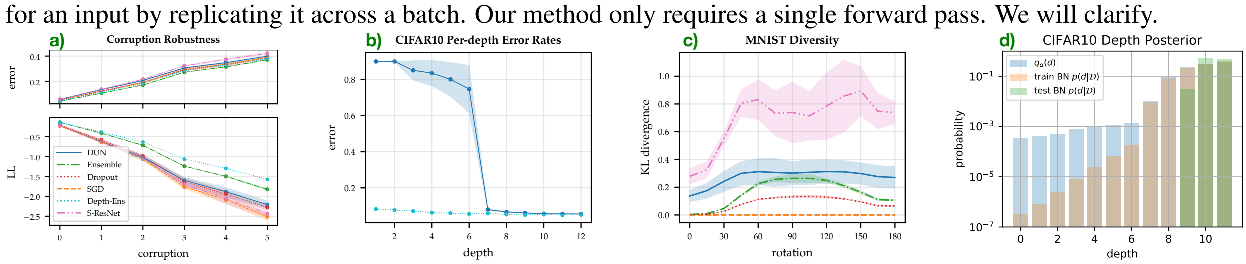
Overall: We thank the reviewers for their time and insightful comments/suggestions. We are happy that the reviewers appreciated the novelty and relevance of our contribution: Probabilistic reasoning is performed over NN depth, as opposed to more common weight space approaches (R1, R3, R4). Competitive uncertainty estimates are obtained with a single NN forward pass (R1, R2, R3, R4). We are also pleased that the reviewers highlighted that our paper is clear and detailed (R1, R2, R3), that our method is simple conceptually and in implementation (R1, R2, R3), and the comprehensiveness of our experimental evaluation (R2, R3). We are glad three reviewers recommend acceptance (R2, R3, R4). R1 is mainly concerned with our experimental evaluation. We provide new experiments and address others' concerns below. We will incorporate suggestions and expanded results using the additional page available.

R1 *Accuracy*: See Fig. 6, top row. For clarity, we will include an appendix table for all datasets. *Baselines*: Results for ensembles of different depths and stochastic depth resnets (ResNet50, uncertain layers 1-13 in both cases) are in **a)**. The former requires training multiple NNs and performs similarly to deep ensembles. Both require evaluating multiple NNs. The latter is a particularisation of MC Dropout, performs worse, and might inherit its limitations (Foong et. al., 2019). We will update all experiments in Fig. 6 with the new baselines. *Architecture, D*: We perform additional experiments exploring the effect of D and width on the depth posterior. These will be added to the Appendix. To summarise: larger D provides more opportunity for explanation diversity, and thus increased performance. Past a large enough D, increases yield diminishing returns. For wider blocks or simpler datasets (e.g. MNIST is simpler CIFAR10), D can be smaller without performance loss. The regularisation impact of D is usually negligible since, unlike the likelihood, the KL term in the ELBO does not scale with the data (see §B). *Depth posteriors*: Please see Fig. 3, and Figs. 7-10. *Expressive power*: DUNs trade off expressivity and explanation diversity automatically; Earlier layers obtain low accuracy's (see **b)**) and are assigned low posterior probabilities (see §3.1). They contribute negligibly to predictions, performing representation learning instead. In the limit of being capacity constrained, we have observed the posterior collapse to a delta, recovering a vanilla NN. In practise, NNs have excess modelling capacity. **a)** shows DUNs performing competitively with baselines given a fixed architecture. *Diversity Analysis:* **c)** shows the mean KL divergence between different depths' predictions, for DUNs, and different samples', for baselines. DUNs present large diversity in-distribution, potentially resulting in some underconfidence (Fig 6. bottom left). DUNs' diversity grows OOD, allowing for robustness. *No. parameters*: DUNs only add parameters where adaption layers are necessary. We adapt channels in CNNs with 1x1 convs (see §E3.1), which add few parameters. ResNet50: 23.52M weights. ResNet50 DUN (1-13): 26.28M. Increase of 1.17%. Our FC DUNs use constant width. Otherwise, width adaption could be efficiently implemented with low rank weight matrices. *Fig. 3 large variance*: As NNs are flexible, often underspecified models, their predictions can diverge OOD. This behaviour is also seen with ensembles (Figs. 4, 13, 19). Will include discussion. *Batch friendly methods*: Some baselines are parallelisable: i.e. multiple forward passes can be performed for an input by replicating it across a batch. Our method only requires a single forward pass. We will clarify.



R2 *Significance*: DUNs are conceptually simple but differ from most previous work in that they are a non weight space approach. This allows DUNs to sidestep the intractabilities and computational cost often associated with BNNs. DUNs are orthogonal to, and could be combined with, weight uncertainty. *Rejection-classification plot*: We agree with your assessment: underconfidence on correctly classified points leads to these being rejected together with OOD / wrongly classified points, flattening the curve. Requested posteriors are shown in **d)**. Batch norm (BN) seems to be the culprit. The exact posterior, computed with train mode BN, matches the variational posterior. The test mode BN posterior is more peaked and fixes underconfidence (Fig. 6). *ResNet50 timing*: We include loading times in our results as storing multiple ResNet50s in memory is often impractical. Without loading, ensemble times match dropout. We will clarify this and mention that inconsistent plot gaps are due to "single element" ensembles not considering loading times.

R3 *Expressive power*: Indeed, a shared output block could be a bottleneck. In practise, we do not observe this to be an issue. More flexible output blocks actually resulted in overfitting (see §I, Concat Pooling). The depth posterior allows blocks to specialise on either representation learning or predictions. Please see R1: Accuracy, Expressive power.

R4 *Comparison to Dropout*: For a fixed architecture, DUNs are always faster than Dropout (see Fig. 6 bottom right). Our regression experiments use Bayesian optimisation (BO) to choose architectures. In Table 1, the DUN is using a significantly deeper model. Even with BO, DUNs are most often faster than Dropout (Fig. 5 timing row). We find DUNs to outperform Dropout in terms of uncertainty estimation in most tasks (Fig. 5 TCE row and Fig. 6). *Limitations*: In practise, the complexity of weight space posteriors limits these methods' expressivity (Foong et. al., 2019). This can be seen in §4.2, §F.1. Depth uncertainty is orthogonal to weight uncertainty, side-stepping this issue. Both can be combined.