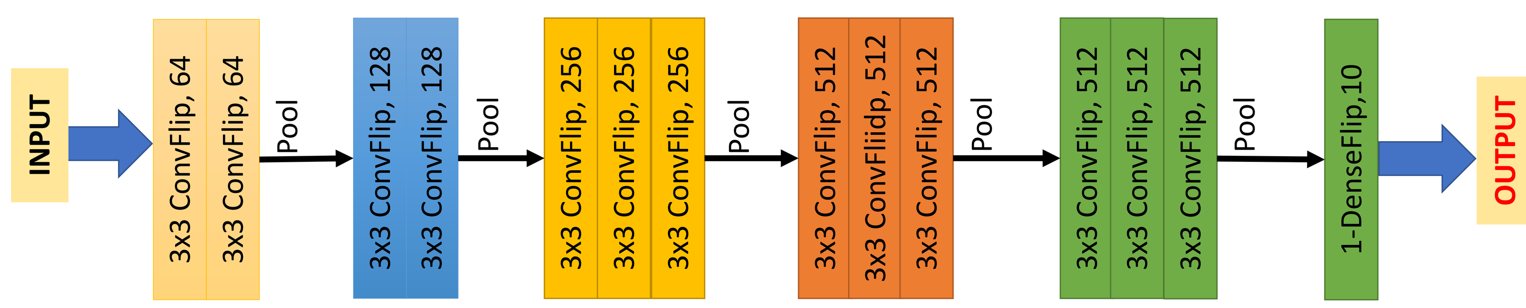
Analysisof Distributed Training of Bayesian Neural Networks at Scale

Himanshu Sharma[1] and Elise Jennings[2]
[1] Pacific Northwest National Laboratory (PNNL),[2] Ireland National High-Performance Centre
himanshu.sharma@pnnl.gov , elise.Jennings@ichec.ie

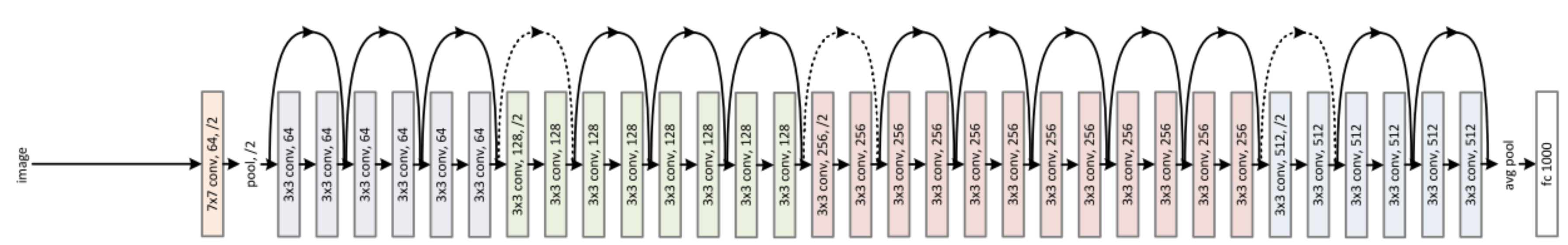
INTRODUCTION

- **Deep network** models are widely used for applications as diverse as skin cancer diagnosis from lesion images, steering in autonomous vehicles, cancer identification and various scientific applications.
- **Quantifying Uncertainties in Deep Neural networks** therefore becomes increasingly important for understanding robustness of these models.
- The uncertainties for these models can be broadly classified as **Aleatoric and Epistemic uncertainties.**
- **Aleatoric uncertainty** measures what you can't understand from the data. Think of aleatoric uncertainty as sensing uncertainty.
- **Epistemic uncertainty** measures what a model doesn't know due to lack of training data. It can be explained away with infinite training data. Think of epistemic uncertainty as model uncertainty.
- Capturing these uncertainty estimates in the models is computationally expensive and requires large computational budgets.
- Analyzing training performance at scale of **Bayesian Neural Net (BNN)** which provide uncertainty is crucial for efficient use of resources.
- The study here undertake two large image classification architecture shown below to understand the scalability of BNN on **High-performance computing systems XC40 10 Petaflop machine Theta at ALCF**.

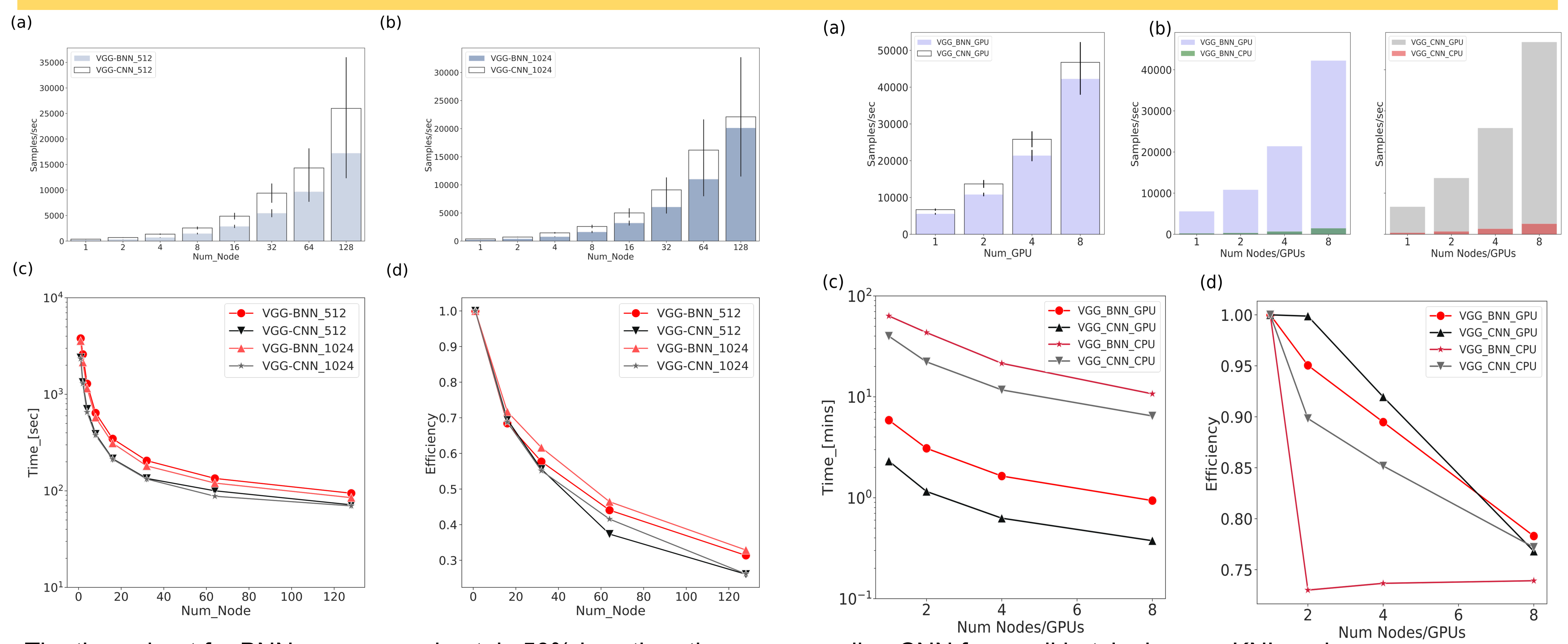
VGG-16 Network Architecture



Resnet-18 Network Architecture



RESULTS

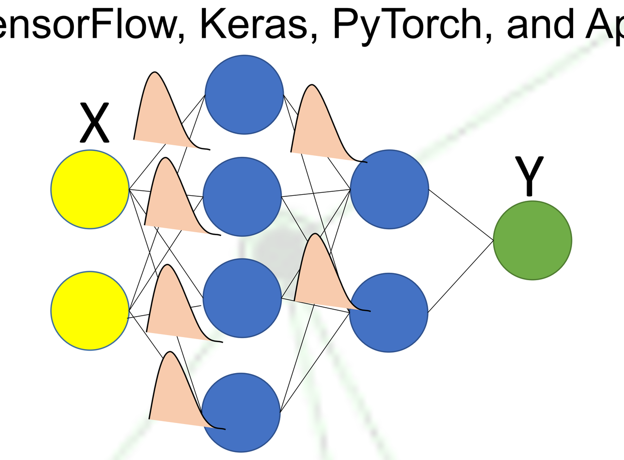


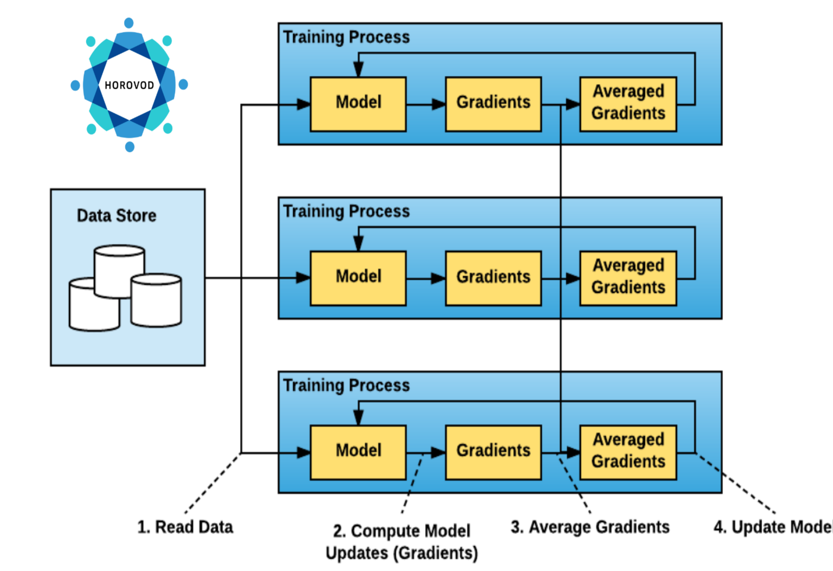
- The throughput for BNNs are approximately 50% less then the corresponding CNN for small batch sizes on KNL nodes..
- For BNN small batch sizes we find approximately a factor of 2.4 increase in the runtime for a fixed number of epochs.
- Overall we see a 30x increase in the FLOP rate for BNNs compared to CNNs.
- Runtime to a fixed accuracy can be up to a factor of $\sim 7\times$ longer on a small number of nodes but reduced to a factor of ~ 3X longer on >= 16 nodes.
- Using 8 GPUs, we find a ~ 29 (18) X increase in the throughput for BNNs (CNNs) compared to running on an equivalent number of KNL nodes.

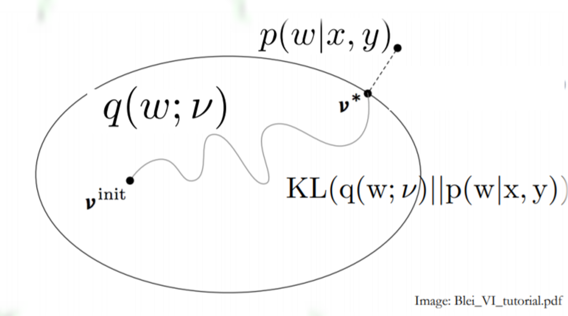
More Results: Sharma, H., Jennings, E. Bayesian neural networks at scale: a performance analysis and pruning study. Journal of Supercomputing (2020). https://doi.org/10.1007/s11227-020-03401-z

BAYESIAN NEURAL NETWORK & DISTRIBUTED TRAINING

- **Bayesian neural networks (BNNs, Bayesian NNs)** offer a probabilistic interpretation of deep learning models by inferring distributions over the models' weights.
- The model offers robustness to over-fitting, uncertainty estimates, and can easily learn from small datasets.
- Given the model parameters—the likelihood p(*D*|θ)—and a prior distributional belief on the model parameters p(θ).
- Bayes' Theorem yields the posterior p(θ|*D*)
- Scaling computation from one to many processors core/ GPUs can enable much faster training and research progress.
- Horovod, is an open source library that employs efficient inter communications via ring reduction and other routines to enable faster, easier distributed training for multiple Machine learning libraries. Such as TensorFlow, Keras, PyTorch, and Apache MXNet.







$$\int q(w,\nu) log(p(y|x,w))dw + KL(q(w;\nu)||p(w))$$

Expected log likelihood   Prior KL divergence

$p(\theta|\alpha)$ prior

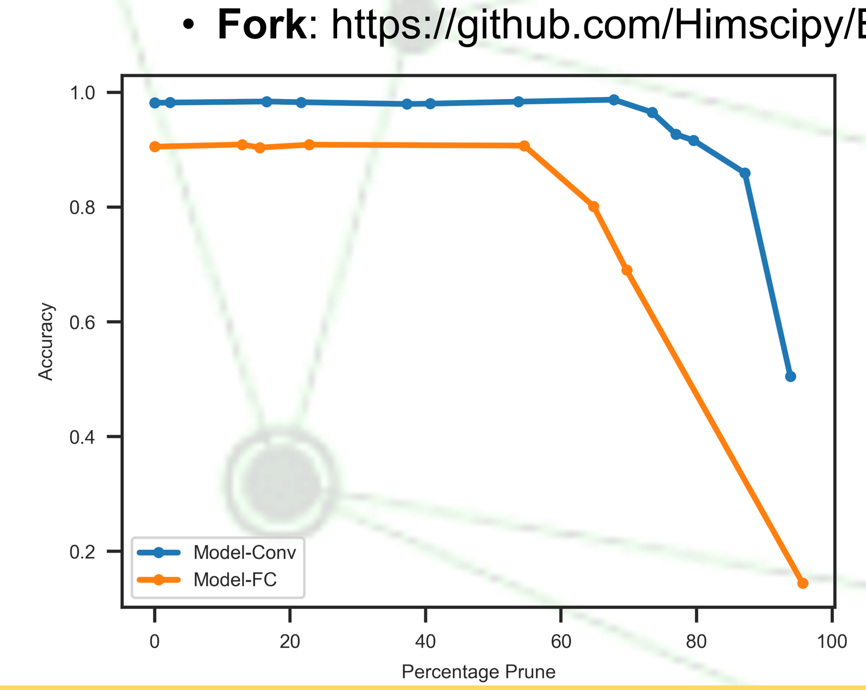
$p(\theta|D,\alpha) \propto p(y|x,\theta)p(\theta|\alpha)$ posterior

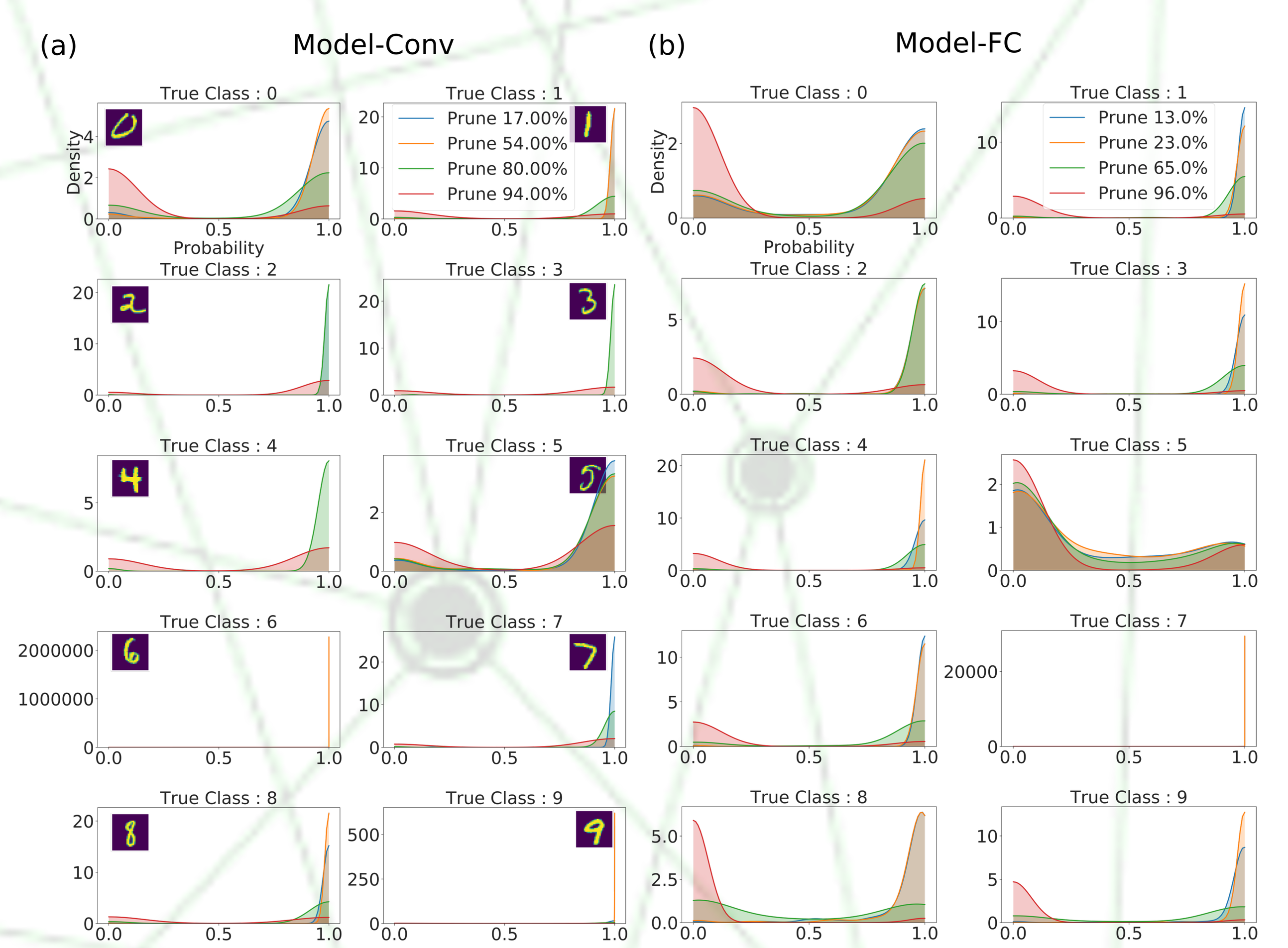
$p(y'|D,x',\alpha) = \int p(y'|x',\theta)p(\theta|D,\alpha)\mathrm{d}\alpha$ prediction

PRUNING BAYESIAN NEURAL NETWORK

BPrune

- The Pruning of trained BNN model are done to demonstrate the impact on prediction accuracy.
- The weights were pruned based on the signal to noise ratio (SNR) (mu/sigma).
- The library developed is available:
  - **PyPI**: https://pypi.org/project/BPrune/
  - **Fork**: https://github.com/Himscipy/BPrune





CONCLUSIONS & REFRENCES

- We compared the distributed training runs of two large BNN architecture.
- The scaling results show's that the CNN architecture outperform BNN in processing samples per sec due to less computational overhead in comparison to BNN, but the speed-up achieved with increasing number of ranks are nearly comparable to each other.
- The additional overheads are reasonable since uncertainty estimates are captured.
- We demonstrated the scalability of the BNN with large batch size and large data 0.1 Million MNIST Transformed Images.

REFERENCES:

- Radford M Neal. Bayesian Learning for Neural Network. Technical report,1995
- Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. arXiv preprint arXiv:1505.05424,2015
- Sergeev, Alexander, and Mike Del Balso. "Horovod: fast and easy distributed deep learning in TensorFlow." *arXiv preprint arXiv:1802.05799* (2018).

Acknowledgement: This research used resources of the Argonne Leader-ship Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357