Uncertainty Quantification in Inverse Scattering Problems with Bayesian Convolutional Neural Networks

Zhun Wei and Xudong Chen

Abstract—Recently, tremendous progress has been made in applying deep learning schemes (DLSs) to solve inverse scattering problems (ISPs), where state-of-the-art performance has been achieved. However, little attention has been paid to the uncertainty quantification of these deep-learning methods in solving ISPs. In other words, the error of the prediction is not known since the ground truth is not available in practice. In this paper, a Bayesian convolutional neural network (BCNN) is used to quantify the uncertainties in solving ISPs. With Monte Carlo dropout, the proposed BCNN is able to directly predict the pixel-based uncertainties of the widely used DLSs in ISPs. To quantitatively evaluate the performance of uncertainty predictions, both correlation coefficient and nonlinear correlation distribution between the predicted uncertainty and true absolute error are calculated. The tests on both synthetic and experimental data show that the predictive uncertainty is highly correlated with true absolute error calculated from the ground truth. Besides, when DLSs fail to solve an ISP, the predicted uncertainty increases significantly, which offers a pixel-based “confidence level” in solving ISPs.

Index Terms—Inverse scattering problems, Uncertainty quantification, Bayesian convolutional neural network, Deep learning schemes.

I. INTRODUCTION

ELECTROMAGNETIC inverse scattering problems (ISPs) are targeted to reconstruct the information of unknown scatterer, such as its position, shape, and electrical properties by probing the unknown target with electromagnetic energy and collecting the response of the target to electromagnetic excitation [1]. Solving ISPs is able to obtain the information of unknown object even when there is a far distance of the targets from sensors or barriers between targets and sensors. Consequently, it is widely seen in through-wall imaging [2], [3], geophysics [4], nondestructive evaluation [5], remote sensing [6] and so on. Nevertheless, due to the ill-posed and nonlinear properties of ISPs, pixel-based reconstruction in ISPs is challenging. To solve ISPs, several non-iterative inversion methods have been proposed, such as Rytov approximation inversion method [7], back-propagation method [8], and extended Born approximation method [9]. One advantage of the non-iterative methods is that they are able to solve ISPs fast, but it suffers from low image qualities of reconstructions, especially for strong scatterers. Besides non-iterative method, some iterative inversion methods have also been proposed, such as distorted Born iterative method [10], contrast source inversion (CSI) method [11], contrast source extended Born method [12], and subspace optimization method (SOM) [13]–[15]. In addition to these objective-function approaches, learning approaches have also been proposed to solve ISPs. For example, artificial neural networks (ANN) [16], [17] and support vector machines [18] have been used in ISPs, where only few parameters are extracted from the scattered electromagnetic field.

Due to the rapid development of artificial intelligence, several DLS have been proposed in ISPs recently. Generally speaking, DLS can be classified into the following categories. The first category is direct learning approach, where the measured scattered field and unknown parameters (permittivity contrast) are directly treated as input and output of DNN, respectively. The “black box” strategy in direct learning approach enables the trained DNN to reconstruct only some simple information [19], such the basic location and shape of the scatterers. The second category is learning-assisted objective-function approach, where ISPs are solved in the framework of traditional object function approach but DNN is employed to learn some components of iterative solvers [20]–[23]. The third category is physics-assisted learning approach, where domain knowledge is incorporated either in the inputs or internal structures of DNN [19], [24]–[28]. It is also shown in [19] by comparisons that physics-assisted learning approach has advantages over direct learning approach, where the former has also been applied to biomedical imaging [29], [30]. Although DNN has been widely used in the ISPs [31], the reconstructed results are highly dependent on training data or network structures. All of the above methods have assumed that the reconstructed results from the network are trustable, but it is not always the case. In this paper, Bayesian convolutional neural network (BCNN) is used to quantify the uncertainties in solving ISPs, where pixel-based uncertainties are obtained through replacing the deterministic weights in traditional DLS structures by distributions over these weights. The main contributions of this paper are summarized as follows:

- The proposed BCNN is able to directly output pixel-based uncertainties of nonlinear reconstructions without any additional calculations, which offers us a “confidence level” of the solutions when solving ISPs.
- Different with previous work [32] that solves ISPs un-
under sparse and weak scatterers conditions, the proposed method in our work do not have such constraints on scatterers.

- Different with the theoretical works in computer vision [33], [34] and application work in phase imaging [35], the proposed framework is designed for ISPs and verified under DLS-based ISP solvers including back-prorogation scheme (BPS) and dominant current scheme (DCS) [19] with both synthetic and experimental data.

- The performance of the predicted uncertainties are quantitatively evaluated with both correlation coefficient and nonlinear correlation distribution. It is also shown that when the DLSs fail in reconstructions, the predicted uncertainty increases significantly.

- The previous work [32] is a model-based method under the framework of first-order Born approximation. On contrary, our method is devoted to qualifying uncertainty when using DLSs to solve ISPs. Thus, the outlined BCNN can be applied to other DLS-based ISP solvers, such as the ones presented in [24]–[28], [36], and to quantify the uncertainties of their reconstructed results.

Notations throughout the paper are as follows: $\tilde{A}$ and $\overline{A}$ are used to denote the matrix and vector forms of the discretized parameter $A$, respectively. We use superscripts $H$, $T$, and $*$ to denote conjugate transpose, transpose, and complex conjugate, respectively. In addition, $|\cdot|$ and $||\cdot||_F$ are used to denote taking element-wise absolute value and Frobenius norm of a matrix.

## II. METHOD

As shown in Fig. 1, a two-dimensional transverse-magnetic case is considered in this paper. A domain of interest (DOI) $D$ is illuminated by $N_t$ line sources located at $r^s_p$ with $p = 1, 2, ..., N_t$, where the electromagnetic waves interact with scatterers in DOI. The scattered field is then collected by $N_r$ antennas at $r^s_q$ with $q = 1, 2, ..., N_r$ on the surface $S$.

### A. Formulation of the problem

In ISPs, the forward model can be formulated by the following two discretized equation based on Lippmann-Schwinger equation:

$$\bar{E}' = \bar{E}' + \bar{G}_D \cdot \bar{\xi} \cdot \bar{E}' , \quad \bar{E}' = \bar{G}_S \cdot \bar{J}, \quad (1)$$

where $\bar{E}'$, $\bar{E}'$, and $\bar{E}'$ denote the discretized forms of total electrical field, incident field, and scattered field, respectively. $\bar{G}_D$ and $\bar{G}_S$ are the free space Green’s functions at domain $D$ and boundary surface $S$, respectively. $\bar{\xi}$ is the contrast in domain $D$, and the contrast current $\bar{J}$ is defined as $\bar{J} = \bar{\xi} \cdot \bar{E}'$ [1], [36]. When solving ISPs with the above equations, the contrast $\bar{\xi}$ is the unknown parameter to be reconstructed based on the measured $\bar{E}'$, whereas $\bar{E}'$, $\bar{G}_D$, and $\bar{G}_S$ are known parameters that are independent of the contrast $\bar{\xi}$.

In typical learning approaches, such as BPS and DCS [19], a training dataset containing $M_t$ pairs of the ground-truth contrasts and their corresponding measured scattered fields is usually generated and used for the training purpose. Specifically, the learning algorithm is expressed as:

$$R^m_t = \arg \min_{R_W} \sum_{m=1}^{M_t} f(R_W(\tilde{\xi}_m), \tilde{\xi}_m) + g(W). \quad (3)$$

with $R_W$ and $W$ denoting DNN architecture and the weights to be learned, respectively. $f$ and $g(W)$ are the measure of mismatch and regularizer to avoid overfitting, respectively. $\tilde{\xi}_m$ is the approximated contrast obtained by back-prorogation (BP) and dominant current (DC) in BPS and DCS, respectively, which is able to convert the inputs of DNN from measurement domain into contrast domain and consequently make the learning process easier. $\tilde{\xi}_m$ is the ground truth contrast. Contrast parameters instead of electrical properties, such as permittivities, are used as inputs and outputs because contrast parameters have subtracted the background relative permittivities from the relative permittivities so that the network focuses on learning the interesting part and avoids learning known background values.

### B. BCNN for ISPs

As shown in Fig. 1, different with traditional convolutional neural network, BCNN replaces the deterministic weights in...
The predicted uncertainty captures most of the features presented in true absolute error without the knowledge of ground truth.

TABLE I

<table>
<thead>
<tr>
<th></th>
<th>In-Range Tests</th>
<th>15% Noise</th>
<th>30% Noise</th>
<th>Cylinder data</th>
<th>Highly nonlinear data</th>
<th>Experimental Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPS (Re)</td>
<td>0.096</td>
<td>0.098</td>
<td>0.11</td>
<td>0.1</td>
<td>0.467</td>
<td>0.14</td>
</tr>
<tr>
<td>DCS (Re)</td>
<td>0.093</td>
<td>0.094</td>
<td>0.098</td>
<td>0.076</td>
<td>0.544</td>
<td>0.13</td>
</tr>
<tr>
<td>BPS (CC)</td>
<td>0.78</td>
<td>0.77</td>
<td>0.76</td>
<td>0.74</td>
<td>0.6</td>
<td>0.81</td>
</tr>
<tr>
<td>DCS (CC)</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.81</td>
<td>0.31</td>
<td>0.83</td>
</tr>
</tbody>
</table>

network by distributions over these parameters [33]. Following the BPS and DCS, BCNN still chooses the approximate contrasts obtained from BP/DC and the ground truth as the inputs and labels, respectively.

We denote a total number of $N_s$ training data pairs as $(X, Y) = \{x_n, y_n\}_{n=1}^N$ with $X$ and $Y$ representing inputs and labels in BCNN, respectively. Given a specific test input $x$, the predictive distribution $p(y|x, X, Y)$ can be modeled by averaging over all possible weights (also known as marginalization):

$$p(y|x, X, Y) = \int p(y|x, W)p(W|X, Y)dW \quad (4)$$

Here, $p(y|x, W)$ describes the predictive distribution given input $x$ and weights $W$, which can be understood as data uncertainty (or aleatoric uncertainty). The posterior distribution $p(W|X, Y)$ captures the set of possible weights given the training data pairs $(X, Y)$, which can be understood as model uncertainty (or epistemic uncertainty) [35].

To model the data uncertainty in ISPs with BCNN, we define a multivariate Gaussian distributed likelihood function as

$$p(y|x, W) = \prod_{j=1}^M p(y_j|x, W) \quad (5)$$

with

$$p(y_j|x, W) = \frac{1}{\sigma_j \sqrt{2\pi}} e^{-\frac{(y_j - \hat{y}_j)^2}{2\sigma_j^2}} \quad (6)$$

In the training stage, for each input $x$, BCNN has its corresponding label $y$ with $y_j$ denoting its $j$th pixel value. The mean $\hat{y}_j$ and standard deviation $\sigma_j$ of the Gaussian distribution are the first and the second channels of output layer in BCNN at the $j$th pixel, respectively. $M$ is the total number of pixels in $y$.

The Gaussian distributed data uncertainties are common in ISPs. In practice, other likelihood functions, such as Laplacian distributed likelihood function, can also be used in accordance with practical situations.

By taking negative and logarithm operations on (5), a negative log-likelihood function, i.e., loss function $L(W)$ is
Fig. 3. Example two (in-range tests for DCS): Reconstructed relative permittivity profiles from scattered fields with 5% noise. The true absolute error is calculated from the ground truth and output following (14), and the predicted uncertainty is output from BCNN. The predicted uncertainty captures most of the features presented in true absolute error without the knowledge of ground truth.

obtained [33]:

\[ L(W) = \frac{1}{N_s M} \sum_{n=1}^{N_s} \sum_{j=1}^{M} \frac{1}{2} \sigma_{j,n}^{-2} |y_{j,n} - \hat{y}_{j,n}|^2 + \frac{1}{2} \log \sigma_{j,n}^2 \]  

Here, a total number of \( N_s \) data pairs in \( \{X, Y\} \) is considered in the training stage. In ISPs, \( y_{j,n} \) is the the label, i.e., ground-truth contrast when training the BCNN. It is important to note that, \( \hat{y}_{j,n}, \sigma_{j,n} \) are the first channel and the second channel in the output layer, respectively, where both of them depend on \( W \) and update with the change of \( W \) during the process of minimizing (7). The two terms in (7) are directly calculated from the output of the network for each trial \( W \). In other words, there is no need to know the ground-truth standard deviation \( \sigma_{j,n} \) in (7).

In (7), the loss function consists of a reconstruction error term \( \frac{1}{2} \sigma_{j,n}^{-2} |y_{j,n} - \hat{y}_{j,n}|^2 \) and an uncertainty term \( \frac{1}{2} \log \sigma_{j,n}^2 \), where the latter serves as a regularization term to balance the residue and uncertainty. More specifically, without the regularization term, the loss function would easily reach to a point with a large uncertainty. In practice, to avoid potential zero at the denominator part of loss function, the variance in loss function is usually replaced by log variance with

\[ v_{j,n} = \log \sigma_{j,n}^2. \]  

(8)

The final loss function is therefore defined as:

\[ L_f(W) = \frac{1}{N_s M} \sum_{n=1}^{N_s} \sum_{j=1}^{M} \frac{1}{2} e^{-v_{j,n}} |y_{j,n} - \hat{y}_{j,n}|^2 + \frac{1}{2} v_{j,n} \]  

(9)

To evaluate the model uncertainty, it is important to evaluate the posterior \( p(W | X, Y) \) given our observations pairs \( \{X, Y\} \) in (4). However, \( p(W | X, Y) \) is not known analytically, thus it is hard to perform inference in BCNN and instead an approximation has to be made.

In this paper, we cast dropout, i.e., individual neural network nodes are dropped out of the net with a specific probability (dropout rate) so that a reduced network is left, as approximate inference in Bayesian neural networks. Specifically, dropout variational inference (DVI) [34] is used to approximate the posterior \( p(W | X, Y) \) by a simple distribution \( q(W) \), where dropout is performed on the convolution layers for both training and testing stages (also referred to as Monte Carlo dropout). The detailed expression of \( q(W) \), which can be found in [34], is neglected in this work since one can extend our method to other deep-learning based ISP solvers without knowledge of it. It has been shown that the approximate inference in Bayesian neural network is related with dropout by Bernoulli approximating variational distribution, where dropout and Bayesian neural network result in the same model parameters that best explain the data [34]. Therefore, the predictive distribution \( p(y | x, X, Y) \) in (4) can be approximated by Monte Carlo integration as:

\[ p(y | x, X, Y) \approx \int p(y | x, W) q(W) dW \approx \frac{1}{K} \sum_{k=1}^{K} p(y | x, W^{(k)}), \]  

(10)

where \( K \) is the total number of times tested with dropout.

To summarize, in the test stage, we can obtain the predicted uncertainty map \( \hat{\sigma} \) with its \( j \)th pixel value \( \hat{\sigma}_j \) derived from...
Fig. 4. Example three (out-of-range tests for BPS): Reconstructed relative permittivity profiles from scattered fields. The first two columns have different noise level with the training data. The third column has different categories of profile with training data, and the fourth column has different profiles and much higher relative permittivity to test the failure cases. The predicted uncertainty of the failure case is much larger than those of successful cases.

the law of total variance [35]:

\[
(\sigma_t^2) = \text{Var}(y|x, X, Y) = E[\text{Var}(y|W, x, X, Y)] + \text{Var}(E[y|W, x, X, Y]) \\
\approx \frac{1}{K} \sum_{k=1}^{K} (\sigma_j^{(k)})^2 + \frac{1}{K} \sum_{k=1}^{K} (\hat{y}_j^{(k)} - \tilde{y}_j)^2
\]

(11)

Here, \( X, Y \) in the second equality are eliminated since once conditioned on model weights, the prediction is no longer depends on training data pairs \( \{X, Y\} \). The approximate equation is made due to the use of Monte Carlo integration.

In the test stage, the standard deviation \( \sigma_j^{(k)} \) and mean \( \hat{y}_j^{(k)} \) in (11) are directly predicted from the second and first channel of the output layer at the \( k \)th test, respectively. Further, we have the reconstructed contrast map \( \tilde{\xi} \) with its \( j \)th pixel value \( \tilde{y}_j \) being

\[
\tilde{y}_j = \frac{1}{K} \sum_{k=1}^{K} \hat{y}_j^{(k)}.
\]

(12)

The square roots of the first term \( \frac{1}{K} \sum_{k=1}^{K} (\sigma_j^{(k)})^2 \) and the second term \( \frac{1}{K} \sum_{k=1}^{K} (\hat{y}_j^{(k)} - \tilde{y}_j)^2 \) in (11) can be understood as data and model uncertainties, respectively. The data uncertainty is able to capture the data imperfections such as data noise and out-of-range testing data. The model uncertainty accounts for uncertainty in the model parameters and is able to characterize the robustness of DNN structures.

C. Implementation Procedures

One of the advantages of the proposed BCNN for ISPs is that it is easy to implement and flexible to be adapted to the existing deep learning schemes in solving ISPs. Specifically, for the BPS and DCS, the implementation procedures of solving ISPs with BCNN are summarized as follows:

- **Step 1)** Prepare the training data \( \{X, Y\} \): The inputs \( X \) are the approximated contrasts that are generated from BP/DC. The labels \( Y \) are ground-truth contrast. Namely, same as (3), \( X = \xi_a^m \) and \( Y = \xi_m^m \).
- **Step 2)** Build the BCNN: the network structure is the same as the BPS/DCS that is detailed in [19] except that one more channel is added on the output layer to represent the uncertainty of prediction as presented in Fig. 1. Moreover, dropout is added on the convolutional layer, which changes the concept of learning all the weights together to learning a fraction of the weights in the network in each training iteration.
- **Step 3)** Train BCNN with a specific dropout rate: The custom loss function in (9) is used with \( \hat{y}_j \) and \( \sigma_j \) being the first channel and the second channel of output layer, respectively. The approximated contrasts, i.e., \( \xi_m^m \) in (3), that are generated from BP/DC are fed into the input of neural network. \( y_j \) serves as the label of the training process, i.e., \( y_j \) equals to the \( j \)th pixel value of ground truth \( y \), i.e., \( \xi_m^m \) in (3).
- **Step 4)** Test the trained BCNN with a specific input \( x \) for \( K \) times by randomly dropping the weights in the convolution layers with a specific dropout rate: Due to the dropout, both channels of the output layer differ each time for a specific input \( x \). To obtain the predicted contrast
map $\xi$, the outputs in the first channel are averaged over $K$ times as $\bar{y}_j = \frac{1}{K} \sum_{k=1}^{K} \hat{y}_j^{(k)}$ following (12), where the physical meaning of $\bar{y}_j$ is the $j$th pixel of the predicted contrast map $\bar{\xi}$. With the output in the second channel $\sigma_j^{(k)}$, the predicted uncertainty map $\bar{\sigma}$ is consequently calculated following (11).

D. Computational Cost

In the proposed BCNN for ISPs, the training procedures are totally the same as those in BPS and DCS except that dropout layers are added, where the computational cost is much smaller than convolution layers. Thus, the computational cost of the proposed method is the same as that of BPS and DCS in training stage. In the testing stage, as presented in step 4) of the above section, it is true that we need to test the trained network for a specific input $K$ times by randomly dropping the weights. However, all these tests are independent with each other, and can be tested in parallel. Therefore, when parallel computation is available, the computational cost of the proposed method is also the same as existing BPS or DCS in the testing stage.

III. Tests with Numerical and Experimental Data

A. Numerical Setup

In this work, we evaluate the proposed uncertainty quantification framework under the recently proposed two deep learning schemes, i.e., BPS and DCS in [19]. Specifically, in the training process, we randomly generate 5000 profiles with relative permittivity in the range of 1.5-2 using the handwritten digits in MNIST datasets to represent scatterers in DOI, among which 500 profiles are used for validation process. Some representative examples of these profiles are also presented in the first row of Fig. 2. To qualify the uncertainties in BPS and DCS, all the other parameters setups are kept the same as those in [19].

When generating the data, scattered fields are collected by 32 receivers, where 16 line sources are used to illuminate the DOI. In the training stage, 5% Gaussian noise are added to the scattered field. In this work paper, we set the hyperparameters of the network structures as follows: Learning rate, dropout rate, batch size, total number of test times $K$ and weight decay equal to $1e^{-4}$, 0.1, 20, 16, and $1e^{-6}$, respectively. A maximum of 100 epoches are set in the training stage. It is noted that, these hyperparameters are empirically chosen and a range of values instead of one single value presented here works for the proposed method. More specifically, for learning rate, batch size, and weight decay, we simply use the values most commonly seen in the computer science community. For dropout rate, according to our experience, dropout between 0.05 to 0.3 works well for the proposed method. For total number of test times $K$, since it represents sampling times and any sufficient large value should work. According to our observations, all the values larger than 10 works well for the proposed method.

To quantitatively evaluate the image qualities of the reconstructions, average relative error $R^e$ is defined as

$$R^e = \frac{1}{L_t} \sum_{L_t} ||\tilde{\varepsilon}_r - \varepsilon_r||_F / ||\varepsilon_r||_F$$

(13)

with $\tilde{\varepsilon}_r$ and $\varepsilon_r$ denoting true and reconstructed relative permittivity.
Fig. 6. Representative histograms to illustrate the individual relative error of the proposed BCNN in Table I for (a) In-Range Tests (b) Highly nonlinear data under BPS.

Fig. 7. Nonlinear correlation distribution (NC) map of out-of-range tests for test #5, test #6, test #7, and test #8 in (a) Fig. 4 (b) Fig. 5. The sharp peak in the correlation distribution means the predicted uncertainty is highly correlated with ground-truth error, whereas the noisy correlation distributions means the correlation between them is low. The horizontal axes refer to “pixels”, and vertical axis is denoted as “a.u.”.

Fig. 2. Numerical Results

In the first and second examples, we test the proposed BCNN on BPS and DCS with the in-range data. For each scheme, 100 random tests are conducted. As presented in Figs. 2 and 3, four representative results are presented for BPS and DCS, respectively. It is seen that both BPS and DCS obtain satisfying reconstructions for in-range tests. More importantly, it is found that the true absolute error highly correlates with the predicted uncertainty from BCNN. In other words, the predicted uncertainty is able to capture the features of the errors without knowing the ground truth. It is also interesting to find that the boundary of scatterers, shape edges, and inner cylinders tend to have larger uncertainty or true absolute error than other parts of the reconstructed scatterers. To further quantitatively evaluate the performance, 100 test are conducted and the average relative error $R_e$ are presented in Table I. The average CC values are also included in Table I to show the correlation between the predicted uncertainty and true absolute error.

In the third and fourth examples, we test the proposed method with out-of-range data under the framework of BPS and DCS. We consider four out-of-range cases, where the level of noise, the shape of scatterer, or the contrast of scatterer are out of the training range. For each out-of-range case, 100 random tests are conducted. Figs. 4 and 5 show the representative results for BPS and DCS, respectively. Test #5 and Test #6 in Figs. 4 and 5 show the reconstructed results and predicted uncertainties when 15% and 30% Gaussian
noise are added in the scattered field. Satisfying results are obtained for both high-level noise contamination cases, which also suggests the robustness of the proposed method to noise contaminations. Test #7 shows the results when testing on cylinder data with the network trained previously with MNIST data. It is found that the predicted uncertainty of the proposed BCNN is still able to capture the features showing in the true absolute error. These results all verify the generalization capability of the proposed method.

In test #8, we purposely increase the difficulties of reconstructions largely by increasing the relative permittivity of the testing scatterers but keeping the training data unchanged. Therefore, not only the testing profiles differ with training data significantly, but also the nonlinearities of the ISPs (multiple scattering effect) are dramatically increased. It is shown in test #8 of Figs. 4 and 5 that both methods fail in reconstructions and both the true absolute error and predicted uncertainty are significantly increased, where the latter also gives us an indicator to monitor the “failure” results. It is known that the current DLS easily fails in solving general ISPs for out-of-range scatterers with permittivity larger than 3. To statistically evaluate the performance when BCNN fails the reconstructions, we have randomly generated 100 cylinder scatterers with relative permittivity 3.5 and marked as “Highly nonlinear data”. The quantitative results for all the out-of-range tests are also included in Table I, which is averaged over 100 tests for each case. Further, two representative histograms to illustrate the individual relative error of the proposed BCNN in Table I are also presented in Figs. 6(a) and (b).

In order to present the correlation between the predicted uncertainty and true absolute error in a more straightforward way, we further calculate the nonlinear correlation distribution (NC) for out-of-range tests and present them in Fig. 7. In NC, when the two parameters are highly correlated with each other, a sharp peak will show, whereas when the two parameters are not related with each other, noisy sidelobes will show. It is seen from Fig. 7, for both BPS and DCS, nonlinear correlation distributions of Test #5, Test #6, and Test #7 show sharp peak, but noisy sidelobes are observed for Test #8. For the failure cases, both the CC values in Table I and NC in Fig. 7 suggest that although one can observe the significant increase of predicted uncertainty when the reconstruction fails, the predicted uncertainty is not correlated with true absolute error any more. Usually, to quantitatively evaluate correlation effects in NC, peak-to-correlation energy (PCE) [38], i.e., $\text{PCE} = \frac{N C_m}{\sum_{j=1}^{M} N C_j}$ with $N C_m$ denoting maximum values of in NC, is used. In our results, it is observed that when PCE<0.03 (representing the lowly correlating with each
suggests that the proposed method is able to predict pixel-based uncertainty of the reconstructions. Firstly, a close correlation between the predicted uncertainty and the true absolute error is observed. In particular, the correlation coefficient and nonlinear correlation distribution are defined to calculate the correlation between the true absolute error and predicted uncertainty. It quantitatively demonstrates that the predicted uncertainty is highly correlated with the true absolute error. Fifthly, the predicted uncertainty is directly output from the network and it avoids complicated computations.

The proposed framework in uncertainty quantification is worth further studying to solve challenging topics in ISPs, which will be our future work. For example, it is difficult to deal with highly nonlinear ISPs [40], and failures in the reconstructions are common when solving these problems. Therefore, building a model which can supervise the uncertainty and give instructions to the algorithms is highly desirable for highly nonlinear ISPs.

IV. CONCLUSION

It is critical to understand what an algorithm does not know when it is used for reconstructions in ISPs. Most recently, deep learning algorithm has become an attractive, powerful, and promising approach to solve the ISPs. Nevertheless, all of these methods assume that the reconstructions from the trained network are trustable, which is not true in practice. Different with traditional objective-function based approaches that rely on physical principles, the learning approach is a data-driven approach, where the reconstructions are highly dependent on whether the test data is close to the training data and also on the architecture of learning model. Therefore, it is extremely desirable to build a framework that allows us to estimate the uncertainty or error of the reconstructions predicted by the network.

In this paper, we proposed a framework of BCNN that is able to solve ISPs and at the same time predict the pixel-based uncertainty of the reconstructions. Firstly, a close correlation between the predicted uncertainty and the true absolute error indicates that the proposed framework is able to capture most of the features shown in the true absolute error, where the latter is known only if we have the information of ground truth. Secondly, the proposed method is highly flexible, which can be applied to most of the DLSs in the community of ISPs. In this paper, we have tested it on two most recent deep learning schemes to illustrate the concept. Thirdly, the proposed framework exhibits the properties of robustness to noise contaminations in both BPS and DCS by testing on 15% and 30% Gaussian noise contaminations. Fourthly, to quantitatively evaluate the proposed framework, correlation coefficient and nonlinear correlation distribution are defined to calculate the correlation between the true absolute error and predicted uncertainty. It quantitatively demonstrates that the predicted uncertainty is highly correlated with the true absolute error.

REFERENCES


