution on $[a, b]$ (we simply take a and b to be the sample minimum/maximum; note that we include cases with $b \neq 1$). Depicted are the variance-based bounds from the Taylor expansion $\frac{1}{2b^2}\sigma^2 \leq J(X) \leq \frac{1}{2a^2}\sigma^2$, the upper bound $J(X) \leq \frac{1}{a}\overline{m}$ and the p-compressed expectation spread $\delta_p \leq J(X)$ for $p \in \{2, 5, 10\}$). Results are based on 100 samples per distribution.

Notably, the function $\delta_p\big(\theta; (y_n, x_n)\big)$ quantifies the variations of the random variable $p(y_n|x_n,\theta)$ by comparing the expectation of the 'compressed' random variable—by taking the pth root which pulls everything towards one—to the uncompressed expectation. We therefore refer to this quantity as the *p-compressed expectation spread*. When the predictions are almost constant (little variation), we have $\mathbb{E}_{q(\theta)}[p(y_n|x_n,\theta)]^{\frac{1}{p}} \approx \mathbb{E}_{q(\theta)}[p(y_n|x_n,\theta)^{\frac{1}{p}}]$, such that $\delta_p \approx 0$. For $p = 2$ the gap relates to the variance (of $\sqrt{p(y_n|x_n,\theta)}$) and usually becomes tighter with growing p.

A simple comparison of the different bounds is presented in Figure 5. We see that the variance-based bounds following Masegosa [2020], Liao and Berg [2019] become tight for very low variances (almost constant random variables), while the p-compressed expectation spread yields tighter lower bounds for higher variances and heavy-tailed distributions. In such

settings, the bound based on the absolute deviation is usually tighter then the variance-based bound. Thus, we decided to include both lower bounds in Proposition 1 to cater to both extremes.

To summarize the key point from Proposition 1: The Jensen gap between the objectives of interest grows (and vanishes) with the variability in the predictions. In the limit, i.e., if $\forall n : \mathrm{Var}_{q(\theta)}[p(y_n|x_n, \theta)] = 0$, equality between $\mathcal{L}_{\mathrm{ML}}$ and $\mathcal{L}_{\mathrm{VI}}$ is reached again.

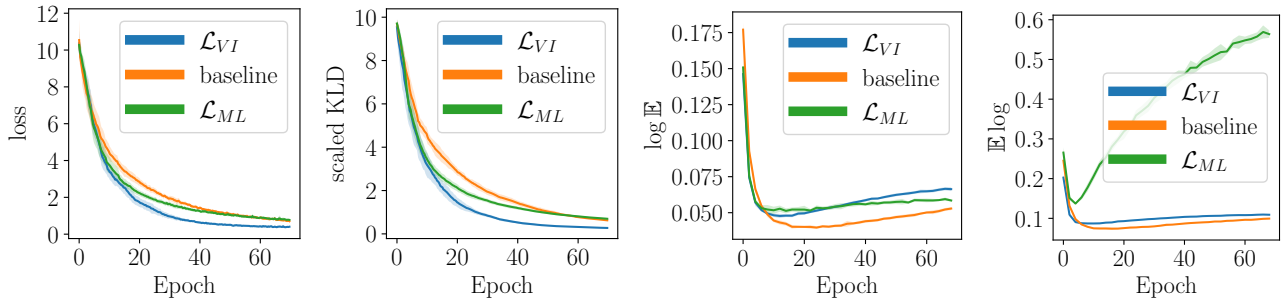
B TRAINING DETAILS

Training hyperparameters All models were trained with AdamW [Loshchilov and Hutter, 2019] with an initial learning rate of 0.001. Note, that for ResNet20 on CIFAR10 we used the example code by Krishnan et al. [2022] without modifications on the hyperparameter setting. Note, that in this implementation, the KL divergence of each layer is given as the mean KL divergence over the parameters in that layer and is hence down-weighted in comparison to the other models we used. All training runs were executed on NVIDIA A40 GPUs (a single A40 is sufficient for each single experiment).

Dataset	Resolution	#Classes	#Train	#Test
MNIST	28x28	10	60k	60k
FashionMNIST	28x28	10	60k	60k
CIFAR10	32x32	10	50k	10k
PathMNIST	224x224	9	89,996	7,180
DermaMNIST	224x224	7	7,007	2,005

Table 3: **Datasets** used in this work.

We used different numbers of epochs for the models trained on different datasets as we observed that although accuracy might be at the highest value, the balance between the expectation- and KL divergence is reached at a much later point in training. Therefore, we fixed the number of epochs to an amount where we saw the loss to be stagnating. That is, on MNIST, FashionMNIST after 70 epochs for the FF architecture and 100 epochs for the FF-MVN architecture, for CIFAR10 after 100 epochs (both ResNet20 and FF), for PathMNIST after 120 epochs and for models trained on DermaMNIST after 150 epochs. Note, that DermaMNIST is by far the smallest datasets which explains the increased number of epochs (cf. Table 3).



(a) Loss values (b) KL term in loss (c) $\ln \mathbb{E}$ (d) $\mathbb{E} \ln$

Figure 6: **Value of the respective loss functions, KL divergence,** $\ln \mathbb{E}$ **and** $\mathbb{E} \ln$ **during training** (on MNIST). The same characteristics were observed for other models and datasets.

Comparing training metrics This subsection presents training metrics for the simple feedforward architectures FF trained on MNIST [Deng, 2012]. Even though this is an easy task, we found the training behavior of this simple approach to be exemplary for all other models and datasets used in this paper.

First, it is noted that training for 70 epochs resulted in a test set accuracy of roughly 0.98 for each objective type, such that all models perform reasonably well. In Figure 6 we see (a) the respective losses of the models throughout training and (b) the respective KL divergence between the learned weight distribution and the prior. The loss is grossly dominated by the Kullback-Leibler divergence, which is more strongly minimized for the model trained with $\mathcal{L}_{\mathrm{VI}}$.

To investigate further differences during training, we tracked $\ln \mathbb{E}$ and $\mathbb{E} \ln$ for each training model in Figure 6(c) and (d). While the results for $\ln \mathbb{E}$, which $\mathcal{L}_{\mathrm{ML}}$ uses in its loss objective are comparable for all three training objectives, the results for $\mathbb{E} \ln$ significantly deviate. This indicates, that the gap described in proposition 1 is significantly increased for $\mathcal{L}_{\mathrm{ML}}$ being indicative of higher diversity in predictions. The baseline and models trained with $\mathcal{L}_{\mathrm{VI}}$ show a similar training behavior with respect to these metrics.

Mean of the learned weight distributions As mentioned in the main part of the paper, the mean weights' mean value distributions do not differ notably between the objectives of interest as can be seen in Figure 7.

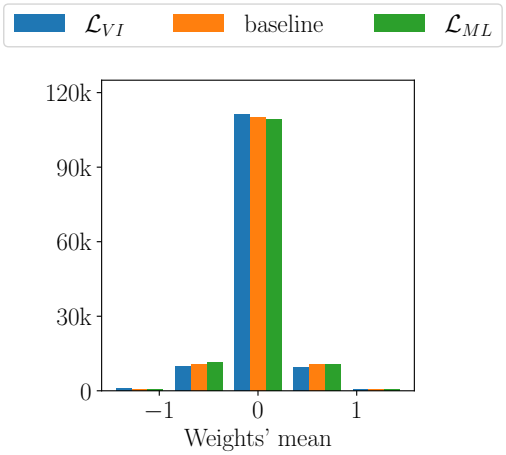


Figure 7: **Weights' mean values of the learned** $q(\theta)$ for the DINOTopping model on PathMNIST.

C PERFORMANCE WITH OTHER HYPERPARAMETERS

To ensure, that the finding of increased prediction variance is not only an artifact to our choice of prior or KL weighting, we experimented with different values of i) λ and ii) a distributional change of $p(\theta)$ to $\mathcal{N}(0, 3)$ and tested how these changes impact accuracy and prediction variance of the models.

In Table 4 we show the results for models trained on MNIST with MVN and observe, that the reduction of λ leads to higher accuracy (in line with findings with regards to the "cold posterior" effect, see e.g., the work by Wenzel et al. [2020]) while reducing the average per sample variance compared to the standard setting. Since λ scales the KL divergence which hinders $q(\theta)$ from collapsing the reduced prediction variance is expected. Interestingly though, a higher prior variance does not translate to a notably higher per-sample prediction variance but decreases the test set accuracy. We find this trend also to be true on the other data sets.

	Standard setting		$\lambda = 0.1$		$\sigma^2 = 3$	
	Accuracy	Avg var	Accuracy	Avg var	Accuracy	Avg var
$\mathcal{L}_{\mathrm{VI}}$	97.21 ± 0.11	0.02	98.39 ± 0.06	0.01	96.91 ± 0.12	0.02
baseline	97.01 ± 0.12	0.02	98.22 ± 0.06	0.01	96.67 ± 0.16	0.02
$\mathcal{L}_{\mathrm{ML}}$	97.21 ± 0.10	0.06	98.25 ± 0.11	0.02	96.83 ± 0.11	0.07

Table 4: **Influence of regularization strength** λ **and prior variance** σ^2 on test set accuracy (in %) and average prediction variance (MNIST). We here report mean and standard deviation over 10 different seeds, trained with Adam.

D ILLUSTRATION OF ENSEMBLES WITH LOW AND HIGH FUNCTION SPACE DIVERSITY

Here, we briefly recapitulate two prototypical scenarios for the behavior of ensemble members, as illustrated in Figure 8. In the left column, the models predict classes in the same ordering, in particular their predictions $\arg\max_c p(y_c|x, \theta_i)$ agree. In contrast, models' predictions in the right column disagree. This is possible even if *ensemble predictions* are identical (top row) or the *prediction variances* are identical (bottom row) in both scenarios. The latter motivates our analysis of the function space diversity in Section 2.2 (to answer the question whether models are just uncertain on some samples but still largely agree in prediction, or whether the models predict different classes).

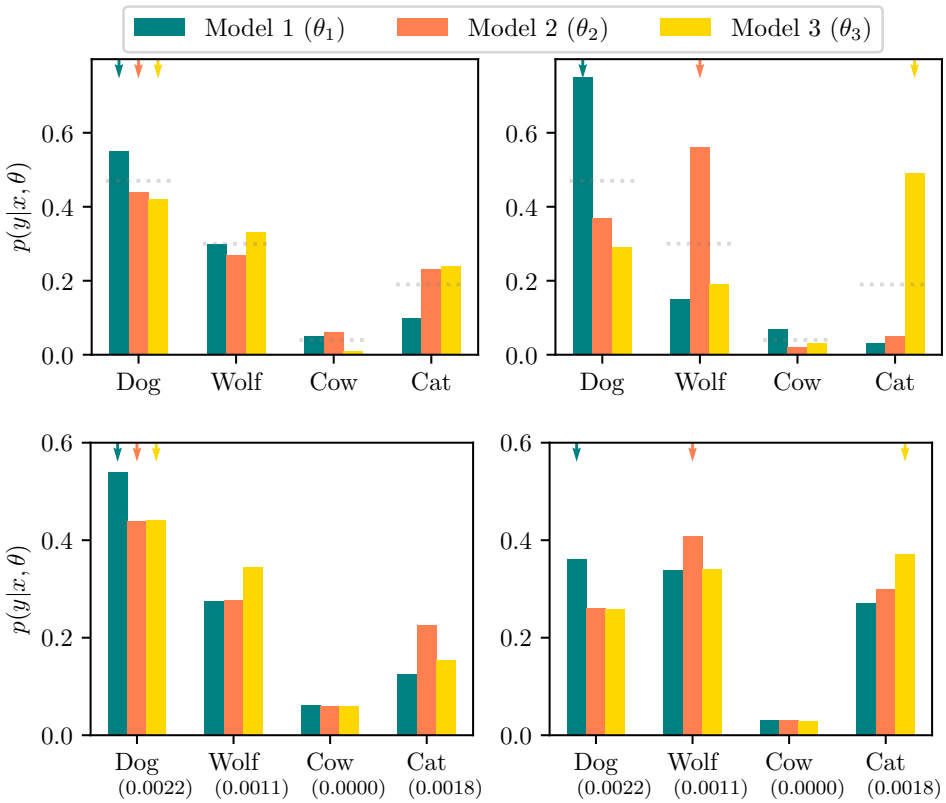


Figure 8: **Two scenarios depicting similar (left) and dissimilar behavior (right) across ensemble members** (model predictions indicated by small arrows on top). *Top row:* Ensemble predictions are identical (grey lines show $\frac{1}{3}\sum_{i=1}^{3} p(y|x, \theta_i)$). *Bottom row:* Prediction variances of the ensemble are identical (in parentheses below).

E EXAMPLES OF IMAGE CORRUPTIONS

In Figure 9 we show exemplary corruptions of the images which compose the OOD data sets used in Section 3.3 of the main paper.

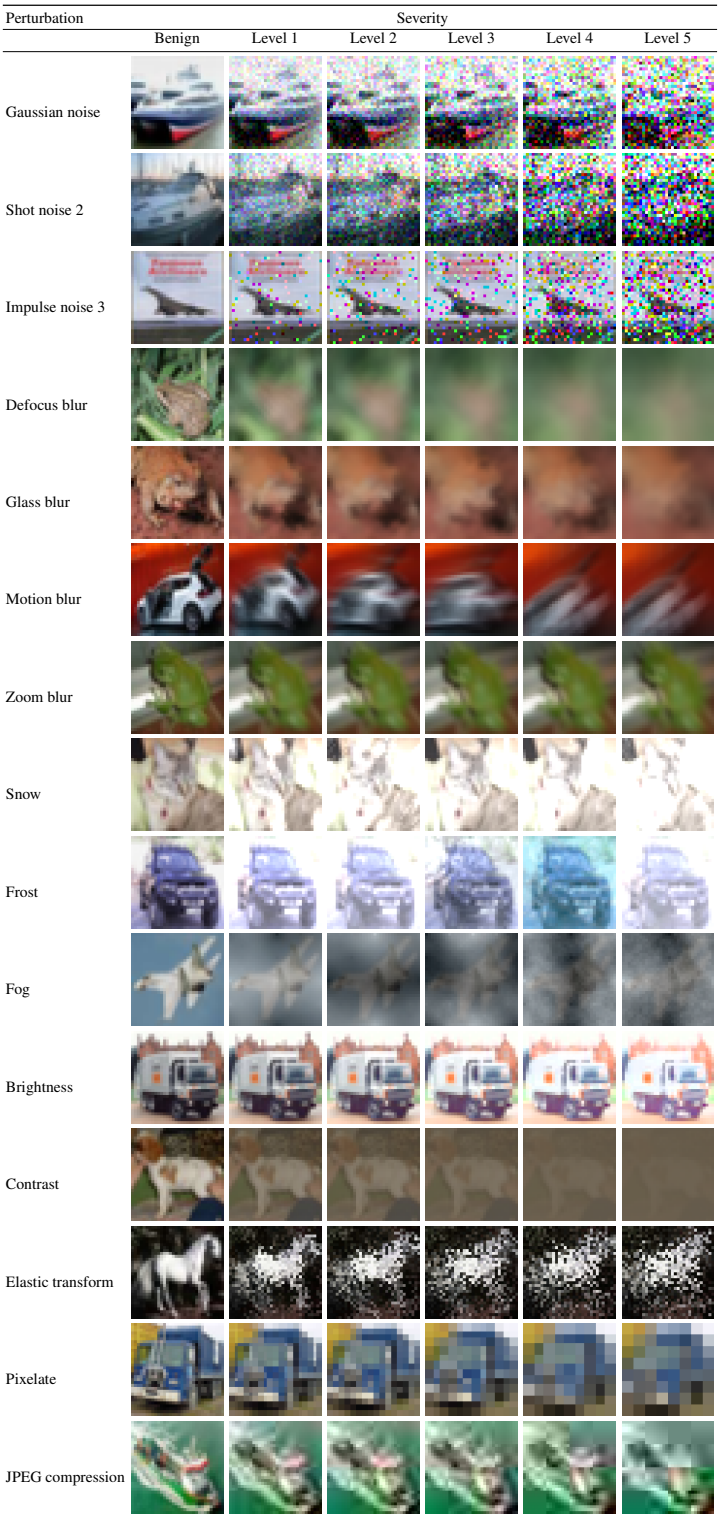


Figure 9: **Example images of CIFAR10 when applying different corruptions with increasing corruption strength** based on Michaelis et al. [2019]'s repository.