Evaluating Deep Learning Uncertainty Quantification Methods for Neutrino Physics Applications

Dae Heun Koh
Stanford University
Stanford, CA 94305
koh0207@stanford.edu

Aashwin Mishra
SLAC National Accelerator Laboratory
Menlo Park, CA 94025
aashwin@slac.stanford.edu

Kazuhiro Terao
SLAC National Accelerator Laboratory
Menlo Park, CA 94025
kterao@slac.stanford.edu

Abstract

We evaluate uncertainty quantification (UQ) methods for deep learning applied to liquid argon time projection chamber (LArTPC) physics analysis tasks. As deep learning applications enter widespread usage among physics data analysis, neural networks with reliable estimates of prediction uncertainty and robust performance against overconfidence and out-of-distribution (OOD) samples are critical for its full deployment in analyzing experimental data. While numerous UQ methods have been tested on simple datasets, performance evaluations for more complex tasks and datasets have been scarce. We assess the application of selected deep learning UQ methods on the task of particle classification in a simulated 3D LArTPC point cloud dataset. We observe that uncertainty enabled networks not only allow for better rejection of prediction mistakes and OOD detection, but also generally achieve higher overall accuracy across different task settings. We also conclude that in in most settings, simple ensembling methods are sufficient in obtaining well calibrated classification probabilities and generally achieve higher overall accuracy.

1 Introduction & Motivation

When a neutrino enters a liquid argon time projection chamber (LArTPC) [32], it reacts with the argon nucleus and produces a unique set of charged particles. Ionization electrons originating from each charged particle's interaction with the medium are drifted towards three wire planes that provide different projections of the full 3D trajectory. The high resolution projection images provided by the LArTPC allow the full three-dimensional assembly of the neutrino interaction as a 3D point cloud image. Determining the types of all final state charged particles is critical for inferring physics behind neutrino events, as the neutrino itself is not directly visible by the LArTPC detector. Hence, designing automated event selection algorithms for neutrino experiments includes developing tools for solving common tasks in computer vision, such as image classification and semantic segmentation. Combined with the large amount of data that accelerator based neutrino experiments such as the Short Baseline Neutrino (SBN) experiment [26] and the Deep Underground Neutrino Experiment (DUNE) [2] offer, deep learning applications [3, 9] have been particularly fruitful.

Using deep learning for fundamental research, however, presents complications that are often omitted in many common industrial use cases, where practitioners generally attend to achieving state-of-the-art with respect to a family of conventional performance metrics. In particular, one of the most

6th Workshop on Bayesian Deep Learning (NeurIPS 2021), Sydney, Australia.

pressing issues with using deep neural networks for fundamental research is developing robust and consistent methods for quantifying uncertainties of its predictions. While deep neural networks are often treated as high-fidelity models, even predictions from an optimally trained neural network may contain significant errors and uncertainties. Neural networks are prone to making overconfident yet erroneous predictions. Our limited understanding of neural network mappings [39], coupled with the difficulty in verification and validation for complex deep neural network based models [20], can lead to unforeseen consequences. Overconfidence for out-of-distribution examples also demonstrate the need for deep learning models to acknowledge whether a given prediction is to be trusted or not. For deep neural nets to be integrated into the physics measurement process, such characteristics of deterministic neural networks must be addressed by an effective method for uncertainty quantification.

As demand for UQ gradually escalated in domains such as automated driving and medicine, UQ methods diversified into a variety of different approaches under the cognomen of Bayesian Deep Learning (BDL), but with scarce substantial application in the physical sciences. Moreover, most BDL methods have been benchmarked on oversimplified datasets (MNIST, CIFAR10), which are not representative of the complexity of physics data analysis process. Modern accelerator neutrino experiments such as ICARUS and DUNE provide ideal grounds for testing the efficacy of BDL in UQ, due to its recent adaptation and moderate success of deep learning based analysis techniques. The benefit derived from a detailed assessment of different UQ algorithms on a complex, multi-objective task such as LArTPC data analysis is two-fold and symbiotic: allow practitioners in machine learning to evaluate BDL's applicability in a real-world setting and enable physicists to design neural network that produce well justified uncertainty estimates for rejecting erroneous predictions and detecting out-of-distribution instances.

In this paper, we select different approaches for quantifying uncertainties in deep learning, and evaluate them with respect to critical intermediate analysis tasks: particle classification and semantic segmentation. We consider three UQ methods–model ensembling, Monte Carlo Dropout [14], and Evidential Deep Learning [34, 6] (EDL)–and evaluate them on single particle classification, semantic segmentation, and multi-particle classification.[1] Using sparse convolutional neural networks provided by the MinkowskiEngine [8] library, we build a ResNet [19] type convolutional image classifier as a backbone network for single particle classification. For semantic segmentation, we employ the UResNet [31, 9] architecture with minor modifications in line with [22]. For multi-particle classification, we model the task by a combination of a graph neural network with node vectors defined by geometric features extracted from particles. In evaluating uncertainty quantification fidelity, we focus on two aspects: calibration and sensitivity to classification errors. For calibration, we use the standard tools of reliability curves and expected calibration error [27, 28] to compare different models. For mis-classification sensitivity, the different UQ methods are assessed by measuring the area under the receiver operating characteristic curve (AUROC) and the 1-Wasserstein distance [29] between the correctly and incorrectly predicted distributions.

2 Methods of Uncertainty Quantification in Deep Learning

Among numerous models and studies on uncertainty-quantifying neural networks [1, 36], we focus on methods designed for multi-class classification tasks that require minimal changes to popular neural network architectures. In this paper, we consider three class of UQ methods: model ensembling [24], Monte Carlo Dropout (MCD) [13], and Evidential Deep Learning (EDL) [33, 5].

For the following discussion, let $X = \{x^{(1)}, x^{(2)}, ..., x^{(n)}\}$ and $Y = \{y^{(1)}, y^{(2)}, ..., y^{(N)}\}$ be data and labels in the training set, and let $\tilde{X} = \{\tilde{x}^{(1)}, \tilde{x}^{(2)}, ..., \tilde{x}^{(M)}\}$ and $\tilde{Y} = \{\tilde{y}^{(1)}, \tilde{y}^{(2)}, ..., \tilde{y}^{(M)}\}$ denote the test set. A neural network f_θ, parametrized by weights θ, is trained on $D_{train} = \{(x^{(1)}, y^{(1)}), ..., (x^{(N)}, y^{(N)})\}$, with logits given by $z^* = f_\theta(x^*; X, Y)$ and labels $\hat{y}^* = argmax_c(f_\theta(x^*; X, Y)_1, ..., f_\theta(x^*; X, Y)_c)$, for some $x^* \in X^* \subset \tilde{X}$.

[1] *Single particle classification* refers to image classification with samples that contain only one isolated particle instance. In *Multi-particle classification*, each image contains one or more particle instances from (possibly) different classes, and the model is trained to predict class labels for all particles that populate a given image.

2.1 Ensembing Methods

Model ensembling [11] in the context of deep learning models refers to the method of training multiple instances of the same architecture with different random initialization seeds. In Naive Ensembling (NE), one trains each member of the ensemble on the same training dataset, resulting in N networks with identical architecture but different parameter values. Often, to achieve better generalization and stability, Bootstrapped Ensembling (BE) (or bagging) is preferred over naive ensembling. This is done by training each ensemble member on a dataset reorganized by sampling N examples from the full training set with replacement. If the size of the resampled dataset is equal to that of the original training set, each ensemble member is expected to see approximately 63% of the original training set. For classification, it is standard to use the most common label among the ensemble members as the final prediction, while for regression one usually computes the empirical mean. When an ensemble consists of a collection of neural networks trained with respect to a *proper scoring rule* [15] and often coupled with an optional adversarial training routine, the ensemble is termed *deep ensembles* [24].

Ensemble methods are the one of the simplest UQ methods that require no additional changes to the underlying model architecture, although the high computational cost in training N architecturally identical models and performing N forward passes for one prediction often renders them inapplicable for some memory or time consuming tasks.

2.2 Monte Carlo Dropout

Monte-Carlo Dropout is a bayesian technique introduced in [13], where one approximates the network's posterior distribution of class predictions by collecting samples obtained from multiple forward passes of dropout regularized networks. *Dropout regularization* [35] involves random omissions of feature vector dimension during train time, which is equivalent to masking rows of weight matrices. Inclusion of dropout layers mitigates model overfitting and is empirically known to improve model accuracy [35]. A key observation of [13] is that under suitable assumptions on the bayesian neural network prior and training procedure, sampling N predictions from the BNN's posterior is equivalent to performing N stochastic forward passes with dropout layers fully activated. This way, the full posterior distribution may be approximated by monte-carlo integration of the posterior softmax probability vector $p(\hat{y}^* \mid x^*; X, Y)$:

$$p(\hat{y}^* \mid x^*; X, Y) \approx \frac{1}{T} \sum_{t=1}^{T} Softmax(\mathbf{f}_{\theta_t}(x^*; X, Y)), \tag{1}$$

where T denotes the number of stochastic forward passes. As with ensembling methods, the final prediction of MCDropout for classification is given by the majority vote among all stochastic forward passes. As evident from the apparent similarities, MCDropout networks may also be interpreted as a form of ensemble learning [35], where each stochastic forward pass corresponds to a different realization of a trained neural network.

Implementing MCDropout requires one to modify the underlying neural network architecture to include dropout layers and configuring them to behave stochastically during test time. Often the location of dropout layers can critically affect prediction performance, and for convolutional neural networks the decision is made via trial-and-error [21]. Also, for memory intensive tasks such as semantic segmentation, sample collection by multiple forward passes can accumulate rapidly towards high compuational cost, similar to ensembling methods.

2.3 Evidential Deep Learning

Evidential Deep Learning (EDL) [33, 5], refers to a class of deep neural networks that exploit conjugate prior relationships to model the posterior distribution analytically. For multi-class classification, the distribution over the space of all probability vectors $\mathbf{p} = (p_1, ..., p_c)$ is modeled by a Dirichlet distribution with c concentration parameters $\alpha = \alpha_1, ..., \alpha_c$:

$$D(\mathbf{p} \mid \alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{c} p_i^{\alpha_i - 1}, \tag{2}$$

where $\alpha_i \geq 1$ for all i, $B(\cdot)$ denotes the c-dimensional multinomial Beta function, and $\mathbf{p}$ is in the c-unit simplex $\mathcal{S}_c$:

$$\mathcal{S}_c = \{\mathbf{v} \in \mathbb{R}^c : \sum_{i=1}^{c} v_i = 1\}. \tag{3}$$

In constrast to deterministic classification neural networks that minimize the cross-entropy loss by predicting the class logits, evidential classification networks predict the concentration parameters $\alpha = (\alpha_1, ..., \alpha_c)$. The expected value of the k-th class probability under the distribution $D(\mathbf{p} \mid \alpha)$ is then given analytically as

$$\hat{p}_k = \frac{\alpha_k}{S}, \quad S = \sum_{i=1}^{c} \alpha_i. \tag{4}$$

To estimate the concentration parameters, several distinct loss functions are available as training criteria. The *marginal likelihood loss* (MLL) is given by:

$$\mathcal{L}_{MLL}(\theta) = -\log\left(\int \prod_{i=1}^{c} p_i^{y_i} D(\mathbf{p} \mid \alpha)\, d\mathbf{p}\right). \tag{5}$$

The *Bayes risk* (posterior expectation of the risk) of the *log-likelihood* (BR-L) formulation yields:

$$\mathcal{L}_{BR}(\theta) = \int \left[\sum_{i=1}^{c} -y \log\left(p_i\right)\right] D(\mathbf{p} \mid \alpha)\, d\mathbf{p}. \tag{6}$$

The *Bayes risk* of the *Brier score* (BR-B) may also be used as an alternative optimization objective:

$$\mathcal{L}_{BS}(\theta) = \int ||\mathbf{y} - \mathbf{p}||_2^2 \, D(\mathbf{p} \mid \alpha)\, d\mathbf{p}. \tag{7}$$

From Sensoy et. al. [33], analytic integration of the aforementioned loss functions give closed form expressions that are suited for gradient based optimization of the parameters θ.

EDL methods have the immediate advantage of requiring only one single pass to access the full posterior distribution, at a price of restricting the space of posterior functions onto the appropriate conjugate prior forms. Also, EDL methods only require one to modify the loss function and the final layer of its deterministic baseline (if necessary), which allows flexible integration with complex, hierarchical deep neural architectures similar to the full LArTPC reconstruction chain. However, due to the strong assumptions made on the posterior analytical form, EDL methods are limited to classification and regression tasks as of now. As we later observe, EDL methods generally fall short on various UQ evaluation metrics compared to ensembling and MCDropout, depending on task specifics.

3 Evaluating Uncertainty Quantification Methods

3.1 Evaluation Metrics

As stated in [24], the goal of uncertainty quantifying models is two-fold: to achieve better alignment of predicted confidence probability with their long-run empirical accuracy and to serve as mis-classification or out-of-distribution alarms that could be used for rejecting unconfident predictions. The first condition, which we term *calibration fidelity*, may be evaluated by plotting the *reliability diagrams* [17] constructed by binning the predicted probabilities (often termed *confidence*) into equal sized bins and plotting the bin centers in the x-axis and the empirical accuracy of the bin members in the y-axis. The closer the reliability diagram is to the diagonal, the more desirable a given classifer is, in the sense of calibration fidelity. The deviation of a given classifier from the diagonal could be summarized by computing the *adaptive calibration error* (ACE) [28]:

$$ACE = \frac{1}{K}\frac{1}{R}\sum_{k=1}^{K}\sum_{r=1}^{R} |acc(r,k) - conf(r,k)|. \tag{8}$$

Here, K denotes the number of unique classes and R denotes the number of equal-sample bins used to plot the reliability diagram for class k, given by confidence $conf(r, k)$ and corresponding empirical accuracy $acc(r, k)$. Although the *expected calibration error* (ECE) [27] is more widely known, we observed in practice that static binning schemes such as ECE is suboptimal for highly skewed predictive probability distributions common to accurate models.

As calibration fidelity measurements using reliability diagrams are originally designed for binary classifiers, there has been numerous proposals for their extensions to multi-class classifiers [16, 37, 38]. We consider two relatively simple methods; the first is a standard used in Guo et. al. [16], where only the predicted probability for the most confident prediction of each sample is used to plot the reliability diagram. We refer to this mode of assessment as *max-confidence* calibration fidelity. An alternative method is to evaluate calibration for each of the K classes separately, as in B. Zadrozny and C. Elkan [38]. We refer to this mode as *marginal* calbiration fidelity.

Another metric of uncertainty quantification measures the model's *discriminative capacity* to misclassified or out-of-distribution samples. In practice, uncertainty quantification models have the capacity to reject predictions based on a numerical estimate of the trustworthiness of the prediction in question. For example, in a classification setting the entropy of the predicted softmax probability distribution (*predictive entropy*) could be used as a measure of confusion, as entropy is maximized if the predictive distribution reduces to a uniform distribution over K classes. In this construction, it is desirable to have the predicted entropy distributions of correctly and incorrectly classified samples to be as separated as possible. To compute the extent of distributional separation, we may use the first Wasserstein distance [30] between the predictive entropy distributions:

$$W_1(u, v) = \inf_{\pi \in \Gamma(u,v)} \int_{\mathbb{R}\times\mathbb{R}} |x - y| \, d\pi(x, y). \tag{9}$$

where u and v are two probability distributions, $\Gamma(u, v)$ is the set of all joint probability measures in $\mathbb{R}^2$. We use the Wasserstein distance with the L_1 metric due to its simple computational implementation [30].

Discriminative capacity may also be measured by computing the area under the receiver operating characteristic curve (AUROC), also known as the concordance statistic (c-statistic) [18]. Using predictive entropy as the thresholding value, the ROC curve is constructed by plotting the false positive rate (incorrect predictions) in the x-axis and the true positive rate (correct predictions) in the y-axis at different threshold levels. In this setting, the AUROC is the probability that a randomly chosen correct prediction will have a lower predictive entropy than that of a randomly chosen incorrect prediction [12].

4 Datasets and Network Architectures

Single Particle Classification: We first implement and assess the different UQ models on the simpler task of single particle classification. The single particle dataset consists of $1024 \times 1024 \times 1024$ 3D images each containing only one particle, where all voxels in the given image belong to the same particle ID. The 3D images have one feature dimension corresponding to the amount of energy deposited in a one-voxel equivalent region of the detector. We use a ResNet [19] type encoder with dropout [35] regularization, where convolution operations are substituted by sparse convolutions implemented in the *MinkowskiEngine* library [7]. For standard deterministic models, ensembles, and MCDropout, the final prediction probabilities are given by softmax activations, whereas for evidential models the concentration parameters α are computed from Softplus [40] activations. The single particle dataset contains five particle classes: photon showers (γ), electron showers (e), muons (μ), pions (π), and protons (p).

Semantic Segmentation As segmentation is a classification task on individual pixels, the details of the implementation are mostly identical to those of single particle classification. We employ *Sparse-UResNet* [10] with dropout layers in the deeper half of the network as the base architecture for semantic segmentation networks and use the 768px resolution PILArNet [4] MultiPartRain (MPR) and MultiPartVertex (MPV) datasets for multiple particle datasets. The five semantic labels provided by PiLArNet consists of the following:

- Shower Fragments: connected components of electromagnetic showers that are above a set voxel count and energy deposition threshold.

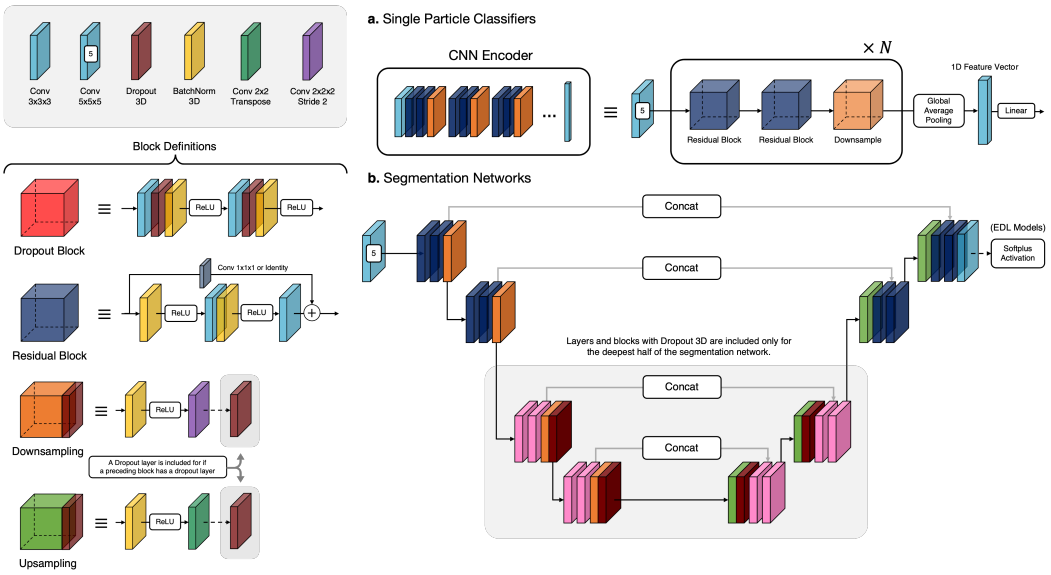

Figure 1: Sparse-CNN architecture for single particle and segmentation neural networks.

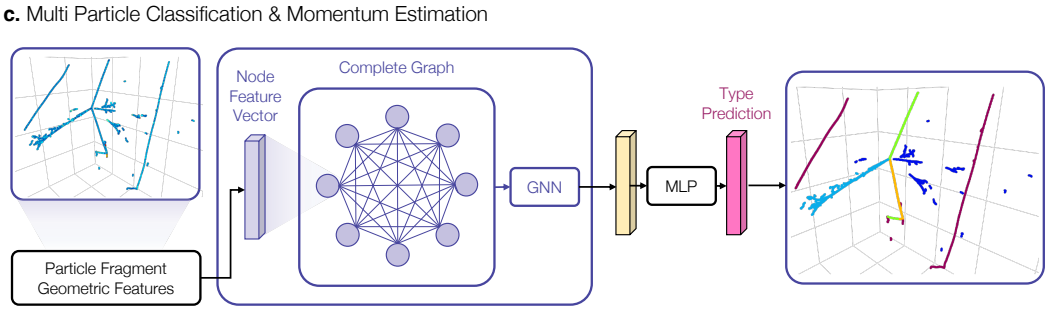

Figure 2: Architecture outline of multi-particle classification network. The geometric node encoder extracts hand-engineered features relevant to particle classification, such as orientation matrix and major PCA axes.

- Tracks: particle trajectories that resemble stright lines, mostly originating from muon, pion, and protons.
- Michel Electrons: an electron produced by muon decay at rest.
- Delta Rays: electrons produced from muon tracks via hard scattering
- Low Energy Depositions: cloud of low energy depositions of electromagnetic showers which are not labeled as shower fragments.

Multi Particle Reconstruction The MPV/MPR dataset also contains particle type labels for each particle instance in a given image. For multi particle classification, we take each cluster of voxels that belong to the same particle and reduce the resulting groups of point cloud into 1-dimensional feature vectors. The node embeddings of each particle consists of geometric features such as its principal component vectors. These feature vectors are then given as input node features to a graph neural network, which performs three message passing operations to incorporate inter-particle relational information.

5 Results

5.1 Training Details

The training set consists of 80k images, and the test set were separated to a 2k validation set used for model selection and a 18k test set used for selected model evaluation with high statistics. All models were trained until the validation accuracy plateaued, and the network weights that achieved the highest validation accuracy were selected for further evaluation on a separate test set. To fully account for possible variations in model accuracy and uncertainty quantification quality due to randomized factors such as parameter initialization, the model selection procedure were repeated for five different random seeds for each model. This results in five independently trained models that share the same architecture but differing in parameter values. We used the Adam optimizer [23] with decoupled weight decay [25].

5.2 Single Particle Classification

Figure 3 shows the predictive entropy distribution, accuracy, and the W_1 distance for single particle classification models. We observe that the distributional separation as measured in W_1 is largest for the ensemble methods, while evidential model trained on the Brier score is also competitive. In general, ensemble methods achieve highest accuracy with better distributional separation compared to monte-carlo dropout and evidential models. The AUROC values in figure 9 also reflect the superior discriminative capacity of ensembling.

The calibration curves for single particle classification is shown in the top row of figure 12, and figure 5 illustrates the adaptive calibration error (ACE) values across different subsets of the test set partitioned by true particle id labels. While all UQ models with the possible exception of EDL-BR-B achieve better calibration compared to standard deterministic neural networks, ensembling methods have the least max-confidence and marginal ACE values.

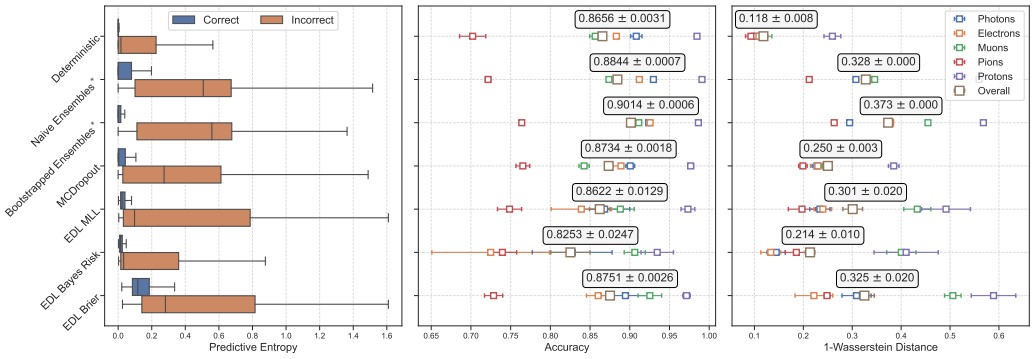

Figure 3: Uncertainty quantification for single particle classification.

5.3 Semantic Segmentation

For segmentation, the best distributional separation is achieved by evidential models, which are evident in Figure 4. The ensemble methods have the highest accuracy and AUROC scores (Figures 7, 11). It is interesting to note that while distributional separation measured in W_1 is greatest for evidential models, the calibration fidelity falls short even with respect to standard deterministic models. As with single particle classification, the best calibration fidelity is realized by ensemble methods. The reliability plots for semantic segmentation is shown in Figure 13.

5.4 Multi Particle Reconstruction

Since contextual information which are useful in determining a given particle's ID can only be used in a multi-particle setting, we expect a gain in accuracy from the single particle datasets. This approach leads to an overall approximate 5% increase in classification accuracy in all models. Again, ensemble

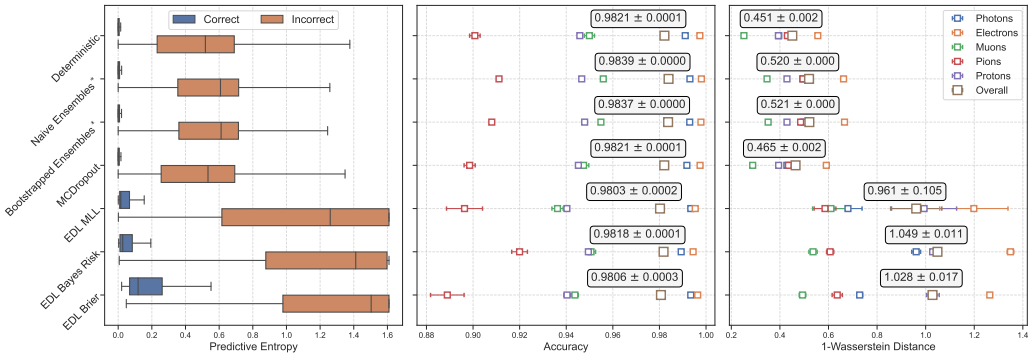

Figure 4: Uncertainty quantification for Semantic Segmentation.

methods provide the highest W_1 distance, overall accuracy, and AUROC values (figures 8, 10) and the best calibration fidelity (figure 6).

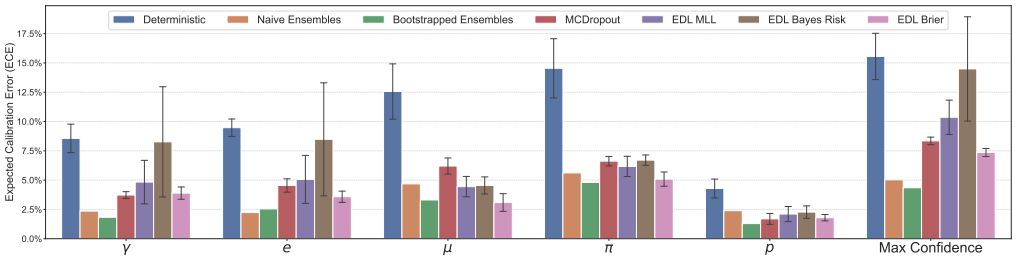

Figure 5: Single particle convolutional neural network classifier adaptive calibration errors (ACEs) for each model and class.

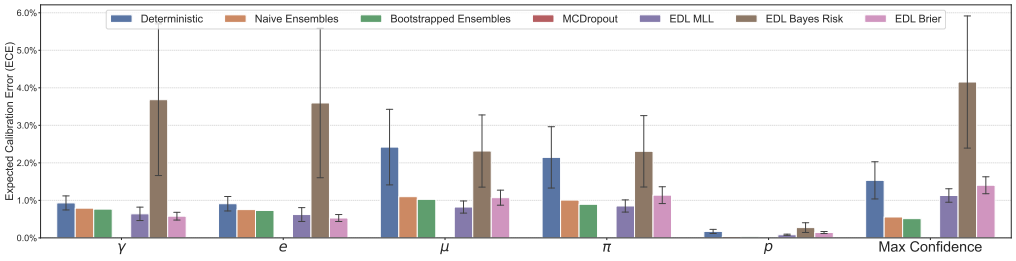

Figure 6: Multi-particle graphical neural network classifier adaptive calibration errors (ACEs) for each model and class.

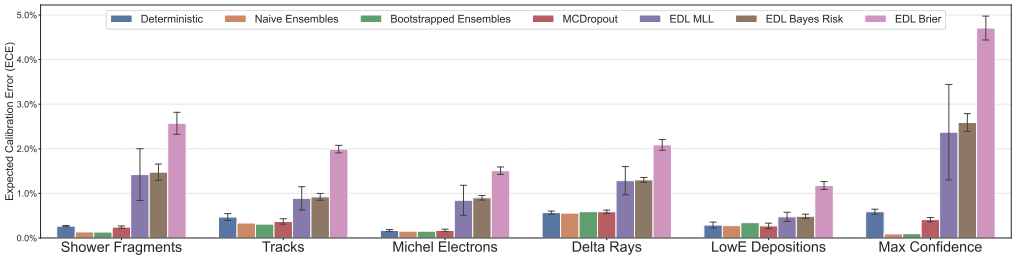

Figure 7: Semantic segmentation convolutional neural network adaptive calibration errors (ACEs) for each model and class.

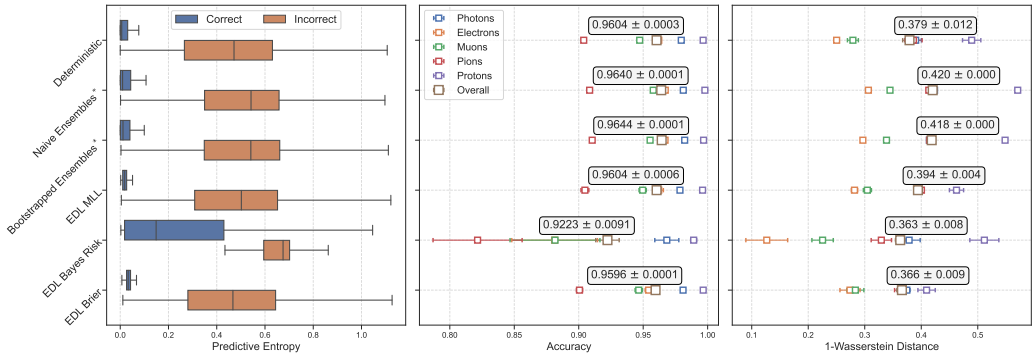

Figure 8: Uncertainty quantification for Multi Particle Classification with GNNs.

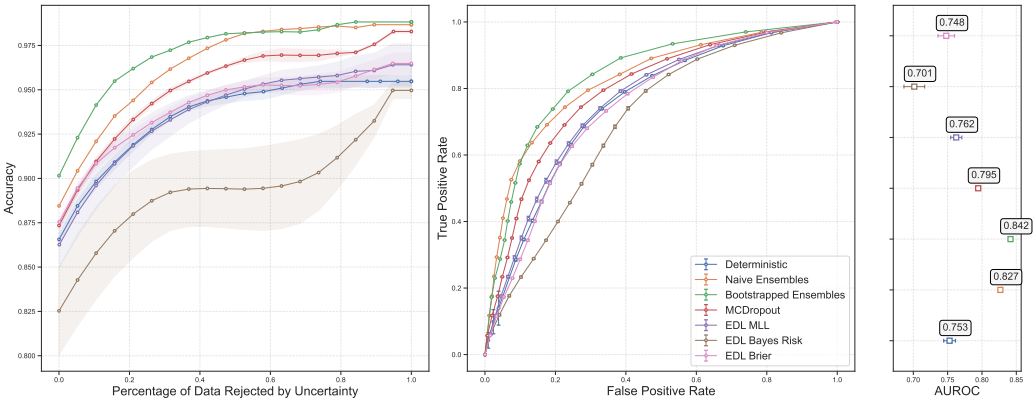

Figure 9: Single particle ROC and percentage rejection curves.

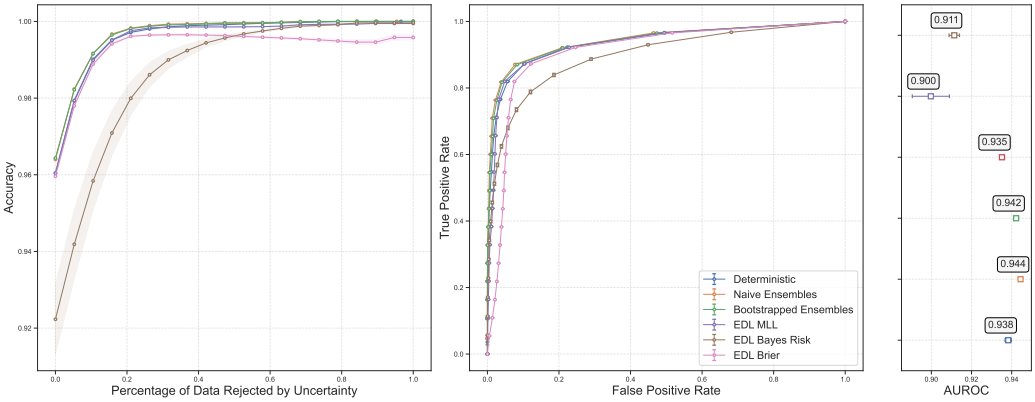

Figure 10: Multi particle ROC and percentage rejection curves.

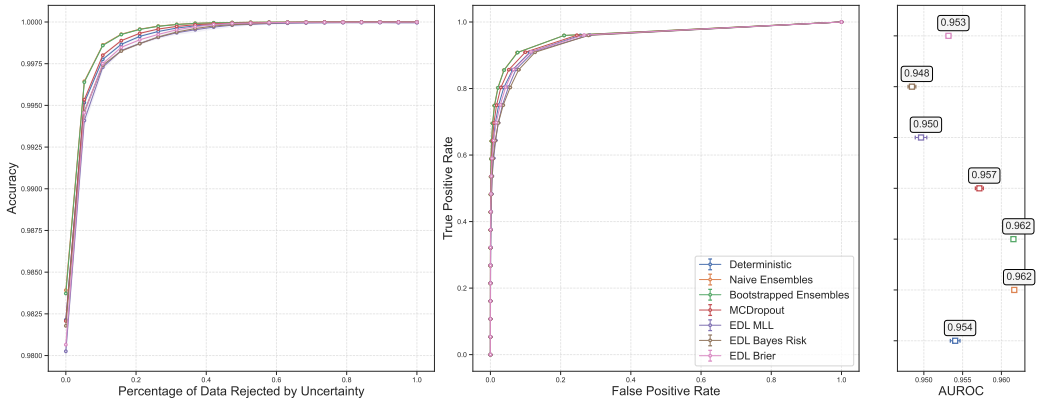

Figure 11: Semantic segmentation ROC and percentage rejection curves.

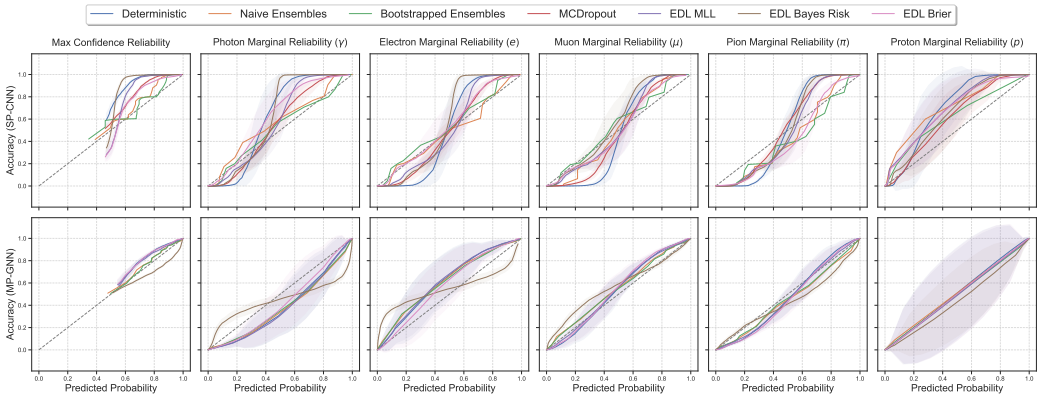

Figure 12: Reliability plots for single (SP-CNN) and multi-particle (MP-GNN) classification.

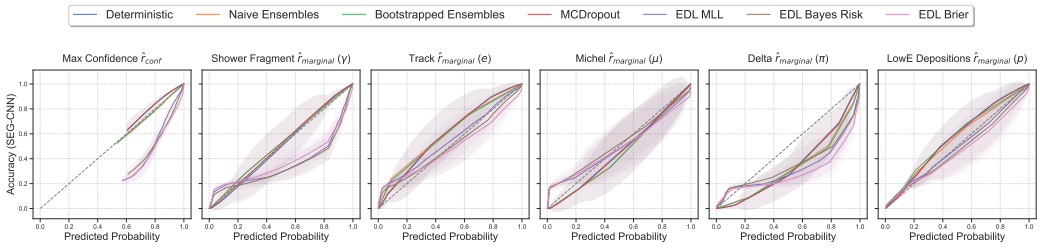

Figure 13: Reliability plots for semantic segmentation.

6 Discussion

We evaluated three different uncertainty quantification methods for deep neural networks on the task of single particle classification, multi-particle classification, and semantic segmentation using high resolution 3D LArTPC energy deposition images. The various metrics evaluating calibration fidelity and discriminative capacity leads to a notable conclusion: simple ensembling of few independently trained neural networks generally achieve highest accuracy and best calibration of output probability values. Also, we observe that the quality of uncertainty quantification depends greatly on the type of the classifier's task, and often it is possible for bayesian models to perform worse than deterministic networks in calibration. While out-of-distribution and mis-classification resilience of uncertainty quantifying neural nets may be used for rejecting unreliable predictions, obtaining calibrated probability estimates would provide further credibility in using deep learning techniques for physical sciences. Future work will focus on model-independent, post-hoc calibration methods such as temperature scaling [16].

References

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2021.05.008. URL `https://www.sciencedirect.com/science/article/pii/S1566253521001081`.

[2] R. Acciarri et al. Long-Baseline Neutrino Facility (LBNF) and Deep Underground Neutrino Experiment (DUNE). 2016.

[3] C. Adams, M. Alrashed, R. An, J. Anthony, J. Asaadi, A. Ashkenazi, M. Auger, S. Balasubramanian, B. Baller, C. Barnes, G. Barr, M. Bass, F. Bay, A. Bhat, K. Bhattacharya, M. Bishai, A. Blake, T. Bolton, L. Camilleri, D. Caratelli, I. Caro Terrazas, R. Carr, R. Castillo Fernandez, F. Cavanna, G. Cerati, Y. Chen, E. Church, D. Cianci, E. O. Cohen, G. H. Collin, J. M. Conrad, M. Convery, L. Cooper-Troendle, J. I. Crespo-Anadón, M. Del Tutto, D. Devitt, A. Diaz, K. Duffy, S. Dytman, B. Eberly, A. Ereditato, L. Escudero Sanchez, J. Esquivel, J. J. Evans, A. A. Fadeeva, R. S. Fitzpatrick, B. T. Fleming, D. Franco, A. P. Furmanski, D. Garcia-Gamez, V. Genty, D. Goeldi, S. Gollapinni, O. Goodwin, E. Gramellini, H. Greenlee, R. Grosso, R. Guenette, P. Guzowski, A. Hackenburg, P. Hamilton, O. Hen, J. Hewes, C. Hill, G. A. Horton-Smith, A. Hourlier, E.-C. Huang, C. James, J. Jan de Vries, X. Ji, L. Jiang, R. A. Johnson, J. Joshi, H. Jostlein, Y.-J. Jwa, G. Karagiorgi, W. Ketchum, B. Kirby, M. Kirby, T. Kobilarcik, I. Kreslo, I. Lepetic, Y. Li, A. Lister, B. R. Littlejohn, S. Lockwitz, D. Lorca, W. C. Louis, M. Luethi, B. Lundberg, X. Luo, A. Marchionni, S. Marcocci, C. Mariani, J. Marshall, J. Martin-Albo, D. A. Martinez Caicedo, A. Mastbaum, V. Meddage, T. Mettler, K. Mistry, A. Mogan, J. Moon, M. Mooney, C. D. Moore, J. Mousseau, M. Murphy, R. Murrells, D. Naples, P. Nienaber, J. Nowak, O. Palamara, V. Pandey, V. Paolone, A. Papadopoulou, V. Papavassiliou, S. F. Pate, Z. Pavlovic, E. Piasetzky, D. Porzio, G. Pulliam, X. Qian, J. L. Raaf, A. Rafique, L. Ren, L. Rochester, M. Ross-Lonergan, C. Rudolf von Rohr, B. Russell, G. Scanavini, D. W. Schmitz, A. Schukraft, W. Seligman, M. H. Shaevitz, R. Sharankova, J. Sinclair, A. Smith, E. L. Snider, M. Soderberg, S. Söldner-Rembold, S. R. Soleti, P. Spentzouris, J. Spitz, J. St. John, T. Strauss, K. Sutton, S. Sword-Fehlberg, A. M. Szelc, N. Tagg, W. Tang, K. Terao, M. Thomson, R. T. Thornton, M. Toups, Y.-T. Tsai, S. Tufanli, T. Usher, W. Van De Pontseele, R. G. Van de Water, B. Viren, M. Weber, H. Wei, D. A. Wickremasinghe, K. Wierman, Z. Williams, S. Wolbers, T. Wongjirad, K. Woodruff, T. Yang, G. Yarbrough, L. E. Yates, G. P. Zeller, J. Zennamo, and C. Zhang. Deep neural network for pixel-level electromagnetic particle identification in the microboone liquid argon time projection chamber. *Phys. Rev. D*, 99:092001, May 2019. doi: 10.1103/PhysRevD.99.092001. URL `https://link.aps.org/doi/10.1103/PhysRevD.99.092001`.

[4] Corey Adams, Kazuhiro Terao, and Taritree Wongjirad. Pilarnet: Public dataset for particle imaging liquid argon detectors in high energy physics, 2020.

[5] Alexander Amini, Wilko Schwarting, A. Soleimany, and D. Rus. Deep evidential regression. *ArXiv*, abs/1910.02600, 2020.

[6] Alexander Amini, Wilko Schwarting, Ava P. Soleimany, and Daniela Rus. Deep evidential regression. *ArXiv*, abs/1910.02600, 2020.

[7] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

[8] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

[9] Laura Dominé and Kazuhiro Terao. Scalable deep convolutional neural networks for sparse, locally dense liquid argon time projection chamber data. *Phys. Rev. D*, 102:012005, Jul 2020. doi: 10.1103/PhysRevD.102.012005. URL `https://link.aps.org/doi/10.1103/PhysRevD.102.012005`.

[10] Laura Dominé and Kazuhiro Terao. Scalable deep convolutional neural networks for sparse, locally dense liquid argon time projection chamber data, 2019.

[11] Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*, pages 569–593. Springer, 1992.

[12] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2005.10.010. URL `https://www.sciencedirect.com/science/article/pii/S016786550500303X`. ROC Analysis in Pattern Recognition.

[13] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1050–1059. JMLR.org, 2016.

[14] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1050–1059. JMLR.org, 2016.

[15] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL `https://doi.org/10.1198/016214506000001437`.

[16] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1321–1330. JMLR.org, 2017.

[17] Holly C. Hartmann, Thomas C. Pagano, S. Sorooshian, and R. Bales. Confidence builders: Evaluating seasonal climate forecasts from user perspectives. *Bulletin of the American Meteorological Society*, 83(5):683 – 698, 2002. doi: 10.1175/1520-0477(2002)083<0683:CBESCF>2.3.CO;2. URL `https://journals.ametsoc.org/view/journals/bams/83/5/1520-0477_2002_083_0683_cbescf_2_3_co_2.xml`.

[18] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[19] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[20] Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks. In *International Conference on Computer Aided Verification*, pages 3–29. Springer, 2017.

[21] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017.

[22] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *ArXiv*, abs/1511.02680, 2017.

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[24] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6405–6416, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[26] Pedro A.N. Machado, Ornella Palamara, and David W. Schmitz. The short-baseline neutrino program at fermilab. *Annual Review of Nuclear and Particle Science*, 69(1):363–387, 2019. doi: 10.1146/annurev-nucl-101917-020949. URL https://doi.org/10.1146/annurev-nucl-101917-020949.

[27] Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 2015:2901–2907, Jan 2015. ISSN 2159-5399. URL https://pubmed.ncbi.nlm.nih.gov/25927013. 25927013[pmid].

[28] Jeremy Nixon, Michael W. Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[29] Aaditya Ramdas, Nicolás García Trillos, and Marco Cuturi. On wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19:47, 2017.

[30] Aaditya Ramdas, Nicolás García Trillos, and Marco Cuturi. On wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2), 2017. ISSN 1099-4300. doi: 10.3390/e19020047. URL https://www.mdpi.com/1099-4300/19/2/47.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[32] Carlo Rubbia. The liquid-argon time projection chamber: a new concept for neutrino detectors. Technical report, CERN, Geneva, 1977. URL https://cds.cern.ch/record/117852.

[33] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/a981f2b708044d6fb4a71a1463242520-Paper.pdf.

[34] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/a981f2b708044d6fb4a71a1463242520-Paper.pdf.

[35] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

[36] Hao Wang and Dit-Yan Yeung. A survey on bayesian deep learning. *ACM Comput. Surv.*, 53 (5), sep 2020. ISSN 0360-0300. doi: 10.1145/3409383. URL https://doi.org/10.1145/3409383.

[37] David Widmann, Fredrik Lindsten, and Dave Zachariah. Calibration tests in multi-class classification: A unifying framework. In *NeurIPS*, 2019.

[38] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 694–699, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 158113567X. doi: 10.1145/775047.775151. URL https://doi.org/10.1145/775047.775151.

[39] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[40] Hao Zheng, Zhanlei Yang, Wenju Liu, Jizhong Liang, and Yanpeng Li. Improving deep neural networks using softplus units. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–4, 2015. doi: 10.1109/IJCNN.2015.7280459.