SAND2020-2658C

We Know Where We Don't Know: 3D Bayesian CNNs for Uncertainty Quantification of Binary Segmentations for Material Simulations

Anonymous ECCV submission

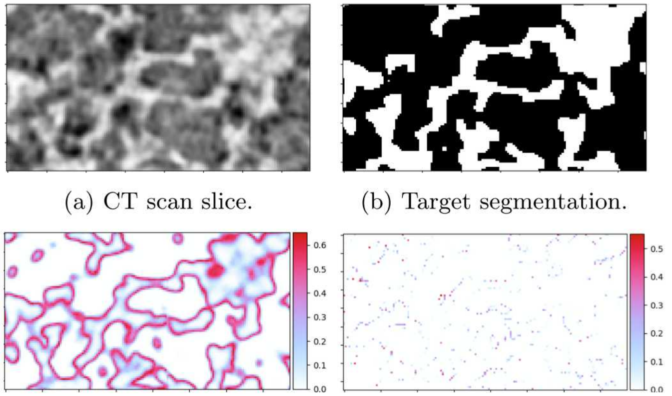
Paper ID 4399

Abstract. Deep learning has been applied with great success to the segmentation of 3D Computed Tomography (CT) scans. Establishing the credibility of these segmentations requires uncertainty quantification (UQ) to identify untrustworthy predictions. Recent UQ architectures include Monte Carlo dropout networks (MCDNs), which approximate Bayesian inference in deep Gaussian processes, and Bayesian neural networks (BNNs), which use variational inference to learn the posterior distribution of the neural network weights. BNNs hold several advantages over MCDNs for UQ, but, to the best of our knowledge, there has not been a successful application of BNNs to 3D domains. We propose a novel 3D Bayesian convolutional neural network (BCNN), the first variational inference-based architecture designed for segmentation and credible UQ in a 3D domain. We present experimental results on CT scans of graphite electrodes and laser-welded metals and show that our BCNN outperforms an MCDN in recent uncertainty metrics. The geometric uncertainty maps generated by our BCNN capture continuity and visual gradients, making them interpretable as confidence intervals in physics simulations.

Keywords: Uncertainty quantification, volumetric segmentation, variational inference

1 Introduction

Non-destructive 3D imaging techniques allow scientists to study the interior of objects which cannot otherwise be observed. For example, radiologists use X-ray Computed Tomography (CT) to measure organ perfusion and Magnetic Resonance Imaging (MRI) to diagnose prostate carcinoma, among other applications [3,22]. In addition to medical applications, CT scans are used in manufacturing to identify defects before a part is deployed in a production environment and to certify physical properties of materials. A critical step in the analysis of CT scans is segmentation, wherein an analyst labels each voxel in a scan (*e.g.,* as a tumor in the medical case or as a defect in the manufacturing case). However, due to the noise and artifacts found in CT scans along with human error, these segmentations are often expensive, irreproducible, and unreliable [17]. Deep learning models such as convolutional neural networks (CNNs) have revolutionized the



(a) CT scan slice. (b) Target segmentation.

(c) BCNN uncertainty (ours). (d) MCDN uncertainty.

Fig. 1: Zoomed Uncertainty Maps on Graphite Test Set Sample III, Slice 64. Note that the BCNN uncertainty map captures continuity and visual gradients while the MCDN uncertainty map is pixelated and uninterpretable.

automated segmentation of 3D imaging by providing a fast, accurate solution to many challenges in segmentation.

For use with high-consequence part certification, segmentation must include uncertainty quantification (UQ). When deploying critical parts, such as those in cars and airplanes, analysts must provide accurate safety confidence intervals. Recent research casts deep neural networks as probabilistic models in order to obtain uncertainty measurements. Two common UQ architectures are Monte Carlo dropout networks (MCDNs) [7] and variational inference-based Bayesian neural networks (BNNs) [1]. MCDNs are easy to implement and enable UQ in the output space with little computational cost, but provide poor geometric uncertainty maps. In contrast, BNNs measure uncertainty in the weight space, resulting in statistically-justified UQ at the cost of at least double the number of trainable parameters and increased convergence time [7].

To the best of our knowledge, there is no existing BNN that successfully generates statistically interpretable uncertainty measurements in 3D domains; recent work has theorized that this is computationally infeasible [7,13]. We refute this and propose a novel 3D Bayesian CNN (BCNN) architecture, the first variational inference-based architecture designed for segmentation and UQ in a 3D domain. Our BCNN effectively predicts binary segmentations of billion-voxel CT scans in addition to generating statistically credible geometric uncertainty maps which the MCDN cannot capture. We show via experimental results on CT scan datasets of graphite electrodes and laser-welded metals that our BCNN outperforms the regularly-adapted MCDN on UQ on recent uncertainty metrics [21]. As shown in Figure 1, the BCNN generates an interpretable uncertainty map that enables uncertainty quantification in material simulations that require precise geometric confidence intervals.

2 Related Work

In this section, we describe recent publications in volumetric segmentation and UQ which enabled the success of our BCNN.

2.1 Volumetric Segmentation

The problem of volumetric segmentation has seen much high-impact work in the past few years. The 2D Fully Convolutional Network [15] and U-Net [27] led Milletari *et al.* [18] to propose the first 3D CNN for binary segmentation of MRI images, called V-Net. At around the same time, Çiçek *et al.* [4] proposed 3D U-Net, a direct extension of the U-Net to a 3D domain. While V-Net was designed for binary segmentation of the human prostate and 3D U-Net was designed for binary segmentation of the kidney of the *Xenopus*, they both employ an encoder-decoder architecture inspired by U-Net [18,4]. In this technique, a 3D volume is mapped to a latent space via successive convolutional and pooling layers; this latent representation is then upsampled and convolved until it reaches the size of the original volume and outputs the resulting per-voxel segmentation [27].

While most volumetric segmentation work pertains to the medical field, 3D materials segmentation is also an active area of research due to the importance of quality segmentations in physics simulations. In 2018, Konopczyński *et al.* [12] employed fully convolutional networks to segment CT scan volumes of short glass fibers, outperforming traditional non-deep learning techniques and achieving the first accurate results in low-resolution fiber segmentation. More recently, MacNeil *et al.* [16] proposed a semi-supervised algorithm for segmentation of woven carbon fiber volumes from sparse input.

2.2 Uncertainty Quantification

While deep learning models often outperform traditional statistical approaches in terms of accuracy and generalizability, they do not have built-in uncertainty measurements like their statistical counterparts. Gal and Ghahramani [7] showed that predictive probabilities (*i.e.,* the softmax outputs of a model) are often erroneously interpreted as an uncertainty metric. Instead, recent work has cast neural networks as Bayesian models via approximating probabilistic models [7] or utilized variational inference to learn the posterior distribution of the network weights [1].

Monte Carlo Dropout Networks (MCDNs) Gal and Ghahramani [7] showed that a neural network with dropout applied before every weight layer (*i.e.,* an MCDN) is mathematically equivalent to an approximation to a deep Gaussian process [5]. Specifically, one can approximate a deep Gaussian process with covariance function $\mathbf{K}(\boldsymbol{x}, \boldsymbol{y})$ by placing a variational distribution over each component of a spectral decomposition of $\mathbf{K}$. This maps each layer of the deep Gaussian process to a layer of hidden units in a neural network. By averaging stochastic forward passes through the dropout network at inference time, one obtains a Monte Carlo approximation of the intractable approximate predictive distribution of the deep Gaussian process [7]; thus the voxel-wise standard deviations of the predictions are usable as an uncertainty metric.

One of the top benefits of the MCDN is its ease of implementation; as an architecture-agnostic technique which is dependent only on the dropout layers, Monte Carlo dropout can easily be added to very large networks without an increase in parameters. As a result, MCDNs have been implemented with good results in several different applications. In particular, Liu *et al.* [14] successfully implemented a 3D MCDN for UQ in binary segmentations of MRI scans of the amygdala, and Martinez *et al.* [17] used V-Net with Monte Carlo dropout for UQ in binary segmentations of CT scans of woven composite materials.

While the MCDN is one of the most common UQ architectures used in deep learning, its statistical soundness has been called into question. Osband [24] argues that Monte Carlo dropout provides an approximation to the risk of a model rather than its uncertainty (in other words, that it approximates the inherent stochasticity of the model rather than the variability of the model's posterior belief). Osband [24] also shows that the posterior distribution given by dropout does not necessarily converge as more data is gathered; instead, the posterior depends only on the interaction between the dropout rate and the model size.

Bayesian Neural Networks (BNNs) Another approach to UQ in deep neural networks is Bayesian learning via variational inference (*i.e.,* a BNN). Instead of point estimates, the network learns the posterior distribution over the weights given the dataset, denoted $P(\boldsymbol{w}|\mathcal{D})$, given the prior distribution $P(\boldsymbol{w})$. However, calculating the exact posterior distribution is intractable due to the extreme overparametrization found in neural networks [1]. Previous work by Hinton and Van Camp [9] and Graves [8] proposed variational learning as a method to approximate the posterior distribution. Variational learning finds the parameters θ of the distribution $q(\boldsymbol{w}|\theta)$ via the minimization of the variational free energy cost function, often called the expected lower bound (ELBO). It consists of the sum of the Kullback-Leibler (KL) divergence and the negative log-likelihood (NLL), which Blundell *et al.* [1] explains as embodying a tradeoff between satisfying the simplicity prior (represented by the KL term) and satisfying the complexity of the dataset (represented by the NLL term):

$$\mathcal{F}(\mathcal{D},\theta) = \mathrm{KL}[q(\boldsymbol{w}|\theta) \parallel P(\boldsymbol{w})] - \mathbb{E}_{q(\boldsymbol{w}|\theta)}[\log P(\mathcal{D}|\boldsymbol{w})]. \tag{1}$$

Blundell *et al.* [1] proposed the Bayes by Backprop algorithm, which combines variational inference with traditional backpropagation to find the best approximation to the posterior in a computationally feasible manner. Bayes by Backprop works by using the gradients calculated in backpropagation to "scale and shift" the variational parameters of the posterior, thus updating the posterior with minimal additional computation [1].

One challenge associated with probabilistic weights is that all examples in a mini-batch typically have similarly sampled weights, limiting the variance reduction effect of large mini-batches [31]. Kingma *et al.* [11] introduced local reparametrization, which greatly reduces the variance of stochastically sampled weights by transforming global weight uncertainty into independent local noise across examples in the mini-batch. In a similar vein, Wen *et al.* [31] proposed the Flipout estimator, which empirically achieves ideal variance reduction by sampling weights pseudo-independently for each example. While local reparametrization only works for fully-connected networks, Flipout can be used effectively in fully-connected, convolutional, and recurrent networks [31].

2.3 Novelty and Advantages of our BCNN

While we have leveraged many ideas from previous work, to the best of our knowledge there is no existing Bayesian CNN that successfully generates statistically interpretable geometric uncertainty maps, in either 2D or 3D. In the 2D domain, Shridhar *et al.* [28] proposed a 2D BCNN that extended local reparametrization to convolutional networks but did not generate geometric uncertainty maps. Furthermore, Ovadia *et al.* [25] showed that 2D BCNNs with Flipout [31] are effective for non-geometric UQ on the MNIST and CIFAR-10 datasets, but they found it was difficult to get BCNNs to work with complex datasets. As such, our work was not a straightforward extension from 2D to 3D, but instead a discovery of a unique synthesis of techniques that enabled successful training and segmentation of large 3D volumes with credible uncertainty quantification.

The major advantage of BCNNs is that they measure uncertainty in the *weight space*, while the MCDNs measure uncertainty in the *output space*. We acknowledge that MCDNs can provide uncertainty maps. However, due to being measured in the output space, these uncertainty maps are in the form of statistics drawn from many runs and are not statistically justified [24]. Given that we are working with 3D volumes of up to a billion voxels, the cost of running inference enough times to characterize the true distribution of the softmax output for each voxel is prohibitive. To obtain credible UQ, we must study the true distribution of sigmoid values – that is, the distribution in the weight space. Because the BCNN measures this distribution, it provides meaningful uncertainty maps that can be directly interpreted: we can easily provide statistically justified 90% confidence intervals on the BCNN prediction by taking the difference of the 0.05 and 0.95 percentiles of the learned distributions.

Finally, as seen in Figure 1 of our paper, the BCNN uncertainty maps capture continuity and visual gradients, which is a major advantage not only for material simulations as discussed in Section 5.4, but for any application where geometric uncertainty must be quantified and understood. The major disadvantages of BCNNs compared to MCDNs are implementation-based, including doubling of trainable parameters [7], lengthy training times, and sensitivity to hyperparameter optimization [25].

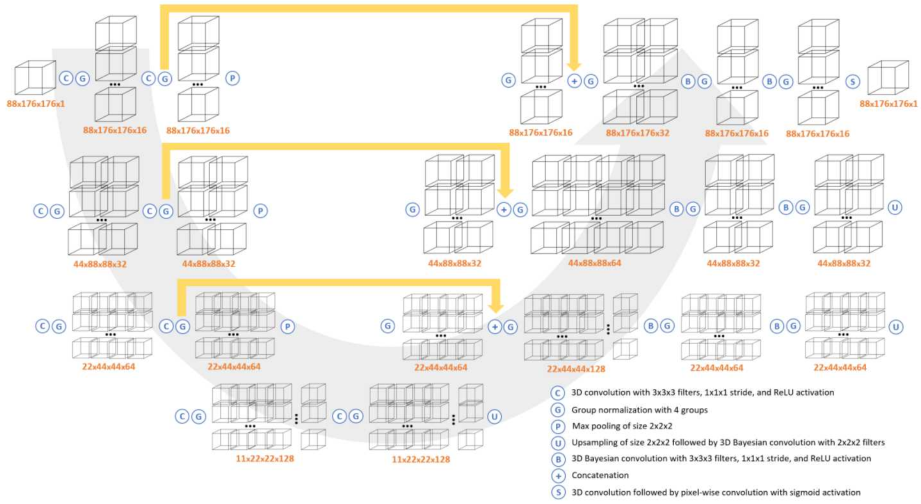


Fig. 2: Schematic of our BCNN architecture with sample volume dimensions from the Graphite dataset. Best viewed in electronic format. Measurements are (depth, height, width, channels).

3 Methodology

In this section, we present our BCNN architecture and describe our reasoning behind several design decisions.

3.1 Architecture

In Figure 2, we present a schematic representation of our BCNN architecture. Similarly to V-Net [18], we employ an encoder-decoder architecture. The encoder half (left) of the network compresses the input into a latent space while the decoder half (right) decompresses the latent representation of the input into a segmentation map. We do not include stochastic layers in the encoder half of the network to maximize the amount of information transfer between the original volume and the latent space.

The encoder half of the network is comprised of four stages, each with two convolutional layers and normalization layers followed by a max pooling layer to reduce the size of the input. Thus, after each layer, the volume's depth, height, and width are halved while its channels are doubled, reducing the size of the volume by a factor of four.

The decoder half of the network consists of three stages, corresponding to the first three layers of the encoder half. First, we upsample the output of the previous layer and apply convolutional and normalization layers to double the volume's depth, height, and width while halving its channels. We then concatenate this volume with the pre-pooling output of the corresponding encoder layer;

this skip connection assists in feature-forwarding through the network. Then, we apply two more convolutional and normalization layers. At the end of the third stage, we apply a final convolutional layer as well as a sigmoid activation. This results in a volume of the same size as the input representing a binary segmentation probability map.

In the decoder half of the network, we implement volumetric convolutional layers with distributions over the weights. Each Bayesian convolutional layer is initialized with a standard normal prior $P(\boldsymbol{w}) = \mathcal{N}(0, 1)$ and employs the aforementioned Flipout estimator [31] to approximate the distribution during forward passes. Our implementation draws from the Bayesian Layers library [30] included in TensorFlow Probability [6], which monitors the KL divergence of the layer's posterior distribution with respect to its prior. Our BCNN has 1,924,964 trainable parameters, while its MCDN counterpart has 1,403,059.

3.2 Design Decisions

Since training volumes can be quite large, our batch size is constrained by the amount of available GPU memory, resulting in a batch size too small for batch normalization to accurately compute batch statistics. Thus, we implement a recent technique proposed by Wu and He [32] called group normalization, which normalizes groups of channels and is shown to have accurate performance independent of batch size. Proper normalization was observed to be a critical factor in the convergence of our model; by tuning the number of groups used in the group normalization layers, we found that our model converged most reliably when using 4 groups.

At each downward layer i, we apply 2^{3+i} filters. This was found to be more effective than a more simple model with 2^{2+i} filters and a more complex model with 2^{4+i} filters. We hypothesize that some minimum amount of learned parameters was necessary to produce accurate segmentations, but with 2^{4+i} filters, the model's overparameterization made training significantly more difficult.

We tested many prior distributions, including scale mixture [1], spike-and-slab [19], and a normal distribution with increased variance, but found that a standard normal prior provided the best balance between weight initialization and weight exploration. Skip connections were found to slightly increase the accuracy of our predictions by forwarding fine-grained features that otherwise would have been lost in the encoder half of the network. We experimented with both max pooling and downward convolutional layers and observed negligible difference.

4 Experiments

In this section, we describe our datasets and detail our training and testing procedures.

4.1 Datasets

Two 3D imaging datasets are used to test our BCNN. The first is a series of CT scans of graphite electrodes for lithium-ion batteries, which we refer to as the Graphite dataset [20,26]. This material consists of non-spherical particles (dark objects in the images) that are coated onto a substrate and calendared to densify. The academically manufactured ("numbered") electrodes [20] were imaged with 325 nm resolution and a domain size of $700 \times 700 \times (48 - 75)$ µm. The commercial ("named") electrodes [26] were imaged at 162.5 nm resolution and a domain size of $416 \times 195 \times 195$ µm. Eight samples were studied, each with 500 million to 1 billion voxels. Each volume was hand-segmented using commercial tools [23]; these manual segmentations were used for training and testing. We trained our BCNN on the GCA400 volume and tested on the remaining seven electrodes.

Laser-welded metal joints comprise a second dataset, which we refer to as the Laser Weld dataset. To generate these volumes, two metal pieces are put into contact and joined with an incident laser beam. The light regions of the resulting scans represent voids or defects in the weld. The Laser Weld dataset consists of CT scans of ten laser-welded metal joint examples, each with tens of millions of voxels. Similarly to the Graphite dataset, these volumes were manually segmented and used for training and testing. We trained a separate BCNN on samples S2, S24, and S25, then tested on the remaining seven held-out volumes.

For both datasets, we normalized each CT scan to have voxel values with zero mean and unit variance. Additionally, each CT scan was large enough to require that we process subvolumes of the 3D image rather than ingesting the entire scan as a whole into the neural network on the GPU. Our algorithm for preprocessing these volumes is set forth in the Appendix.

4.2 Training

We use the Adam optimizer [10] with learning rate $\alpha = 0.0001$ for the Graphite dataset and $\alpha = 0.001$ for the Laser Weld dataset; this difference is necessary because the volumes in the Graphite dataset are significantly larger than those of the Laser Weld dataset.

We utilize Graves' [8] amendment of variational free energy (originally Equation 1) to mini-batch optimization for mini-batch $i \in \{1, 2, \ldots, M\}$ by dividing the KL term by M. This factor distributes the KL divergence penalty evenly over each minibatch; without this scaling, the KL divergence term dominates the equation, causing the model to converge to a posterior with suboptimal accuracy.

We also use monotonic KL annealing [2] as detailed in Equation 2; this annealing was necessary for the reliable convergence of our model as it allowed the model to learn the 3D segmentation before applying the KL divergence penalty. We denote the current epoch as E and accept as hyperparameters a KL starting epoch s, initial KL weight k_0, and step value k_1 to obtain the KL weight for the current epoch k_E as follows:

$$k_E = \begin{cases} k_0 & \text{if } E \leq s \\ \min(1, k_0 + k_1(E - s)) & \text{if } E > s \end{cases} \tag{2}$$

For the Graphite dataset we use $s = 1, k_0 = 1/2, k_1 = 1/2$ and for the Laser Weld dataset we use $s = 1, k_0 = 0, k_1 = 1/4$. We use the aforementioned Bayes by Backprop [1] algorithm to train our BCNN under the resultant loss function:

$$\mathcal{F}_i^E(\mathcal{D}_i, \theta) = \frac{k_E}{M}\mathrm{KL}[q(\boldsymbol{w}|\theta) \parallel P(\boldsymbol{w})] - \mathbb{E}_{q(\boldsymbol{w}|\theta)}[\log P(\mathcal{D}_i|\boldsymbol{w})]. \tag{3}$$

We parallelized our model and trained on two NVIDIA Tesla V100 GPUs with 32GB of memory each. For our BCNN, one epoch of 1331 chunks of size $88\times176\times176$ took approximately 17 minutes and 30 seconds with a maximum batch size of 3. We trained each model for 2 epochs on the 4913-sample Graphite dataset; for the 549-sample Laser Weld dataset, we trained each model for 7 epochs.

4.3 Testing

We computed 48 Monte Carlo samples on each test chunk to obtain a distribution of sigmoid values for each voxel. The Monte Carlo dropout technique is justified in representing uncertainty as the standard deviation of the sigmoid values because it approximates a deep Gaussian process [7]; however, the BCNN does not guarantee adherence to a normal distribution in practice. Thus, in order to effectively compare the outputs of both networks while mimicking the standard deviation measurement of the MCDN, we represent confidence intervals on the segmentation as the 33^{rd} and the 67^{th} percentiles of the sigmoid values, and uncertainty as the difference. We compare our results against an MCDN of identical architecture to our BCNN except with regular convolutional layers instead of Bayesian convolutional layers and spatial dropout [29] applied at the end of each stage prior to upsampling.

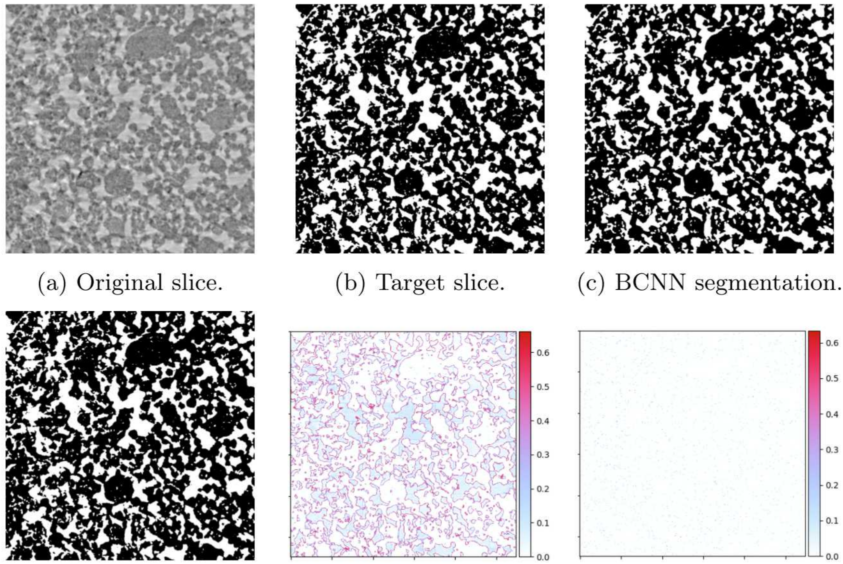
5 Results

In this section, we present inference results of our BCNN and compare its performance with the MCDN.

5.1 Graphite Dataset

Figure 3 shows a successful segmentation and uncertainty measurements on the GCA2000 sample from the Graphite dataset. Our BCNN provides an equivalent or better segmentation than the MCDN and produces an interpretable geometric uncertainty map. Figure 1 shows a zoomed-in portion of the III sample uncertainty map which highlights the continuity and visual gradients captured in our BCNN uncertainty map, while the MCDN produces uninterpretable voxel-by-voxel uncertainty measurements. We hypothesize that this is an advantage of our BCNN measuring the uncertainty in the weight space, rather than in the output space like the MCDN.

Table 1 lists a selection of descriptive statistics regarding model performance on the Graphite dataset. Our BCNN achieves a higher segmentation accuracy



(a) Original slice. (b) Target slice. (c) BCNN segmentation.

(d) MCDN segmentation. (e) BCNN uncertainty (ours). (f) MCDN uncertainty.

Fig. 3: Results on Graphite Test Set Sample GCA2000, Slice 212. Note that our BCNN uncertainty is focused around the light gray edges of the material in the original slice, while the MCDN uncertainty is pixelated and uninterpretable.

Sample	Method	Accuracy	UQ Mean ($\times 10^{-2}$)
I	MCDN	0.8295	0.7566
	BCNN (ours)	0.8452	7.991
III	MCDN	0.7410	
	BCNN (ours)	0.7560	
IV	MCDN	0.6925	0.7696
	BCNN (ours)	0.7226	7.871
GCA2000	MCDN		
	BCNN (ours)		
25R6	MCDN		
	BCNN (ours)		
E35	MCDN		
	BCNN (ours)		
Litarion	MCDN		
	BCNN (ours)		

Table 1: Graphite Test Set Statistics. Note that our BCNN has roughly the same accuracy performance as the MCDN. Additionally, our BCNN has an order of magnitude more uncertainty due to its increased stochasticity.

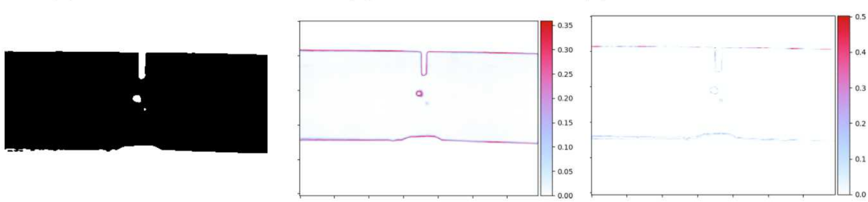
than the MCDN on the numbered datasets but slightly lower accuracy on the named datasets. The manual labels resulted from thresholding techniques and are known to contain inaccuracies, especially at particle boundaries. As such, we conclude that the accuracy performance of our BCNN is similar to that of the MCDN with respect to these labels, but further assessments against refined labels are left for future work.

5.2 Laser Weld Dataset

Figure 4 shows a successful segmentation and uncertainty measurements on the S33 sample from the Laser Weld dataset. Note that the BCNN uncertainty map captures the uncertainty gradient (corresponding to the gray portion of the CT scan slice) at the top left and bottom left of the segmentation, while the MCDN uncertainty map displays a straight line. Figure 5 shows another successful segmentation and uncertainty measurements on the S4 sample from the Laser Weld dataset.

Table 2 lists a selection of descriptive statistics regarding model performance on the Laser Weld dataset. Note that it is slightly more difficult for our BCNN to produce accurate segmentations on the Laser Weld dataset than the Graphite dataset.

(a) Original slice. (b) Target slice. (c) BCNN segmentation (ours).



(d) MCDN segmentation. (e) BCNN uncertainty (ours). (f) MCDN uncertainty.

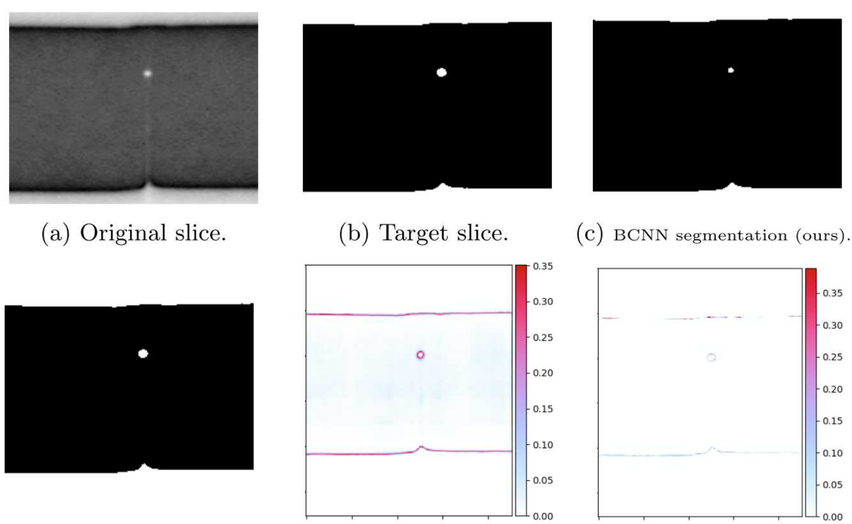
Fig. 4: Results on Laser Weld Test Set Sample S33, Slice 604. Notice that our BCNN achieves a more accurate segmentation in addition to producing an uncertainty map with consistent uncertainty measurements across the borders of the weld. Additionally, our BCNN learned that there is a distribution of uncertainty around the central void, whereas the MCDN represents it with a voxel-wide line.

5.3 Validation

Validation of UQ results is a difficult subject with no standard practice for determining whether a model's UQ is justified given the dataset. In validating our

Sample	Method	Accuracy	UQ Mean ($\times 10^{-3}$)
S1	MCDN	0.9949	0.8704
	BCNN (ours)	0.9943	6.560
S4	MCDN	0.9948	
	BCNN (ours)	0.9926	
S15	MCDN	0.9984	1.115
	BCNN (ours)	0.9921	12.74
S26	MCDN	0.9861	0.8969
	BCNN (ours)	0.9931	9.035
S31	MCDN	0.9972	
	BCNN (ours)	0.9889	
S32	MCDN	0.9914	
	BCNN (ours)	0.9885	
S33	MCDN	0.9941	1.619
	BCNN (ours)	0.9882	7.283

Table 2: Laser Weld Test Set Statistics. Similarly to the Graphite dataset, our BCNN has roughly the same accuracy performance as the MCDN with an order of magnitude more uncertainty due to its increased stochasticity.



(a) Original slice. (b) Target slice. (c) BCNN segmentation (ours).

(d) MCDN segmentation. (e) BCNN uncertainty (ours). (f) MCDN uncertainty.

Fig. 5: Results on Laser Weld Test Set Sample S4, Slice 372. While the BCNN segmentation underestimates the size of the void, it expresses a thick uncertainty band reflecting its correct size. Note also that the BCNN uncertainty better captures the continuity of the edges of the weld.

BCNN, the most relevant work in this area is due to Mukhoti and Gal [21]. They define two desiderata for quality uncertainty maps: a high probability of being accurate when the model is certain, denoted $P(A|C)$, and a high probability of being uncertain when the model is inaccurate, denoted $P(U|I)$.

They estimate these quantities by evaluating accuracy and uncertainty by sliding a square patch across the image; if the patch accuracy is equal to or above a certain threshold, the entire patch is labeled accurate, and if the patch uncertainty is equal to or above a certain threshold, the entire patch is labeled uncertain. They define a metric called Patch Accuracy vs. Patch Uncertainty (PAvPU), which encodes the above two desiderata in addition to penalizing patches which are simultaneously accurate and uncertain [21]. If n represents the total number of patches, n_{ac} represents the number of patches which are accurate and certain, and n_{iu} represents the number of patches which are inaccurate and uncertain, PAvPU is defined as follows:

$$\text{PAvPU} = \frac{n_{ac} + n_{iu}}{n} \tag{4}$$

We implement PAvPU to validate our uncertainty results using a 3×3 patch with accuracy threshold 8/9 and uncertainty threshold equal to the mean of the uncertainty map. We detail our results in Table 3. In particular, note that our BCNN consistently outperforms the MCDN in both conditional probabilities, even doubling the $P(U|I)$ score. Thus, we conclude that our BCNN has more justified UQ than the MCDN, and it is more effective than the MCDN in encoding the relationship between uncertainty and accuracy.

As PAvPU was designed for use with 2D semantic segmentations and not for 3D binary segmentations, it may not be sufficient to characterize the improvement in UQ achieved by the BCNN. Furthermore, the PAvPU calculation involves a penalty for patches which are accurate and uncertain, which may not necessarily be a detrimental characteristic of the segmentation [21]. This is the term that most significantly affects the PAvPU values where MCDN achieves a better result than our BCNN: our BCNN simply measures *more* uncertainty than the MCDN. Additionally, introducing this penalty term encodes the goal of training a network which is not simultaneously uncertain and accurate; however, in the Bayesian view, uncertainty and accuracy are not mutually exclusive because uncertainty quantifies the proximity of a sample to the training distribution rather than confidence in a correct segmentation. We leave the development of a more relevant uncertainty metric as future work.

5.4 Advantages for Material Simulations

The objective of performing UQ on materials datasets is to obtain uncertainties which can inform and propagate throughout simulations involving said materials. For example, when simulating the performance of a sample from the Graphite dataset to bound its various physical properties, it is crucial to know the contact points of the material. The uncertainty maps generated by our BCNN represent

Sample	Method	$P(A\|C)$	$P(U\|I)$	PAvPU
Litarion, Slice 324	MCDN			
(Graphite)	BCNN			
GCA2000, Slice 212	MCDN			
(Graphite)	BCNN			
III, Slice 64	MCDN			
(Graphite)	BCNN			
S1, Slice 176	MCDN			
(Laser Weld)	BCNN			
S26, Slice 596	MCDN			
(Laser Weld)	BCNN			

Table 3: PAvPU Validation Results. Note that our BCNN consistently and vastly outperforms the MCDN in the $P(A|C)$ and $P(U|I)$ scores, implying that our BCNN better encodes the relationship between uncertainty and accuracy. However, our BCNN underperforms in the PAvPU metric because it is penalized for being simultaneously accurate and uncertain.

confidence intervals on the segmentation, so we can infer the probability of a certain contact point occurring in the CT scanned material.

The voxel-by-voxel nature of the uncertainty maps given by the MCDN produce very jagged, unrealistic confidence intervals with little physical meaning. In contrast, the continuity and visual gradients of the uncertainty map generated by our BCNN enable better approximations to the actual geometric uncertainty in both the Graphite and Laser Weld materials. Our BCNN allows us to smoothly probe the uncertainty when performing simulations and justify each error bound we obtain with interpretable uncertainty maps, a major advantage when performing simulations for high-consequence scenarios.

6 Conclusion

We propose a novel BCNN for UQ of binary segmentations in 3D domains, the first variational-inference based architecture to do so. By measuring uncertainty in the weight space, our BCNN provides interpretable uncertainty maps and outperforms the state-of-the-art Monte Carlo dropout technique. We present UQ results on CT scans of graphite electrodes and laser-welded metals used in physics simulations where accurate geometric UQ is critical. Our BCNN produces uncertainty maps which capture continuity and visual gradients, outperforms Monte Carlo dropout networks (MCDNs) on recent uncertainty metrics, and achieves equal or better segmentation accuracy than MCDNs in most cases. Future investigation will likely include extending our BCNN to semantic segmentation and medical applications and comparing our results with other UQ techniques such as the deep ensembles of Lakshminarayanan *et al.* [13].

References

1. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural networks. In: Proceedings of the 32nd International Conference on Machine Learning (2015) 2, 3, 4, 7, 9
2. Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A.M., Józefowicz, R., Bengio, S.: Generating sentences from a continuous space. In: Proceedings of the SIGNLL Conference on Computational Natural Language Learning (2016), arXiv preprint 1511.06349 8
3. Buzug, T.M.: Computed tomography. In: Kramme, R., Hoffman, K.P., Pozos, R.S. (eds.) Springer Handbook of Medical Technology, chap. 16 (2010) 1
4. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d U-net: Learning dense volumetric segmentation from sparse annotation. In: Proceedings of the 19th International Conference on Medical Image Computing and Computer-Assisted Intervention (2016) 3
5. Damianou, A.C., Lawrence, N.D.: Deep gaussian processes. In: Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (2013) 3
6. Dillon, J.V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., Saurous, R.A.: Tensorflow distributions (2017), arXiv preprint 1711.10604 7
7. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: Proceedings of the 33rd International Conference on Machine Learning (2016) 2, 3, 4, 5, 9
8. Graves, A.: Practical variational inference for neural networks. In: Proceedings of the 24th Conference on Advances in Neural Information Processing Systems (2011) 4, 8
9. Hinton, G.E., Camp, D.V.: Keeping neural networks simple by minimizing the description length of the weights. In: Proceedings of the 16th Conference on Learning Theory (1993) 4
10. Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (2015) 8
11. Kingma, D.P., Salimans, T., Welling, M.: Variational dropout and the local reparameterization trick. In: Proceedings of the 28th Conference on Advances in Neural Information Processing Systems (2015) 5
12. Konopczyński, T., Rathore, D., Rathore, J., Kröger, T., Zheng, L., Garbe, C.S., Carmignato, S., Hesser, J.: Fully convolutional deep network architectures for automatic short glass fiber semantic segmentation from ct scans. In: Proceedings of the 8th Conference on Industrial Computed Tomography (2018) 3
13. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Proceedings of the 30th Conference on Advances in Neural Information Processing Systems (2017) 2, 14
14. Liu, Y., Zhao, G., Nacewicz, B.M., Adluru, N., Kirk, G.R., Ferrazzano, P.A., Styner, M., Alexander, A.L.: Accurate automatic segmentation of amygdala subnuclei and modeling of uncertainty via bayesian fully convolutional neural network (2019), arXiv preprint 1902.07289 4
15. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition (2015) 3

16. MacNeil, J.M.L., Ushizima, D.M., Panerai, F., Mansour, N.N., Barnard, H.S., Parkinson, D.Y.: Interactive volumetric segmentation for textile micro-tomography data using wavelets and nonlocal means. Statistical Analysis and Data Mining: The ASA Data Science Journal **12**(4), 338–353 (jun 2019) 3
17. Martinez, C., Potter, K.M., Smith, M.D., Donahue, E.A., Collins, L., Korbin, J.P., Roberts, S.A.: Segmentation certainty through uncertainty: Uncertainty-refined binary volumetric segmentation under multifactor domain shift. In: Proceedings of the 32nd IEEE Conference on Computer Vision and Pattern Recognition: Women in Computer Vision Workshop (2019) 1, 4
18. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: Proceedings of the 4th International Conference on 3D Vision (2016) 3, 6
19. Mitchell, T., Beauchamp, J.: Bayesian variable selection in linear regression. Journal of the American Statistical Association **83**(404), 1023–1032 (1988) 7
20. Müller, S., Pietsch, P., Brandt, B.E., Baade, P., De Andrade, V., De Carlo, F., Wood, V.: Quantification and modeling of mechanical degradation in lithium-ion batteries based on nanoscale imaging. Nature Communications **9**(1), 2340 (Jun 2018) 8
21. Mukhoti, J., Gal, Y.: Evaluating bayesian deep learning methods for semantic segmentation (2019), arXiv preprint 1811.12709 2, 13
22. Nitz, W.: Magnetic resonance imaging. In: Kramme, R., Hoffman, K.P., Pozos, R.S. (eds.) Springer Handbook of Medical Technology, chap. 23 (2010) 1
23. Norris, C., Mistry, A., Mukherjee, P.P., Roberts, S.A.: Microstructural screening for variability in graphite electrodes. *In preparation* 8
24. Osband, I.: Risk versus uncertainty in deep learning: Bayes, bootstrap and the dangers of dropout. In: Proceedings of the 29th Conference on Advances in Neural Information Processing Systems: Workshop on Bayesian Deep Learning (2016) 4, 5
25. Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J.V., Lakshminarayanan, B., Snoek, J.: Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In: Proceedings of the 33rd Conference on Neural Information Processing Systems (2019) 5
26. Pietsch, P., Ebner, M., Marone, F., Stampanoni, M., Wood, V.: Determining the uncertainty in microstructural parameters extracted from tomographic data. Sustainable Energy Fuels **2**, 598–605 (2018) 8
27. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (2015) 3
28. Shridhar, K., Laumann, F., Liwicki, M.: Uncertainty estimations by softplus normalization in bayesian convolutional neural networks with variational inference (2019), arXiv preprint 1806.05978 5
29. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional neural networks. In: Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition (2015) 9
30. Tran, D., Dusenberry, M.W., van der Wilk, M., Hafner, D.: Bayesian layers: A module for neural network uncertainty (2019), arXiv preprint 1812.03973 7
31. Wen, Y., Vicol, P., Ba, J., Train, D., Grosse, R.: Flipout: Efficient pseudo-independent weight perturbations on mini-batches. In: Proceedings of the 6th International Conference on Learning Representations (2018) 5, 7
32. Wu, Y., He, K.: Group normalization. In: Proceedings of the 2018 European Conference on Computer Vision (2018) 7