Data augmentation in Bayesian neural networks and the cold posterior effect

Seth Nabarro*[,1], Stoil Ganev*[,2], Adrià Garriga-Alonso[3], Vincent Fortuin[4], Mark van der Wilk[†,1], Laurence Aitchison[†,2]

[1]Imperial College London, [2]University of Bristol, [3]University of Cambridge, [4]ETH Zürich *†equal contribution

TL;DR

Existing methods which train Bayesian neural networks (BNNs) with data augmentation (DA) result in invalid likelihoods [7, 4]. A functional invariance viewpoint prescribes an easy-to-implement adjustment which lower bounds a proper log-likelihood. This improves performance, but does not reduce the cold posterior effect (CPE, [7]).

Data Augmentation and Cold Posteriors

- When training BNNs using DA we condition on more information than the N unaugmented observations
- However, the augmented $\mathbf{x}_i, y_i$ pairs do not have independent $p(y_i|\mathbf{x}_i)$, so cannot be treated as "real" observations with their own likelihood terms

Q1: How should we model DA when training BNNs?

- Previous works include DA in BNN training by replacing each input $\mathbf{x}_i$ with a sampled augmentation $\mathbf{x}'_i \sim p(\mathbf{x}'_i|\mathbf{x}_i)$, leaving inference algorithm and N unchanged
- This resulting targeted likelihood is invalid: $\sum_{\mathcal{Y}} \exp \mathbb{E}_{\mathbf{x}'|\mathbf{x}}[\log p(y|\mathbf{x}', \mathbf{w})] = Z(\mathbf{x})$
- It is known that the introduction of DA can induce a CPE in these cases [5, 3, 4]

Q2: Does a proper model of DA remove the CPE?

Probabilistic Model of Data Augmentation

We build functional invariance into the model, not the training data.

- We seek a function $h : \mathcal{X} \to \mathcal{Y}$ which is *invariant* wrt augmentations $p(\mathbf{x}'|\mathbf{x})$
- Following van der Wilk et al. [6] we construct the invariant function by summing a non-invariant function $g(.;\mathbf{w})$ (NN in this case) over the augmentation distribution

$$h(\mathbf{x};\mathbf{w}) = \int_{\mathcal{X}'} g(\mathbf{x}';\mathbf{w}) p(\mathbf{x}'|\mathbf{x}) d\mathbf{x}' , \tag{1}$$

and do inference on its parameters $\mathbf{w}$

- For classification, we have a choice as to what $g(.;\mathbf{w})$ represents: should we average logits ($g(.;\mathbf{w}) = \mathbf{f}(.;\mathbf{w})$, as in [6]) or probabilities ($g(.;\mathbf{w}) = \mathrm{softmax}\, \mathbf{f}(.;\mathbf{w})$)? We assess both empirically.
- (1) is intractable, however, we can lower bound any resulting log-concave likelihood via Jensen. Wenzel et al. [7] noted this in the case of averaging model probabilities

$$\mathcal{L}^i_{\text{prob}}(y_i;\mathbf{w}) = \log \mathbb{E}[\mathrm{softmax}_{y_i} \mathbf{f}(\mathbf{x}'_i;\mathbf{w})] \geqslant \mathbb{E}[\log \mathrm{softmax}_{y_i} \mathbf{f}(\mathbf{x}'_i;\mathbf{w})] = \hat{\mathcal{L}}^i_{\text{prob}} \tag{2}$$

but can we do better?

Tighter Bounds

Yes! **We can use multi-sample estimators [2] to get tighter bounds**

$$\hat{\mathcal{L}}^i_{\text{prob},K}(y_i;\mathbf{w}) = \mathbb{E}\left[\log \tfrac{1}{K}\textstyle\sum_{k=1}^K \mathrm{softmax}_{y_i} \mathbf{f}(\mathbf{x}'_{i;k};\mathbf{w})\right], \tag{3}$$

$$\hat{\mathcal{L}}^i_{\text{logits},K}(y_i;\mathbf{w}) = \mathbb{E}\left[\log \mathrm{softmax}_{y_i} \tfrac{1}{K}\textstyle\sum_{k=1}^K \mathbf{f}(\mathbf{x}'_{i;k};\mathbf{w})\right]. \tag{4}$$

At $K = 1$ both bounds reduce to standard DA, we explore the impact of $K > 1$ experimentally.

Further, we define **finite orbit** augmentations which admit **exact log-likelihood calculation**. We use this as a diagnostic to test if invalid likelihoods (such as in standard DA) cause CPE. For some deterministic functions $\{a_k(.)\}$ the finite orbit augmentation distribution is

$$p(\mathbf{x}'_i|\mathbf{x}_i) = \tfrac{1}{K}\textstyle\sum_{k=1}^K \delta(\mathbf{x}'_i - a_k(\mathbf{x}_i)). \tag{5}$$

Comment (3) and (4) describe a simple change to BNN training to incorporate DA: average the network output over multiple augmentations!

Non-Bayesian Network Results

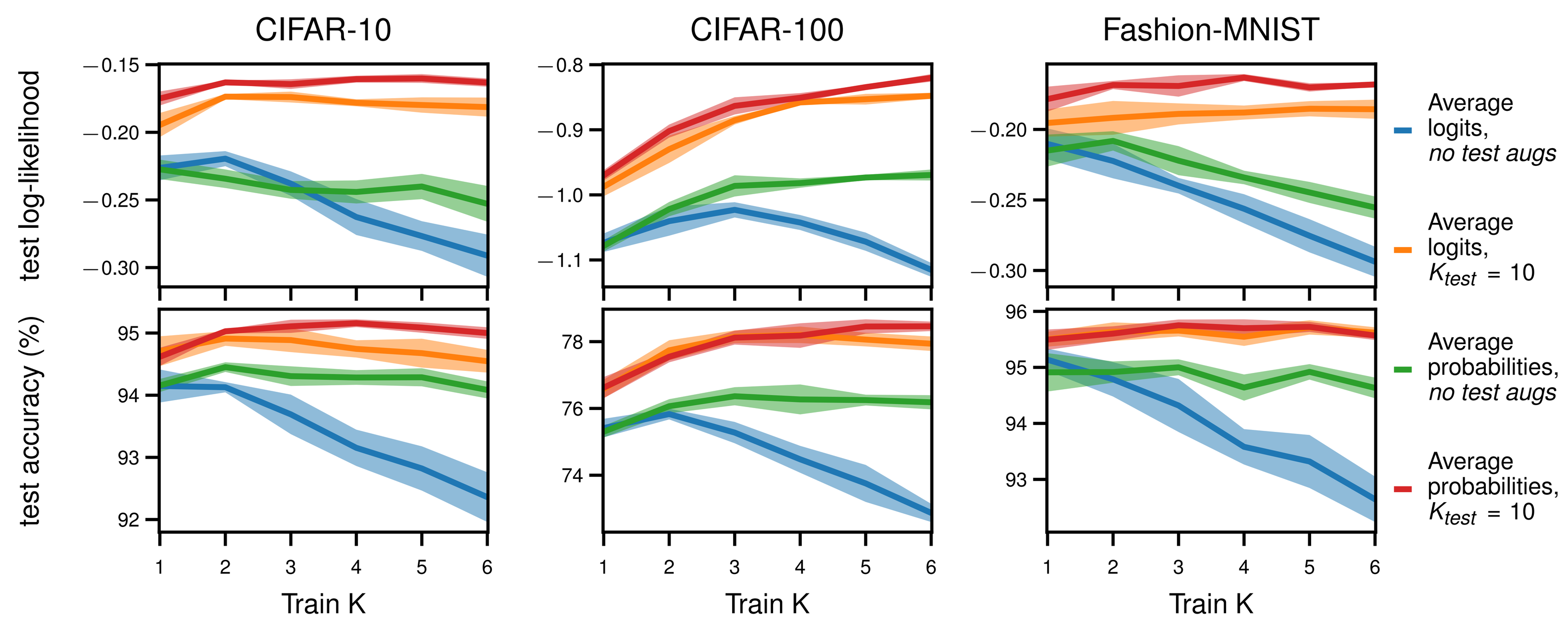

FIGURE 1: Variation of test performance with ResNet18 trained with SGD.

- Test time augmentation and averaging gives big improvement, even at $K_{\text{train}} = 1$
- When using test-time augmentation, optimal performance is at $K_{\text{train}} > 1$
- Overall, averaging probabilities > averaging logits
- Models trained with logit-averaging, but tested without any augmentation degrade with K_{train}
  - Underlying NN may become *less* invariant, but model construction (1) ensures invariance through averaging

Bayesian Neural Network Results

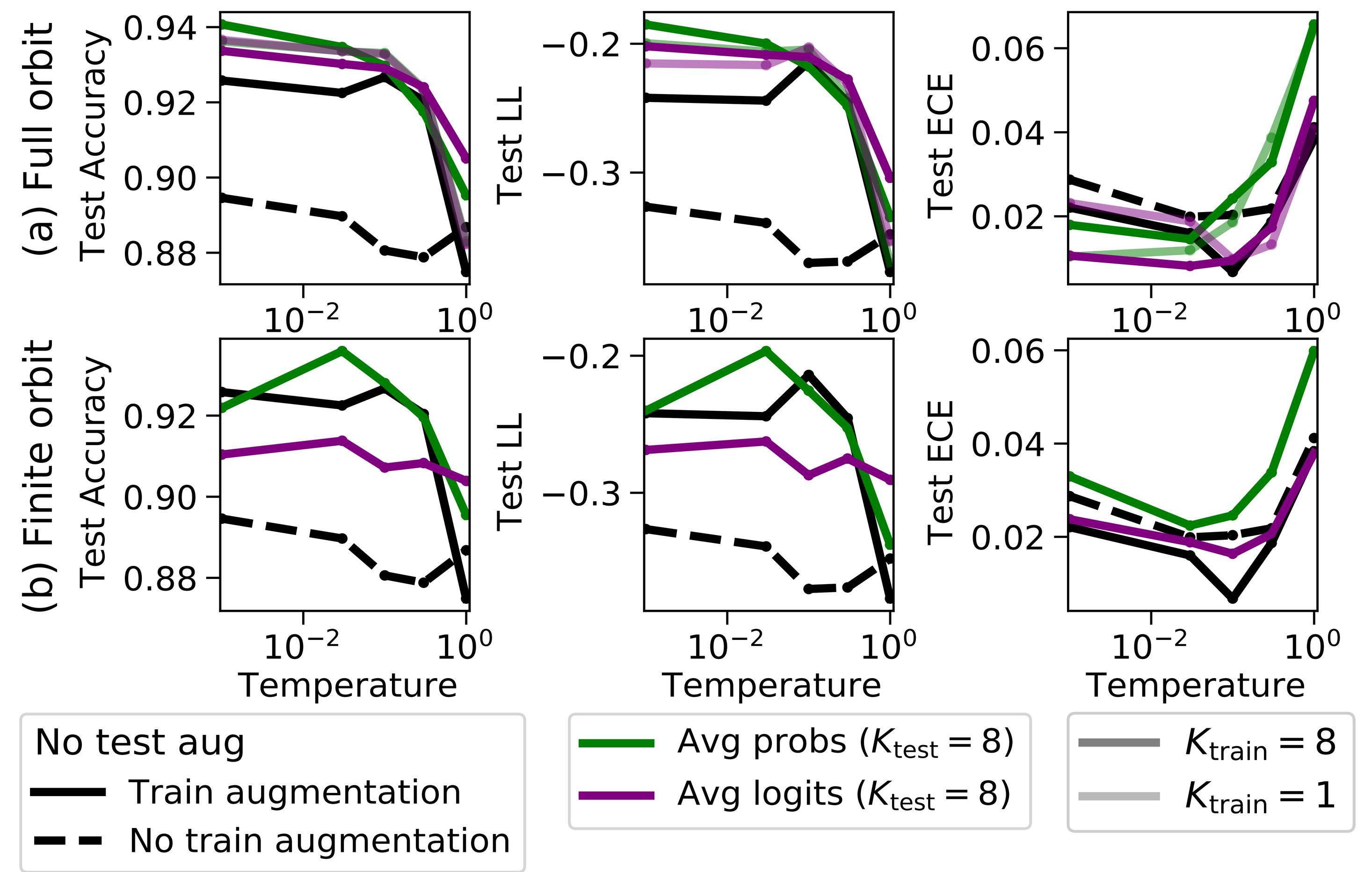

FIGURE 2: CPE for our augmentation configs. GGMC, CIFAR-10, ResNet20.

- Multi-sample log-likelihood bounds improve peak performance over standard DA
- The CPE persists, with the exception of averaging logits over finite orbit
- Averaging probabilities at low T has best overall performance
- Averaging logits performs best at $T = 1$

Conclusion

Our probabilistic model of DA, which produces a valid likelihood. can improve NN performance. However, the CPE remains, supporting other proposed explanations such as data curation [1], or prior misspecification [7, 5].

References

[1] Aitchison, L. A statistical theory of cold posteriors in deep neural networks. *arXiv preprint arXiv:2008.05912*, 2020.

[2] Burda, Y. et al. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

[3] Fortuin, V., Garriga-Alonso, A., et al. Bayesian neural network priors revisited. *arXiv preprint arXiv:2102.06571*, 2021.

[4] Izmailov, P., Vikram, S., Hoffman, M. D., and Wilson, A. G. What are Bayesian neural network posteriors really like? *arXiv:2104.14421*, 2021.

[5] Noci, L., Roth, K., Bachmann, G., et al. Disentangling the roles of curation, data-augmentation and the prior in the cold posterior effect. *arXiv preprint arXiv:2106.06596*, 2021.

[6] van der Wilk, M. et al. Learning invariances using the marginal likelihood. *arXiv preprint arXiv:1808.05563*, 2018.

[7] Wenzel, F., Roth, K., Veeling, B. S., et al. How good is the Bayes posterior in deep neural networks really? *arXiv preprint arXiv:2002.02405*, 2020.