Revista Colombiana de Computación
Vol. 25, No.2. July – Dec 2024, pp. 23-34
e-ISSN: 2539-2115, https://doi.org/ 10.29375/25392115.5274
Selected paper previously presented at the Latin America High Performance Computing Conference (CARLA 2023), an event held in Cartagena de Indias, Colombia, September 18-22, 2023.

REVISTA COLOMBIANA DE COMPUTACIÓN

Evaluation of Novel AI Architectures for Uncertainty Estimation

Evaluación de nuevas arquitecturas de IA para la estimación de la incertidumbre

Erik Pautsch[1,3], John Li[2,3], Silvio Rizzi[3], George K. Thiruvathukal[1,3], and Maria Pantoja[4,3]

[1] Loyola University Chicago, Chicago IL 60660, USA
[2] University of California San Diego, La Jolla CA 92093-0021, USA
[3] Argonne National Laboratory, Lemont IL 60439, USA
[4] California Polytechnic State University, San Luis Obispo CA 93407, USA

epautsch@luc.edu, jzl011@ucsd.edu, srizzi@anl.gov, gthiruvathukal@luc.edu, mpanto01@calpoly.edu

(Received: 6 February 2024; accepted: 18 June 2024, Published online: 31 December 2024)

Abstract. Deep Learning (DL) has advanced computer vision, delivering impressive performance on intricate visual tasks. Yet, the need for accurate uncertainty estimations, particularly for out-of-distribution (OOD) inputs, persists. Our research evaluates uncertainty in Convolutional Neural Networks (CNN) and Vision Transformers (ViT) using the MNIST and ImageNet-1K datasets. Using High-Performance (HPC) platforms, including the traditional Polaris supercomputer and AI accelerators like Cerebras CS-2 and SambaNova DataScale, we assessed the computational merits and bottlenecks of each platform. This paper delineates key considerations for using HPC in uncertainty estimations in DL, offering insights that guide the integration of algorithms and hardware for robust DL applications, especially in computer vision.

Keywords: Uncertainty, Deep Learning, Ensembles, Evidential Learning, Artificial intelligence.

Resumen. El Aprendizaje Profundo (AP) ha hecho avanzar la visión por ordenador, ofreciendo un rendimiento impresionante en tareas visuales complejas. Sin embargo, persiste la necesidad de estimaciones precisas de la incertidumbre, en particular para las entradas fuera de distribución (OOD, en su acrónimo en inglés). Nuestra investigación evalúa la incertidumbre en Redes Neuronales Convolucionales (CNN, en inglés) y transformadores de visión (ViT, en inglés) utilizando los conjuntos de datos MNIST e ImageNet-1K. Utilizando plataformas de Alto Rendimiento (HPC, en inglés), incluidos el superordenador tradicional Polaris y aceleradores de IA como Cerebras CS-2 y SambaNova DataScale, evaluamos los méritos computacionales y los cuellos de botella de cada plataforma. En este artículo se describen las consideraciones clave para utilizar la HPC en la estimación de la incertidumbre en el AP, y se ofrecen ideas que guían la integración de algoritmos y hardware para aplicaciones de AP robustas, especialmente en visión por ordenador.

Palabras claves: Incertidumbre, Aprendizaje Profundo, Aprendizaje por conjuntos, Aprendizaje evidencial, Inteligencia Artificial.

1 Introduction

What is Uncertainty? Let us assume that we have an unknown function $f(x)$ that represents the predicted output based on a given input. For this example, $f(x) = x$, the true underlying function, is represented by the green dashed line in Figure 1. When using a neural network (NN) to predict this underlying function, we sample over a reduced range of the function at different locations (blue dots in Figure 1). Training the model with the sampled data, we attempt to infer the values of $f(x)$ over the full range of the function, including areas that have not been sampled. In the unsampled regions, the model is essentially guessing, which should

lead to high uncertainty. Uncertainty also arises when the model tries to infer values from locations where we did sample, but due to factors such as sensor error, the samples deviate from the real value. This leads to increased uncertainty, especially when compared to areas where the samples closely follow the real function.

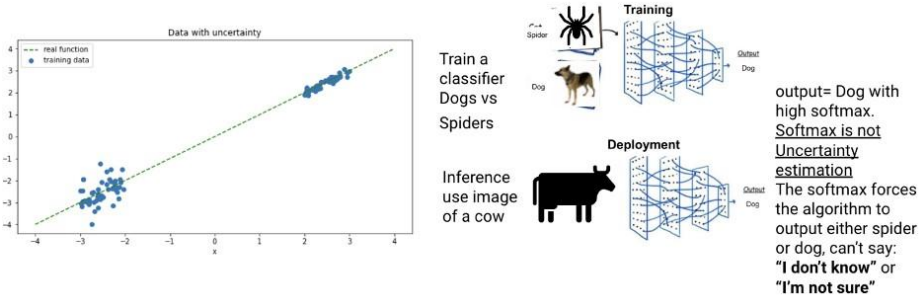

Figure 1. Left: An example of highly uncertain sampled data from a linear underlying function Right: Demonstration of the necessity for uncertainty quantification in out-of-distribution (OOD) data

Uncertainty in DL can be categorized into two main categories: epistemic and aleatoric. Epistemic uncertainty, also known as model uncertainty, originates from insufficient or flawed model training. On the other hand, aleatoric uncertainty denotes data-related uncertainties, often stemming from sensor inaccuracies or obstructions in imaging. Figure 1 shows areas with high aleatoric uncertainty (lower blue dots) and areas with low aleatoric uncertainty (higher blue dots). Simply put, uncertainty refers to the lack of confidence in each output of a DL model.

Why is uncertainty important in DL? In some DL tasks, such as generating artistic patterns, uncertainty evaluations may not be necessary. However, in certainty-critical applications, like pedestrian detection for self-driving cars (Bojarski, et al., 2017), where an AI model might fail to recognize a partially obscured human, evaluating uncertainty becomes crucial. Being able to output "I don't know" and ask for help can be as vital as achieving high accuracy and robustness. Recent research has emphasized the importance of uncertainty in DL. For example, in Ren, et al. (2023), researchers trained a robot for path planning to ask for help when uncertain by measuring the uncertainty of Large Language Models (LLM). In MacDonald, et al. (2023), various Bayesian DL models were benchmarked to predict uncertainty in the classification of cancer of an unknown primary, using three RNA-seq datasets.

We will summarize several popular uncertainty quantification methods: Temperature scaling is a post-processing technique to calibrate neural networks (Guo, Pleiss, Sun, & Weinber, 2017). After temperature scaling, you can trust the probabilities output by a neural network: Temperature scaling divides the logits (inputs to the SoftMax function) by a learned scalar parameter. Unlike deterministic DL models, Bayesian Neural Networks (BNNs) (Gal & Ghahramani, 2016) treat each weight as a probability distribution, producing different outputs for the same image (if the image is a label, it should always produce the same output label, but if there is uncertainty on the input, the output will vary). Therefore, the output will be the label and the standard deviation (uncertainty).

One of the most common ways of implementing a BNN is by Monte Carlo Dropout (Gal & Ghahramani, 2016), where a random number of neurons is dropped during training and prediction. Some of the most successful solutions to quantify uncertainty are based on ensembles (Ovadia, et al., 2019), where an ensemble of models is trained with the same data but different hyperparameters and initial weights, and the results are aggregated. The problem with ensembles is that they are computationally expensive since they require training the same model several times. Inference is also slow since it needs to conduct a forward pass using each of the several models trained in order to aggregate them. Evidential Learning (Amini, Schwarting, & Rus, 2020) alleviates this issue, variability is added to the loss function instead of the weights. Liu and Guo (2020) introduces Peer Loss Functions, a new family of loss functions that enables learning from noisy labels without needing prior specification of the noise rates. It works by pairing each sample with another random sample and learning to distinguish between them. This both simplifies model development and improves robustness when faced with noisy labels.

Pretrained Transformers such as BERT achieve high accuracy on in distribution examples and anomalous or Out of Distribution Data (ODD) samples (Hendrycks, et al., 2020). Models specifically trained to be robust against OOD inputs encourage networks to memorize the specific corruptions seen during training. Therefore, they propose using pretraining techniques like self-supervised learning. In selective classification for DL (Geifman & El-Yaniv, 2017), models reduce the error rate by abstaining from

prediction when uncertain, using Monte Carlo dropout plus a SoftMax threshold to avoid predicting what it does not know. Uncertainty estimation is also vital in active learning, allowing models to resolve task ambiguity by identifying informative examples through uncertainty sampling (Tamkin, Nguyen, Deshpande, Mu, & Goodman, 2022).

In our study, we investigate two leading uncertainty quantification techniques: ensembles and evidential learning. We selected ensembles due to their established track record in enhancing accuracy and uncertainty estimation, albeit at a higher computational cost. Conversely, evidential learning stands out because it sidesteps additional training or inference phases. Subsequent sections delve into evaluations of these methods across diverse architectures and datasets. The main contribution of this paper encompasses the integration of uncertainty assessments to novel application areas, such as adversarial attacks on ViTs, juxtaposing various uncertainty estimation strategies within these domains, and streamlining uncertainty assessments on emerging HPC platforms like the Cerebras CS-2 and Sambanova DataScale.

The organization of the paper is as follows: In Section 2, we describe the datasets, uncertainty evaluation methods, and the HPC systems we used for our experiments. Section 3 provides a summary of our findings, and in Sections 4 and 5, we present our conclusions and directions for future work.

2 Methods

Uncertainty Evaluation: Our research focused on understanding uncertainty estimation in DL. We evaluated various methods and found ensembles and evidential learning to be especially relevant due to their unique attributes.

Ensembles: Ensembles (Wenzel, Snoek, Tran, & Jenatton, 2020) aggregate outputs from several models trained independently on identical data. These models arise from Hyperparameter Optimization (HPO), thus negating extra computational time. By aggregating the M best models, we yield optimal hyperparameters and uncertainty estimations. With input x and label y, a neural network (NN) predicts the distribution $p\theta(y \mid x)$ where θ represents the NN's parameters. We harness an ensemble of M models and their hyperparmeters, $(\theta_m)_{m=1}^{M}$, training them in parallel on the same dataset. The combined prediction is: $p(y \mid x) = M^{-1} \sum_{m=1}^{M} p_{\theta_m}(y \mid x, \theta_m)$. Uncertainties are calculated from the standard deviation of predicted probabilities across the ensemble.

Evidential Learning: Evidential Learning (Amini, Schwarting, & Rus, 2020) differentiates itself by replacing a classical NN's SoftMax layer with an evidence vector, predicting the Dirichlet distribution parameters. For a given sample i with $f(x_i \mid \theta)$, the parameters are $\alpha_i = f(x_i \mid \theta)+1$. Class-specific probabilities are computed accordingly. Obtaining an overall uncertainty, instead of per-class values, was a major limitation. To address this, we introduced *Binary Evidential Learning*: a method that trains k binary classifiers, with calculations resembling standard evidential learning, but for $k = 2$ classes.

2.1. Models and Datasets

We worked with two pivotal datasets in the DL realm: MNIST (Lecun, et al., 1995) and ImageNet1K (Cordonnier, Loukas, & Jaggi, 2020). Both of these datasets are highly regarded in the community and offer different challenges due to their diverse complexity and content.

CNN with MNIST: The MNIST (Modified National Institute of Standards and Technology) dataset is among the most well-known datasets in the DL domain. It comprises 60,000 grayscale images (28x28 pixels) of handwritten digits ranging from 0 to 9, along with an additional 10,000 images designated for testing. The simplicity and clarity of MNIST make it a suitable benchmark for methods and models. It allows researchers to understand and validate basic functionalities before diving into more intricate datasets.

ViTs and ImageNet: ImageNet-1K, a subset of the larger ImageNet database, is a challenging dataset containing over 1.2 million high-resolution images across 1,000 classes. These classes span a vast spectrum, from everyday objects to abstract concepts. Due to its size and variety, ImageNet-1K is widely used to benchmark algorithms intended for real-world applications. Our ViTs, paired with this dataset, provided an in-depth examination space for understanding model robustness in complex environments. Additionally,

the introduction of the Projected Gradient Descent (PGD) adversarial attack simulated real-world conditions where subtle changes in input data can dramatically affect model predictions. While the MNIST dataset tested foundational principles, ImageNet1K with the PGD attack gauged the scalability and resilience of our methods.

2.2. HPC Architectures

We utilized both traditional HPC systems like Polaris and modern DL hardware such as the Cerebras CS-2 and SambaNova DataScale. Here's a brief overview of these systems:

Polaris: Developed by Argonne National Laboratory and housed within the Advanced Leadership Computing Facility (ALCF), Polaris serves as an example of the ongoing evolution of supercomputing systems (ANL, 2021). Polaris draws from and enhances the classical designs rooted in traditional supercomputers that have long served the scientific community. Traditional supercomputers are often characterized by their large-scale parallel processing capabilities. These machines typically harness thousands of processors to undertake tasks ranging from complex simulations to vast data analyses. Such systems are meticulously designed with specialized architectures to ensure rapid data communication between processors, efficient cooling mechanisms, and optimized power usage. They often prioritize raw computational power and memory bandwidth, utilizing vast arrays of CPUs coupled with large storage systems.

Building on this foundation, Polaris incorporates AMD EPYC CPUs and NVIDIA A100 GPUs. While the AMD EPYC CPUs attend to general-purpose computations, the NVIDIA A100 GPUs, specifically tailored for HPC, specialize in accelerating DL tasks. The system's design further underscores the importance of memory placement, ensuring data transfer efficiency and, by extension, overall performance. A modern network fabric interlinks these components, promoting consistent and smooth data exchange. In essence, Polaris offers a bridge by merging the traditional attributes of previous supercomputers with contemporary advancements, balancing versatility and raw performance. More information about Polaris, showcasing its specifications and design, can be viewed in Figure 2.

Polaris System Specs

Peak Performance	44 Petaflops (tensor core flops)
NVIDIA GPU	A100
AMD EPYC Processor	Milan
Platform	HPE Apollo Gen10+
Compute Node	1 AMD EPYC "Milan" processor; 4 NVIDIA A100 GPUs; Unified Memory Architecture; 2 fabric endpoints; 2 NVMe SSDs
GPU Architecture	NVIDIA A100 GPU; HBM stack
CPU-GPU Interconnect	CPU-GPU: PCIe; GPU-GPU: NVLink
System Interconnect	HPE Slingshot 11*; Dragonfly topology with adaptive routing
Network Switch	200 Gbps (after Slingshot-11 upgrade*)
Node Performance	78 Teraflops (double precision)
System Size	560 nodes

Figure 2. Polaris Description (ANL, 2021)

Cerebras CS-2: The Cerebras CS-2 (Lie, 2022) system is designed as an AI accelerator, with the intent of decreasing training and inference times in comparison to typical GPU-based AI solutions (Figure 3). The system houses the Cerebras Wafer-Scale Engine 2 (WSE-2), which covers an area of more than 46 cm^2.

This engine incorporates 850,000 AI-optimized compute cores, complemented by 40 GB of on-chip SRAM. The WSE-2 provides a memory bandwidth of 20 PB/s and an interconnect bandwidth of 220 Pb/s. A notable feature of the WSE-2 is its memory distribution strategy. The architecture spreads the 40 GB of on-chip SRAM across the chip, granting each core access to this high-speed memory within a single clock-cycle. This design aims to facilitate the simultaneous execution of large models and datasets on a single chip. Communications within the WSE-2 is facilitated by a communication fabric connecting the wafer's processing elements. This fabric is software-configurable, aiming to provide a high bandwidth and low latency interconnect, measured at 220 Pb/s processor-processor interconnect bandwidth. In terms of software integration, the Cerebras platform offers compatibility with PyTorch and TensorFlow, using a custom-built API. The software environment attempts to be fully programmable, and includes a library tailored for DL computations and a C-like interface for custom development. The Cerebras CS-2, leveraging the WSE-2, provides a specific approach to AI computation, focusing on its unique chip architecture as a solution to the challenges inherent to DL.

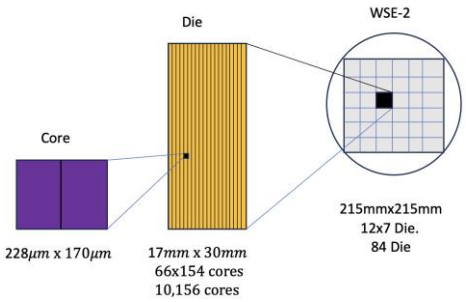

Figure 3. Cerebras Cores and Chip (Lie, 2022)

SambaNova DataScale: The SambaNova DataScale system (Emani, et al., 2021) is constructed with the Reconfigurable Dataflow Architecture (RDA), a departure from traditional computer architectures (Figure 4). A core component of this system is the SambaNova Reconfigurable Dataflow Unit (RDU). The architecture is built to work with terabytes of memory, addressing the needs of substantial data-intensive tasks. DataScale's approach emphasizes a shift from core-based designs, focusing instead on the efficient flow of data to reduce caching and extraneous data movement. In terms of software compatibility, SambaNova provides a custom software stack called SambaFlow, enabling users to continue using popular frameworks like PyTorch and TensorFlow, with some adjustments that are needed to fully adapt DL tasks to the DataScale. There is a particular emphasis placed on the system's ability to extract, optimize, and execute dataflow graphs of user models on the RDUs. This feature aims to provide scalability across systems, while maintaining consistent rack-to-rack bandwidth and latency. Overall, SambaNova DataScale provides an alternative approach to DL computation, concentrating on its distinct dataflow architecture and integration with existing DL frameworks.

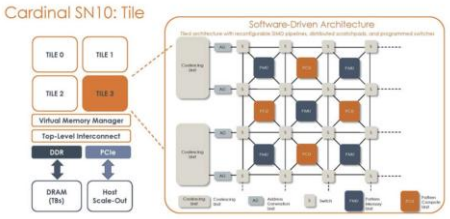

Figure 4. SambaNova RDU Chip Overview Tile (Emani, et al., 2021)

3 Results

Our comprehensive experiments on the MNIST and ImageNet-1K datasets with CNNs and ViTs across three distinct system architectures —Polaris supercomputer (ANL, 2021), SambaNova DataScale (Emani, et al., 2021), and Cerebras CS-2 (Lie, 2022)— have yielded significant insights into the quantification of uncertainty in DL models. An example output from out ViT model is depicted in Figure 5.

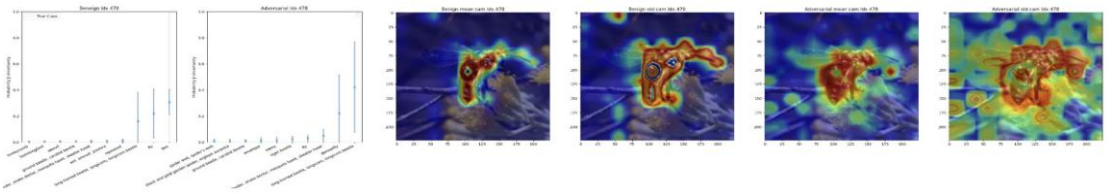

Figure 5. Error bars signify increased uncertainty during misclassification of adversarially perturbed images. Both the 'mean' and 'std' Class Activation Maps (CAMs) are associated with the bee's body, but the adversarial 'std' CAM reveals more dispersed, imprecise activations. This suggests heightened uncertainty and possible perturbations.

3.1. Uncertainty Measures

To effectively compare various uncertainty methodologies, we adopted the following measures:

1. Brier Score: Quantifying the accuracy of probabilistic predictions, the Brier Score is essentially the mean squared error of a forecast. For single outputs, it is defined as:

$$BS(p, \hat{p}) = \sum_{1}^{c} (\hat{p}_i - P_i)^2$$

2. Expected Calibration Error:

$$ECE = E_{\hat{p}}\left[|P\left(\hat{Y} = y|\hat{P} = p\right) - p|\right]$$

3.2. Uncertainty Evaluations on Polaris

Using the Polaris supercomputer, which is equipped with 4 GPUs per node, necessitated particular attention to GPU affinity. It was vital to assign each MPI rank node to a unique GPU. Otherwise, a default setting could cause all ranks to use GPU 0, leading to memory shortages. Further insights are depicted graphically in Figure 6, where the inverse relationship between computational time and the number of nodes is evident. The variations in color represent experiments using different ensemble sizes. Naturally, larger ensemble sizes require more computation time for a given node number. The results are outlined in Table 1. Table 2 and Table 3 provide additional timings for evaluations on ImageNet and MNIST datasets, respectively. For ImageNet, performance metrics were recorded based on the number of ensembles, nodes, GPUs, and execution time. Similarly, the timings for the MNIST dataset, when employing evidential learning, are presented in Table 3.

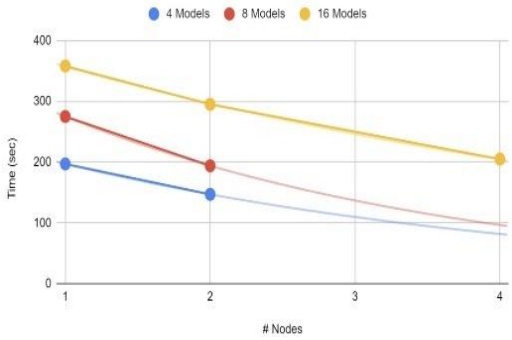

Figure 6. MNIST Dataset Uncertainty Evaluation Timings on Polaris using Ensembles. As node count increases, computation time diminishes, showcasing the scalability of our method

Table 1. MNIST Dataset Timings on Polaris: Uncertainty Evaluation using Ensembles

Number of Ensembles	Number of Nodes	Number of GPUs	Time (in sec)	Sample per sec
4	1	4	3 min 17 sec	232.077
4	2	8	2 min 27sec	232.077
8	1	4	4 min 35 sec	251.149
8	2	8	3 min 14sec	251.149
16	1	4	5 min 58 sec	252.748
16	2	8	4 min 55 sec	252.748
16	4	16	3 min 25 sec	252.748

Table 2. ImageNet-1K Dataset Timings on Polaris: Uncertainty Evaluation using Ensembles

Number of Ensembles	Number of Nodes	Number of GPUs	Time (in sec)
8	2	4	117 min 54 sec
8	8	32	28 min 21 sec

Table 3. MNIST Dataset Timings on Polaris: Uncertainty Evaluation using Evidential Learning

Number of Ensembles	Number of Nodes	Number of GPUs	Time (in sec)
1	1	1	10 min 54 sec (5 epochs)
8	8	32	56 min 30 sec (200 epochs)

The differences in uncertainty estimations between ensemble methods and evidential learning are depicted in Figure 7. The right figure, representing evidential learning, showcases higher concentrations of correct predictions at low uncertainty. However, there is notable overlap of incorrect predictions, indicating inconsistency in uncertainty estimations. In contrast, the ensemble method (left figure) offers a nuanced representation of uncertainty. The distinct gap between the mixed predictions signifies the model's proficiency in discerning when it is uncertain. To provide an even clearer perspective on uncertainty values generated by ensembles and evidential learning, we employed a Binary Classifier. This classifier, instead of predicting digits from $0-9$, determines if an image represents a specific digit or not. For instance, for the digit 1, the output would be 1 if it matches or 0 if it does not. Figure 8 offers a visual representation and interpretation of this binary evidential and ensemble approach, using a challenging MNIST 4 sample. Figure 9 provides an illustrative example from the ImageNet dataset, evaluated using ensemble learning on Polaris. While there is a high SoftMax probability prediction, the significant uncertainty (indicated by long tails on the prediction) suggest potential inaccuracies.

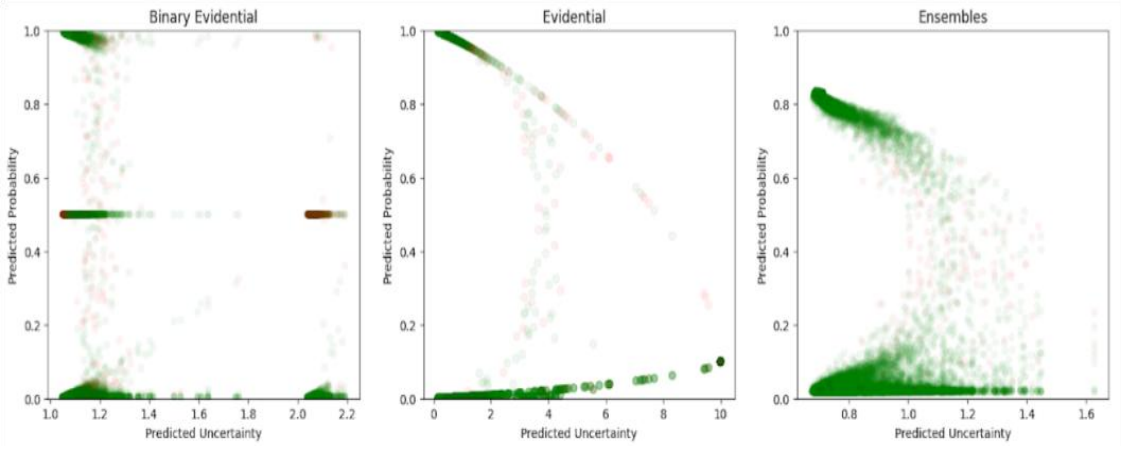

Figure 7. Comparison of probability/uncertainty between Ensembles and Evidential Learning Uncertainty Evaluation

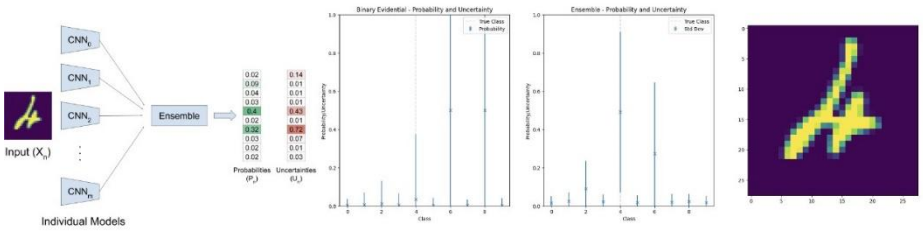

Figure 8. Left: Aggregation of ensemble logits into a single probability distribution, with the standard deviation denoting the model's uncertainty. Right: Binary Evidential Classifier comparison for digit 4 in MNIST

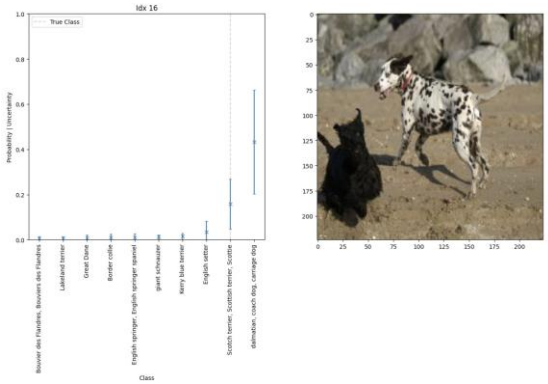

Figure 9. Ensembles. Probability and uncertainty for index 16 in ImageNet validation set. High probability, but also high uncertainty. Prediction is incorrect, predicted Dalmatian, actually Scottish Terrier. Image sample from the ImageNet Database (Deng, et al., 2009)

3.3. Uncertainty Evaluations on Cerebras CS-2

Our experiments with the Cerebras CS-2 offered valuable insights into the practical nuances of adopting this novel architecture for DL workflows.

Adapting Established Workflows: Migrating to the Cerebras system required notable adjustments, particularly in data handling. The system's attempt to retain existing DL workflows (Lie, 2022) provided direction during our adaptation phase. Nonetheless, one imperative aspect was ensuring the right data sharding across the allocated worker nodes, especially in scenarios involving distributed settings.

Neural Network Architectural: Considerations While we experienced success in implementing CNNs on the CS-2, our attempts to migrate ViTs did not yield the expected results. Despite thorough reviews of our codebase and implementation, the exact cause remains elusive. This observation underscores the importance of understanding the architectural nuances and potential pitfalls or challenges when transitioning certain NN architectures to the CS-2.

Scalability and Physical Advantages: The CS-2, with its massive computational prowess, presents its own set of considerations when assessing its alignment with specific scalability demands, especially when contrasted against platforms like Polaris and SambaNova. Conversely, one of CS-2's distinct advantages is its minimal physical footprint, which presents potential space and energy savings–a consideration that may appeal to those aiming to optimize their infrastructure.

3.4. Uncertainty Evaluation on SambaNova DataScale

Having introduced the SambaNova DataScale system and its RDA in Section 2.2, our experiences further elaborate on its practical implications in the sphere of uncertainty evaluation for DL.

Workflow Differences: SambaNova's departure from traditional architectures, as captured by its emphasis of efficient data flow and reduction of caching and data movement, has led to tangible workflow distinctions. Central to this is the need to optimize dataflow graphs for the RDUs. As experienced, this demand generating a computational map that eventually compiles into a Plasticine Executable Format (.pef) file. A significant point of consideration is the time commitment: compile duration could sometimes surpass 40 minutes. Additionally, SambaNova's approach requires a closer integration of the loss function within the model's class definition, while adjusting the forward function to return both loss and outputs.

Scaling and Parallelization Capabilities: Our experiments demonstrate SambaNova's parallelization and scalability capabilities with consistent performance. Though ensemble modeling was not directly supported in our setup, we successfully harnessed multiple nodes for parallel training. Specifically, our configurations were able to span a maximum of seven nodes, each equipped with eight RDUs, cumulatively activating 56 MPI ranks, with eight ranks delegated to each node. This demonstrated SambaNova's adeptness in balancing computational demands.

Integration and Transition: The software transition to SambaNova DataScale does require certain adjustments, but it is achievable due to SambaFlow's compatibility with standard DL frameworks, such as PyTorch. Despite the needed modifications, most of PyTorch's familiar structure was kept. This was indicative of a potentially smoother onboarding for users accustomed to conventional DL paradigms.

4 Conclusion

Our exploration dives deep into characterizing uncertainty within Neural Networks using various HPC systems. We highlighted the unique features, strengths, and challenges of implementing uncertainty evaluations on Polaris, SambaNova DataScale, and Cerebras CS-2 (Figure 10). While Polaris showcases impressive scalability, SambaNova DataScale, with its distinctive dataflow architecture, offers rapid acceleration and flexibility in parallelization for single-model training. Cerebras, on the other hand, impresses with its commendable performance on a singular node, though with limitations for specific model types. Our engagements with these platforms underscore the relevance and adaptability of uncertainty evaluations in DL models. Using datasets like MNIST and ImageNet, we've illustrated how uncertainty inherent in image classifications can be synthesized to produce holistic confidence distributions. For scenarios where computational time might be a constraint, our results suggest that evidential learning is a promising route, offering critical uncertainty insights with minimal computational demands.

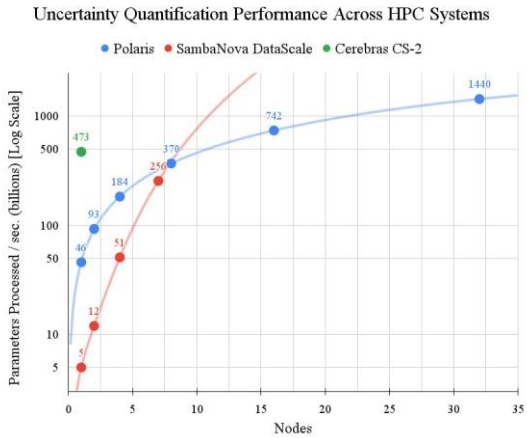

Figure 10. Performance Metrics of Uncertainty Quantification Across Various HPC Platforms. Comparing the Parameters processed per second (PPS) across different node counts for Polaris, SambaNova DataScale, and Cerebras CS-2: Polaris exhibits consistent linear consistent linear scaling, SambaNova shows swift acceleration at lower node counts, and Cerebras demonstrates an impressive PPS even with one node.

5 Future Work

Future endeavors will focus on refining our code distribution strategy via MPI, transitioning to a more dynamic task allocation for nodes. We're also curious about the potential of uncertainty evaluations on black-box models. The present approach necessitates a known model for hyperparameter adjustments, ensemble formulations, or loss function modifications for evidential learning. Can uncertainty be effectively gauged when the model remains undisclosed? While this area remains relatively uncharted, we're eager to pioneer research and experiment with diverse hypotheses to ascertain its practicality.

Acknowledgements

Part of this research was supported by Sustainable Horizons Institute which is part of the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration and by Argonne National Laboratory. This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

Appendix A - Availability of data and materials

Access our experimental code at https://github.com/epautsch/UncertaintyANL

ORCID ID

Erik Pautsch https://orcid.org/0000-0003-0028-5598
John Li https://orcid.org/0000-0002-3730-3713
Silvio Rizzi https://orcid.org/0000-0002-3804-2471
George K. Thiruvathukal https://orcid.org/0000-0002-0452-5571
María Pantoja https://orcid.org/0000-0002-1942-9769

References

Amini, A., Schwarting, W., & Rus, D. (2020, December 6). Deep evidential regression. In H. Larochelle, M. Ranzato, R. T. Hadsell, M. F. Balcan, & H. Lin (Eds.), *NIPS'20: 34th International Conference on Neural Information Processing Systems, Vancouver BC, Canada, December 6-12,* (pp. 14927-14937, Article 1251). Red Hook, NY, USA: Curran Associates Inc. doi:10.5555/3495724.3496975

ANL. (2021, August 26). *Polaris*. (Argonne National Laboratory) Retrieved July 2023, from ANL website: https://www.alcf.anl.gov/polaris

Bojarski, M., Yeres, P., Choromanska, A., Choromanski, K., Firner, B., Jackel, L., & Muller, U. (2017, April 25). Explaining how a deep neural network trained with end-to-end learning steers a car. *arXiv:1704.07911v1 [cs.CV]*, 1-8. doi:10.48550/arXiv.1704.07911

Cordonnier, J.-B., Loukas, A., & Jaggi, M. (2020). On the relationship between selfattention and convolutional layers. *Eighth International Conference on Learning Representations - ICLR 2020, April 26-30.* Addis Ababa. Retrieved from https://infoscience.epfl.ch/entities/publication/48815b9c-e947-4c4d-84fa-7ebf1f6df4dd/conferencedetails

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Li, F.-F. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition, 20-25 June* (pp. 248-255). Miami, FL, USA: IEEE. doi:10.1109/CVPR.2009.5206848

Emani, M., Vishwanath, V., Adams, C., Papka, M. E., Stevens, R., Florescu, L., . . . Sujeeth, A. (2021, March 26). Accelerating scientific applications with sambanova reconfigurable dataflow architecture. *Computing in Science & Engineering, 23*(2), 114–119. doi:10.1109/MCSE.2021.3057203

Gal, Y., & Ghahramani, Z. (2016, June). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In M. F. Balcan, & K. Q. Weinberger (Ed.), *Proceedings of The 33rd International Conference on Machine Learning. 48*, pp. 1050-1059. New York, New York, USA (20–22 Jun 2016): PMLR. Retrieved from https://proceedings.mlr.press/v48/gal16.html

Geifman, Y., & El-Yaniv, R. (2017, December 4). Selective classification for deep neural networks. In U. von Luxburg, I. M. Guyon, S. Bengio, H. M. Wallach, & R. Fergus (Eds.), *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA, December 4 - 9, 2017* (pp. 4885-4894). Red Hook, NY, USA: Curran Associates Inc. doi:10.5555/3295222.3295241

Guo, C., Pleiss, G., Sun, Y., & Weinber, K. Q. (2017). On calibration of modern neural networks. In D. Precup, & Y. W. Teh (Ed.), *Proceedings of the 34th International Conference on Machine Learning. 70*, pp. 1321-1330. PMLR. Retrieved from https://proceedings.mlr.press/v70/guo17a.html

Hendrycks, D., Liu, X., Wallace, E., Dziedzic, A., Krishnan, R., & Song, D. (2020, July). Pretrained transformers improve out-of-distribution robustness. In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault (Eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 2744–2751). Online: Association for Computational Linguistics. doi:10.18653/v1/2020.acl-main.244

Lecun, Y., Jackel, L. D., Bottou, L., Cortes, C., Denker, J. S., Drucker, H., . . . Vapnik, V. (1995). Learning algorithms for classification: A comparison on handwritten digit recognition. In J. H. Oh, C. Kwon, & S. Cho (Eds.), *Learning algorithms for classification: A comparison on handwritten digit recognition* (pp. 261-276). World Scientific. Retrieved from https://nyuscholars.nyu.edu/en/publications/learning-algorithms-for-classification-a-comparison-on-handwritte

Lie, S. (2022). Cerebras architecture deep dive: First look inside the hw/sw co-design for deep learning. In *2022 IEEE Hot Chips 34 Symposium (HCS), 21-23 August* (pp. 1–34). Cupertino, CA, USA: IEEE. doi:10.1109/HCS55958.2022.9895479

Liu, Y., & Guo, H. (2020, July 13). Peer loss functions: Learning from noisy labels without knowing noise rates. In H. C. Daumé, & A. Singh (Eds.), *ICML'20: International Conference on Machine LearningJuly 13 - 18* (Vols. 119, Article 578, pp. 6226–6236). JMLR.org.

MacDonald, S., Foley, H., Yap, M., Johnston, R. L., Steven, K., Koufariotis, L. T., . . . Trzaskowski, M. (2023, May 6). Generalising uncertainty improves accuracy and safety of deep learning analytics applied to oncology. *Scientific Reports, 13*, 7395. doi:10.1038/s41598-023-31126-5

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., . . . Snoek, J. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 32). Red Hook, NY, USA: Curran Associates Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2019/file/8558cb408c1d76621371888657d2eb1d-Paper.pdf

Ren, A. Z., Dixit, A., Bodrova, A., Singh, S., Tu, S., Brown, N., . . . Majumdar, A. (2023, September 4). Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv:2307.01928v2 [cs.RO]*, 1-24. doi:10.48550/arXiv.2307.01928

Tamkin, A., Nguyen, D., Deshpande, S., Mu, J., & Goodman, N. (2022). Active learning helps pretrained models learn the intended task. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in Neural Information Processing Systems* (Vol. 35, pp. 28140-28153). Curran Associates Inc. Retrieved from

https://proceedings.neurips.cc/paper_files/paper/2022/file/b43a0e8a35b1c044b18cd843b9771915-Paper-Conference.pdf

Wenzel, F., Snoek, J., Tran, D., & Jenatton, R. (2020, Decembre 6). Hyperparameter ensembles for robustness and uncertainty quantification. In H. Larochelle, M. Ranzato, R. T. Hadsell, M. F. Balcan, & H. Lin (Eds.), *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, December 6 - 12, 2020* (pp. 6514–6527). Red Hook, NY, USA, Article 546: Curran Associates Inc. doi:10.5555/3495724.3496270