Tackling covariate shift with node-based Bayesian neural networks

Trung Trinh[1], Markus Heinonen[1], Luigi Acerbi[2], Samuel Kaski[1,3]
[1]Aalto University, [2]Helsinki University, [3]University of Manchester
{trung.trinh, markus.o.heinonen, samuel.kaski}@aalto.fi, luigi.acerbi@helsinki.fi

Aalto-yliopisto HELSINGIN YLIOPISTO MANCHESTER 1824 The University of Manchester ICML International Conference On Machine Learning

Overview

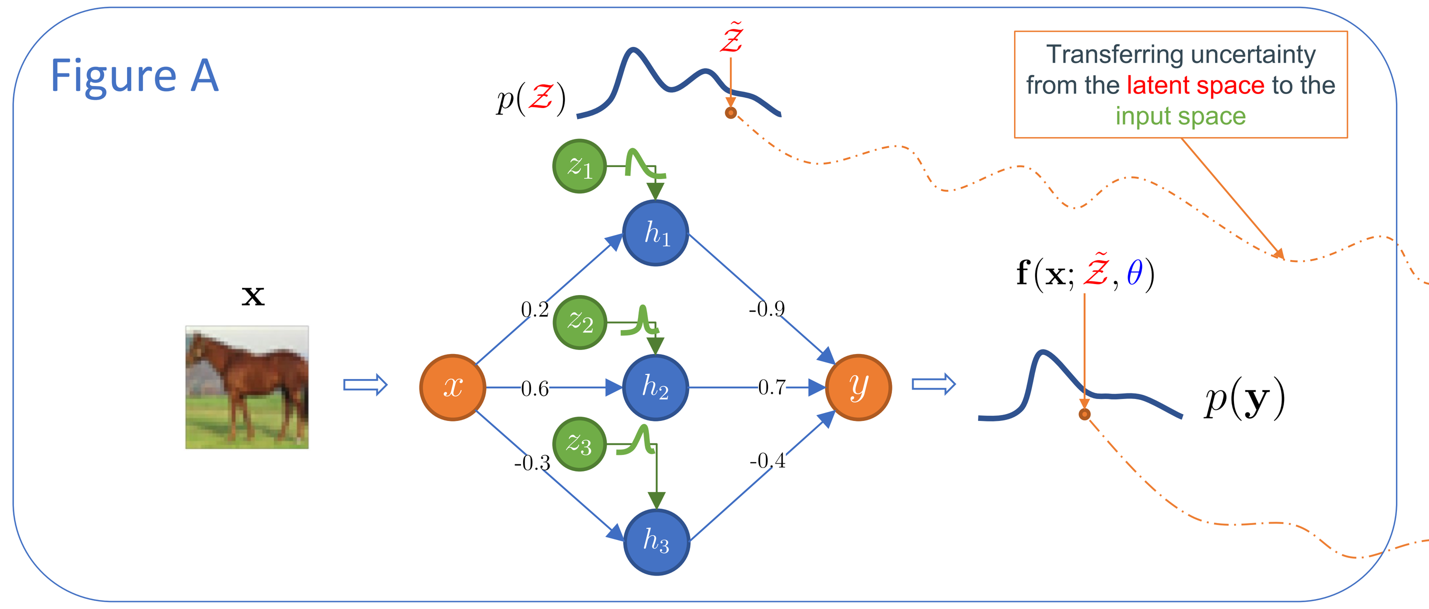
TL;DR: We explain why node-based BNNs, such as MC-Dropout [1] and Rank-1 BNNs [2], are robust against input corruptions and propose a method that further improves this robustness.

Node-based Bayesian neural networks

Description: The input vector of each layer is multiplied point-wise with a vector of latent random variables.

$$\mathbf{f}^{(\ell)}(\mathbf{x};\mathcal{Z},\theta)=\sigma\Big(\mathbf{W}^{(\ell)}\underbrace{(\mathbf{f}^{(\ell-1)}(\mathbf{x};\mathcal{Z},\theta)\circ\mathbf{z}^{(\ell)})}_{\text{Hadamard product}}+\mathbf{b}^{(\ell)}\Big)$$

$\mathcal{Z}=\{\mathbf{z}^{(\ell)}\}_{\ell=1}^{L}$: all the latent random variables with their own distribution $p(\mathcal{Z})$.
$\theta=\{(\mathbf{W}^{(\ell)},\mathbf{b}^{(\ell)})\}_{\ell=1}^{L}$: the weights and biases.
Given a sample $\tilde{\mathcal{Z}}\sim p(\mathcal{Z})$, we use $\mathbf{f}(\mathbf{x};\tilde{\mathcal{Z}},\theta)$ to denote the output of the node-BNN under this sample.



Training: We use variational inference [3] to simultaneously
1. find a MAP estimate of the weights and biases $\theta=\{(\mathbf{W}^{(\ell)},\mathbf{b}^{(\ell)})\}_{\ell=1}^{L}$,
2. infer the posterior distribution of the latent parameters $\mathcal{Z}=\{\mathbf{z}^{(\ell)}\}_{\ell=1}^{L}$.

We approximate the joint posterior $p(\theta,\mathcal{Z}|\mathcal{D})$ using the variational posterior:

$$q_{\phi,\hat{\theta}}(\theta,\mathcal{Z})=q_{\hat{\theta}}(\theta)q_{\phi}(\mathcal{Z})=\underbrace{\delta(\theta-\underbrace{\hat{\theta}}_{\text{MAP estimate}})}_{\text{Dirac delta}}\underbrace{q_{\phi}(\mathcal{Z})}_{\text{Mixture of Gaussians}}$$

Minimizing $\mathrm{KL}[q_{\phi,\hat{\theta}}(\theta,\mathcal{Z})||p(\theta,\mathcal{Z}|\mathcal{D})]$ is equivalent to maximizing the evidence lower bound (ELBO):

$$\mathcal{L}(\hat{\theta},\phi)=\underbrace{\mathbb{E}_{q_{\phi}(\mathcal{Z})}[\log p(\mathcal{D}|\hat{\theta},\mathcal{Z})]}_{\text{expected log-likelihood}}-\underbrace{\mathrm{KL}[q_{\phi}(\mathcal{Z})||p(\mathcal{Z})]}_{\text{KL divergence}}+\underbrace{\log p(\hat{\theta})}_{\text{log prior}}$$

Why are node-BNNs robust against corruptions?

Proposition: The distribution of the *latent variables* $p(\mathcal{Z})$ induces a distribution of *implicit corruptions* $p(\mathbf{x}^c)$ in the input space and by training under these corruptions, node-BNNs become robust against natural corruptions.

Given a node-BNN with a MAP estimate $\hat{\theta}$ and a latent distribution $p(\mathcal{Z})$, there exists a distribution $p(\mathbf{x}^c)$ such that the output distribution of the following two models are equal:

Node-BNN with a latent distribution (Fig. A)

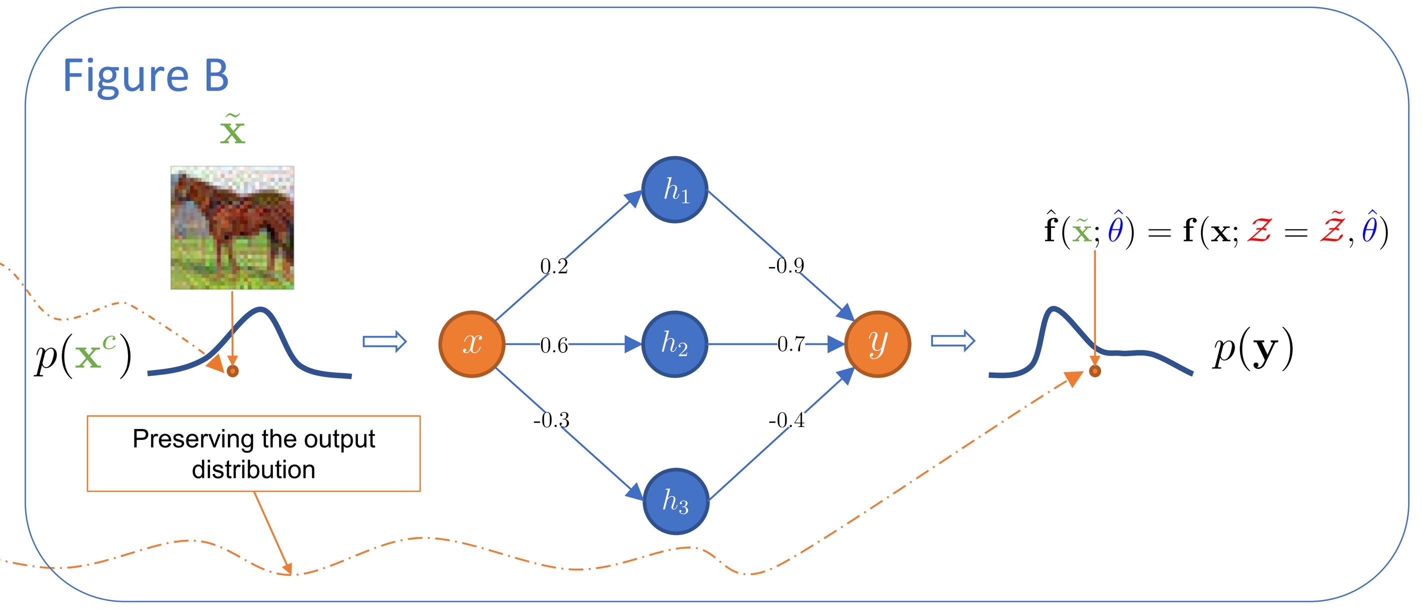
$$\tilde{\mathcal{Z}}\sim p(\mathcal{Z})$$
$$\tilde{\mathbf{y}}=\mathbf{f}(\mathbf{x};\mathcal{Z}=\tilde{\mathcal{Z}},\hat{\theta})$$

$\Longleftrightarrow$

Deterministic NN with an input distribution (Fig. B)

$$\tilde{\mathbf{x}}\sim p(\mathbf{x}^c)$$
$$\tilde{\mathbf{y}}=\hat{\mathbf{f}}(\tilde{\mathbf{x}};\hat{\theta})$$

where $\hat{\mathbf{f}}(\cdot;\hat{\theta})=\mathbf{f}(\cdot;\mathcal{Z}=\mathbf{1},\hat{\theta})$ is the deterministic NN obtained by removing all the latent variables in the node-BNN.



Entropic regularization

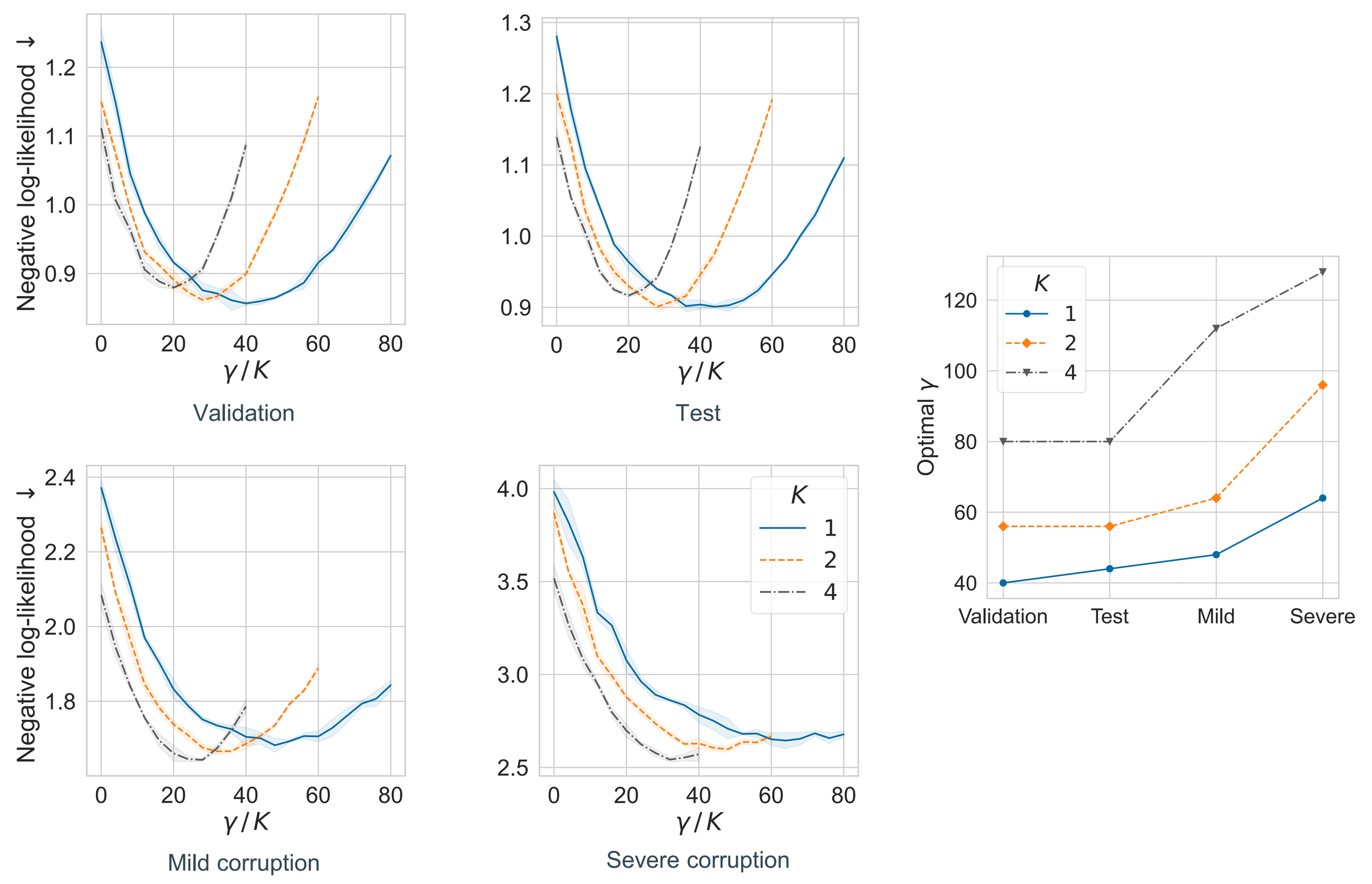
Proposition: Increasing the *latent entropy (the entropy of the latent variables)* diversifies the *implicit corruptions*, thereby making node-BNNs robust against a wider range of corruptions.

We maximize the latent entropy while approximating the true posterior by adding the γ-entropy term to the original ELBO where $\gamma>0$:

$$\underbrace{\mathcal{L}_{\gamma}(\hat{\theta},\phi)}_{\gamma\text{-ELBO}}=\underbrace{\mathcal{L}(\hat{\theta},\phi)}_{\text{original ELBO}}+\underbrace{\gamma\mathbb{H}[q_{\phi}(\mathcal{Z})]}_{\gamma\text{-entropy}}$$

Thus γ controls the trade-off between approximating the true posterior (via maximizing the original ELBO) and maximizing the latent entropy.
Higher γ results in higher latent entropy.

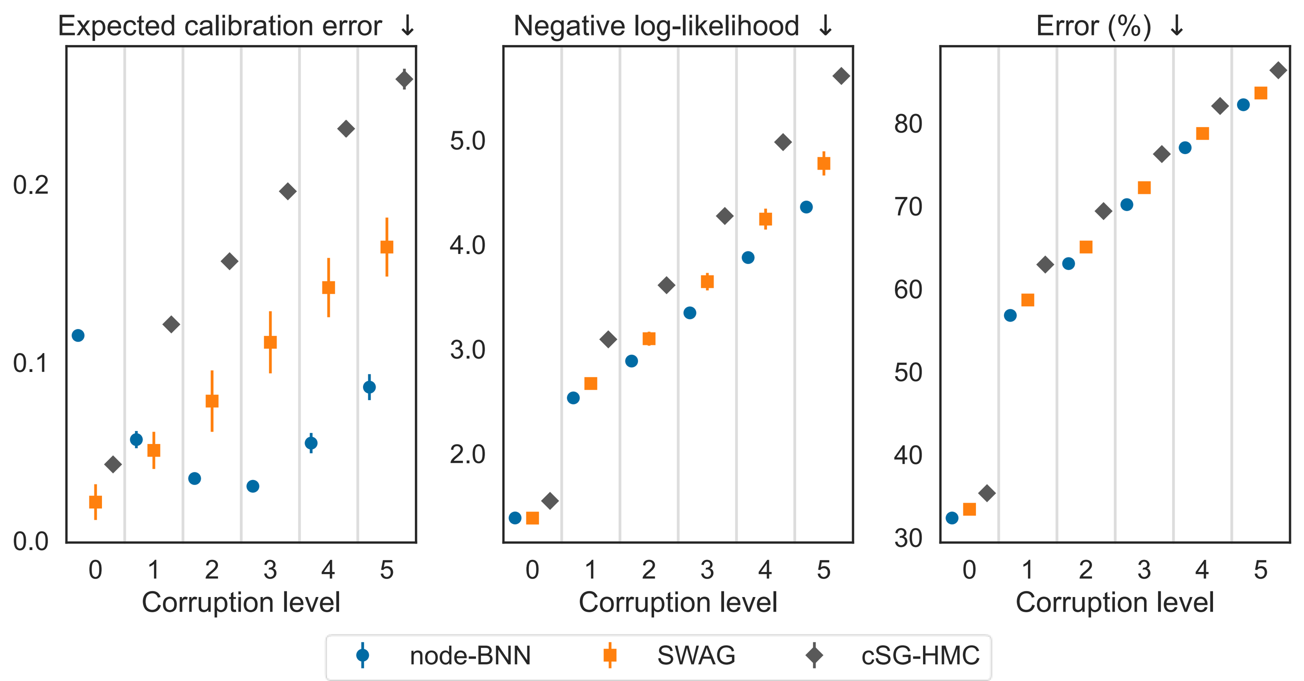
The latent entropy controls ID vs. OOD trade-off



Results of VGG16 [4] on CIFAR100 [5] and CIFAR100-C [6]. K is the number of Gaussian components in $q_{\phi}(\mathcal{Z})$.
For all test sets, performance improves as γ increases up to an optimal value then degrades afterwards.
More severe corruptions require higher optimal γ as shown in the right most plot.

Remark: The latent entropy controls the trade-off between performance on in-distribution (ID) samples and out-of-distribution (OOD) performance, with more severe corruptions require higher latent entropy which in turn decreases the ID performance.

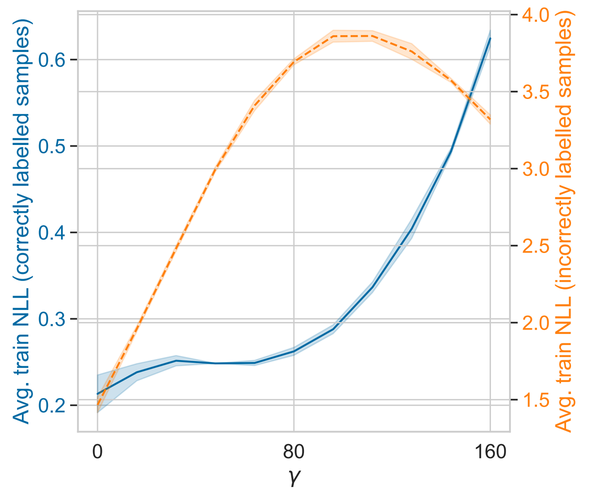
Benchmark comparison



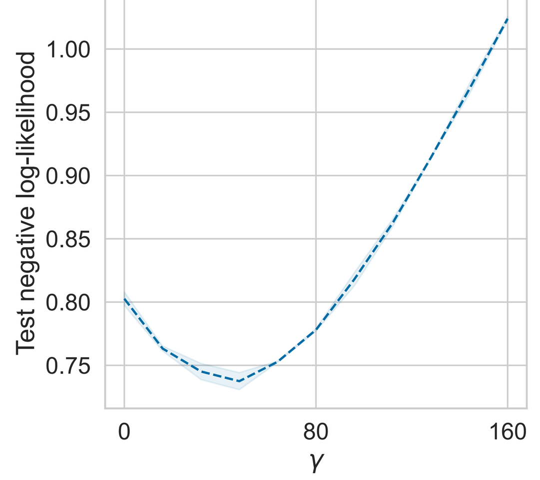
Results of PreActResNet18 [7] on TinyImageNet [8]. We chose SWAG [9] and cSG-HMC [10] as weight-based BNN baselines. Overall, entropy regularized node-BNN performs best under corruptions in all metrics and is only worse than the baselines in expected calibration error on corruption level 0 and 1.

Robustness against noisy training labels

Proposition: Learning generalizable patterns from correctly labelled samples is easier than memorizing random patterns from wrongly labelled samples [11]. Thus, if we corrupt wrongly labelled samples with sufficiently diverse corruptions then the model will fail to memorize these spurious patterns.



Train NLL of incorrectly labelled samples (in orange) increase much faster than the train NLL of correctly labelled samples (in blue).



Results on the clean CIFAR-10 test set

Results of ResNet18 [12] on CIFAR10 [5] where we randomly select 40% training samples and corrupt the labels of these samples.
On the left plot, we show the average NLL of the training samples with correct and incorrect labels separately. The high avg. NLL of training samples with noisy labels at high γ indicates that the model fails to memorize these samples. Hence, the model achieves better generalization on the clean test set at higher γ, as shown on the right plot.

Conclusion

1. The distribution of the latent variables induces a distribution of implicit corruptions in the input space and by training under these corruptions, a node-BNN becomes robust against natural corruptions.
2. Increasing the latent entropy (the entropy of the latent variables) diversifies the implicit corruptions, thereby improving the corruption robustness of the node-BNN.
3. The latent entropy controls the induced trade-off between ID performance and generalization under corruptions, with more severe corruptions require a higher latent entropy which in turn decreases ID performance.
4. As a side effect, a high latent entropy also provides robust learning under noisy training labels.

References

[1] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In ICML, 2016.
[2] Michael W. Dusenberry, et al. Efficient and scalable Bayesian neural nets with rank-1 factors. In ICML, 2020.
[3] David M. Blei, et al. Variational inference: a review for statisticians. arXiv preprint arXiv:1601.00670, 2017.
[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
[5] Alex Krizhevsky, et al. CIFAR-10 and CIFAR-100 datasets. 2009.
[6] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In ICLR, 2019.
[7] Kaiming He, et al. Identity mappings in deep residual networks. In ECCV, 2016.
[8] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. 2015.
[9] Wesley J. Maddox, et al. A simple baseline for bayesian uncertainty in deep learning. In NeurIPS, 2019.
[10] Ruqi Zhang, et al. Cyclical stochastic gradient MCMC for Bayesian deep learning. In ICLR, 2020
[11] Devansh Arpit, et al. A Closer Look at Memorization in Deep Networks. In ICML, 2017.
[12] Kaiming He, et al. Deep Residual learning for Image Recognition. arXiv preprint arXiv:1512.03385, 2015.