Visual Analysis of Prediction Uncertainty in Neural Networks for Deep Image Synthesis

Soumya Dutta, Faheem Nizar, Ahmad Amaan, and Ayan Acharya

Abstract—Ubiquitous applications of Deep neural networks (DNNs) in different artificial intelligence systems have led to their adoption in solving challenging visualization problems in recent years. While sophisticated DNNs offer an impressive generalization, it is imperative to comprehend the quality, confidence, robustness, and uncertainty associated with their prediction. A thorough understanding of these quantities produces actionable insights that help application scientists make informed decisions. Unfortunately, the intrinsic design principles of the DNNs cannot beget prediction uncertainty, necessitating separate formulations for robust uncertainty-aware models for diverse visualization applications. To that end, this contribution demonstrates how the prediction uncertainty and sensitivity of DNNs can be estimated efficiently using various methods and then interactively compared and contrasted for deep image synthesis tasks. Our inspection suggests that uncertainty-aware deep visualization models generate illustrations of informative and superior quality and diversity. Furthermore, prediction uncertainty improves the robustness and interpretability of deep visualization models, making them practical and convenient for various scientific domains that thrive on visual analyses.

Index Terms—Uncertainty, Deep Learning, Visualization, Monte Carlo Dropout, Deep Ensembles, Image Synthesis, CNN.

1 Introduction

The indisputable success of deep learning techniques [1] has ushered new research frontiers in the scientific visualization community. The novel ideas pioneered by the latest development in learning theory find applications in some of the primary areas of data visualization: data representation and generation, the genesis of visualization, predictive analytics, and feature extraction [2]. Unlike other deep learning applications, scientific visualization and data analysis applications require a thorough understanding of the underlying models' quality, robustness, confidence, and prediction uncertainty. Such knowledge helps domain scientists make informed decisions for furthering domain-specific discovery [3], [4]. However, the literature survey reveals that a thorough uncertainty analysis for the deep visualization models is still missing – a gap that this paper attempts to bridge.

Without any insight regarding the robustness, uncertainty, and sensitivity of predictions of the deep visualization models, application scientists could get misled into flawed judgments and biased interpretations of the data. In contrast, visualization models that convey prediction confidence and sensitivity information can facilitate building trust among the domain scientists, and as a result, such uncertainty-aware models could be easily adopted in real-life scientific applications. Hence, it is essential to thoroughly study the effectiveness and robustness of the uncertainty-aware deep visualization models. While the core machine learning community utilizes several approaches for estimating uncertainty in DNNs, their pertinence for deep visualization models is yet to be explored. From the existing deep uncertainty estimation methods, it has been found that the ensemble-based methods often outperform other alternatives by producing more accurate predictions, and by measuring the variations from ensemble predictions, the uncertainty can also be estimated [5], [6]. However, deep ensembles inherently suffer from high training costs, and if many ensemble members are used, inference from them can also be expensive. In contrast, estimating prediction uncertainty using Monte Carlo dropout (MC-Dropout) is a pragmatic approach that is computationally feasible and produces robust uncertainty estimates [4]. The inference with the MC-Dropout method is conceptually similar to drawing samples from an implicit ensemble model. Theoretically, MC-Dropout is also closely related to approximate inferencing in deep Gaussian processes [4], [7]. More importantly, the generalizability, robustness, and wide applicability of the MC-Dropout with minimal modification to the network architecture makes it a promising candidate for uncertainty estimation in deep visualization models.

This work comprehensively studies two principled deep uncertainty estimation techniques: (1) deep ensembles and (2) MC-Dropout-based methods for deep learning-based volume-rendered image synthesizing tasks. Given input view parameters, our deep uncertainty analysis framework produces volume-rendered images and corresponding fine-grained pixel-wise uncertainty and error estimates. The wide applicability of such deep learning model-driven image-based visualization for scientific data analysis has been thoroughly demonstrated in several recent works [8], [9], [10]. To comprehend, compare, and contrast the uncertainty estimates of both of these methods, we provide a visual analytics tool that enables in-depth exploration of the characterization of model uncertainty and sensitivity across the entire view space. Our tool also offers interactive view space error visualization capabilities that help the users

- *S. Dutta is with IIT Kanpur.*
  E-mail: soumyad@cse.iitk.ac.in
- *S. Dutta, F. Nizar, and A. Amaan are with Indian Institute of Technology Kanpur (IIT Kanpur).*
- *A. Acharya is with LinkedIn Inc.*

Manuscript received April xx, xxxx; revised August xx, xxxx.

comprehend the accuracy of the predicted images.

We focus on two broad analysis goals as outlined below:

Comparing Uncertainty Estimation Methods: The first objective is to compare deep ensembles and MC-Dropout methods for estimating uncertainty in deep visualization models [5], [6]. These methods are chosen for their ability to add uncertainty information to existing neural networks without significant changes. The study aims to determine if both methods produce similar uncertainty patterns. If they do, the more computationally efficient MC-Dropout method may be preferable for resource-constrained applications, and the study also examines how many ensemble members are needed for reliable uncertainty estimates.

Utilizing Fine-Grained Uncertainty Information: The second objective is to showcase the value of detailed uncertainty and sensitivity information from models. This information is particularly useful when making predictions for unknown scenarios without ground truth data. Predicted uncertainty can convey model confidence, even when error measurements are unavailable. The study also investigates the relationship between predicted uncertainty and prediction error. If low uncertainty consistently corresponds to low error (and vice versa), scientists can trust the model's output based solely on uncertainty information. Conversely, the study explores scenarios where low uncertainty might still lead to high error and vice versa. These analyses aim to enhance our understanding of prediction reliability, accuracy, and sensitivity in DNNs for scientific applications.

Below, we succinctly summarize our contributions:

- We propose a comparative framework for comprehending, comparing, and contrasting uncertainty, error, and sensitivity estimates generated from deep ensembles and MC-Dropout-based methods for image-based deep volume visualization models.
- We develop an interactive visual analytics tool for effectively exploring fine-grained prediction uncertainty, error, and sensitivity estimates from multiple image-based deep volume visualization models. We demonstrate how incorporating uncertainty estimates makes the model predictions more informative, trustable, and interpretable.

2 Research Background and Uncertainty in Deep Neural Networks

In this section, first, we briefly summarize the relevant research on deep learning for scientific visualization and uncertainty visualization methods. Then, we introduce various approaches to quantifying uncertainty in deep neural networks and focus on two well-known principled approaches of uncertainty estimation in deep neural networks that are explored in this work for deep image synthesizing models.

2.1 Deep Learning for Scientific Visualization

The application of deep learning in scientific visualization is manifold. Lu *et al.* [11], and Weiss *et al.* [12] propose techniques for generating compact neural representations of scientific data. Visualization of scalar field data using volume-rendered images is studied by Hong *et al.* [9], He *et al.* [8], and Berger *et al.* [10] and using isosurfaces by Weiss *et al.* [13]. Weiss *et al.* [14] further uses an adaptive sampling-guided approach for volume data visualization. Another research area of focus is the generation of spatiotemporal super-resolution volumes from low-resolution data [15], [16], [17], [18]. New models for domain knowledge-aware latent space generation techniques for scalar data are also proposed for compressing the volume data [19]. Furthermore, DNNs are used as surrogates for the generation of visualization and exploration of parameter spaces for ensemble data [8], [20], [21]. Variable-to-variable translation technique for scientific data is proposed by Han *et al.* [22]. Han and Wang [23] explore Graph convolutional networks for learning surface representations. For a more comprehensive review of other deep learning applications in scientific visualization, please refer to the state-of-the-art survey [2].

2.2 Uncertainty in Scientific Visualization

Visualizing uncertainty in scientific data analysis is a well-studied research area. One of the earliest summaries of uncertainty visualization techniques is by Pang *et al.* [24]. Potter *et al.* [25] conduct the visualization of spatial probability distributions defined over triangular meshes, which is anteceded by a taxonomy of uncertainty visualization techniques [26]. Brodlie *et al.* [27] report visualization methods that are augmented with facilities for uncertainty estimation. Liu *et al.* [28] use flickering to represent uncertainty in volume data. Athawale *et al.* [29] further explore uncertainty in volume rendering using non-parametric models.

Uncertainty visualization techniques for isocontouring methods are also well-studied. Pöthkow *et al.* [30] compute level crossing probability of adjacent points, which is further enhanced to calculate the level crossing probability for each cell in [31]. Whitaker *et al.* [32] inspect the visualization of uncertainty in an ensemble of contours using contour boxplots. Visual analysis of fiber uncertainty is studied in [33]. Bonneau *et al.* [3] compile a state-of-the-art survey of many uncertainty visualization techniques. Recently, Gillmann *et al.* summarize uncertainty visualization techniques for image processing applications [34] and medical imaging [35]. Kamal *et al.* [36] describe the latest challenges and progress in introducing uncertainty in visualization research.

2.3 Uncertainty in Deep Neural Networks

The wide popularity of deep learning techniques does not accord with the overarching concern about their interpretability, robustness, and generalizability in real-world applications [37]. In general, the inability of the DNNs to cater to uncertainty estimates can undermine the superior empirical gains they achieve in various applications like natural language processing, computer vision, and visual analytics. Since the sources and characterization of uncertainty vary widely from one application domain to the other, formalizing and generalizing the techniques to measure and quantify the uncertainty in an *application-agnostic* way remains an open challenge. In what follows, we briefly explain some sources of uncertainty in DNNs and different methods adopted to emulate and address the same.

The predictive uncertainty [38], [39] of a DNN can be broadly categorized into two groups – *data or aleatoric uncertainty* and *model or epistemic uncertainty*. Data uncertainty arises due to errors and noise in measurement systems. The modeling (epistemic) uncertainty, on the other hand, can

be due to multiple reasons. First, the state-of-the-art DNN models produce a compact representation of the real-world system that generates the observations. Such parsimony often leads to prediction errors and associated uncertainty. Second, over-parameterized networks often exhibit *double descent* [40] – a phenomenon that leads to sharper prediction and lower uncertainty. Third, many DNNs must be carefully tuned using dropout [4], [41], learning-rate warm-up and decay [42], regularization [43] etc. Different decisions for such parameters lead to different learned configurations.

2.4 Techniques for Modeling Uncertainty in DNNs

In the following, we first briefly highlight several existing methods of uncertainty modeling in DNNs and then discuss deep ensembles and MC-Dropout-based uncertainty estimation in detail, which are employed in our work.

Deterministic Methods. Enabling deterministic models with prediction uncertainty is counter-intuitive. However, one can explicitly model and train a network to quantify uncertainty [44]. Alternative approaches use additional components, extrinsic to the prediction models, to produce uncertainty estimates [45].

Bayesian Methods. Probabilistic Bayesian methods enforce flexible prior distributions on the parameters of DNN to facilitate precise estimation of the predictive uncertainty [38], [39]. In literature, stochastic gradient MCMC [46], and variational inference [47] are such techniques. However, they are often slower compared to the first-order [48] and second-order methods [49].

Test-time Augmentation Methods. The test-time augmentation methods derive inspiration from the utility of ensemble methods and enrich model training using adversarial examples [50], [51]. The key idea is to apply data augmentation during test time and infer prediction uncertainty.

2.5 Ensemble Method

Ensemble methods work on the presumption that a set of learners of similar capacity is often better than a single learner [52]. Besides improving the generalization error, ensemble methods provide a natural way to compute the prediction uncertainty by evaluating the variety across different predictions [53], [54]. Hence, one can adopt ensemble methods for quantifying the predictive uncertainty of DNNs [5], [39], [55]. To that end, Lakshminarayanan *et al.* propose deep ensembles [39] where individual DNNs are equipped with two heads that model both the prediction and the corresponding uncertainty. Moreover, the shuffling of the training data and a random initialization of the training process induces a good variety in the models to predict the uncertainty for the given architectures and data sets. Subsequent works inspired by deep ensembles [5], [6], [56], [57] find ensemble methods for DNNs often outperform techniques that rely on Monte Carlo-based dropout and probabilistic back-propagation; and are more immune to changes in data distribution.

2.6 MC-Dropout Method

Dropout. At a high level, Dropout [4], [41] is a regularization technique that prevents DNNs from overfitting on the training data by randomly masking a subset of the weights during the forward and backward propagation. However, deviating from this traditional application of dropout, Gal *et al.* [4] discovered that dropout training in DNNs can be cast as approximate Bayesian inference in deep Gaussian processes. A pivotal consequence of such wisdom is that dropout, when used in test time, can provide convenient information about model (epistemic) uncertainty, which is usually derived by collecting Monte Carlo (MC) samples of the network output.

In the context of a supervised learning problem, consider that we have a set of training data denoted as $\mathcal{D} := \{\mathbf{x}_n, y_n\}_{n=1}^N$. In this scenario, we aim to model the conditional probability $p_{\boldsymbol{\theta}}(y_n|\mathbf{x}_n)$ using an NN that is characterized by its parameters $\boldsymbol{\theta}$. When we talk about applying dropout to a neural network, it essentially involves the process of adjusting the weights of each layer individually. This adjustment is done by using a random mask, denoted as $\mathbf{z}_n$, which follows either a Bernoulli or Gaussian distribution specific to the data point. These masks, represented as $\mathbf{z}_n$, are iid drawn from a prior distribution denoted as $p_{\boldsymbol{\eta}}(\mathbf{z})$, and this distribution itself is parameterized by $\boldsymbol{\eta}$ [41], [58].

Dropout training can be viewed as approximate Bayesian inference [4], [59]. To elaborate further, we can interpret the goal of training a supervised learning model with dropout as aiming to maximize a log-marginal-likelihood:

$$\log \int \prod_{n=1}^{N} p(y_n|\mathbf{x}_n, \mathbf{z}) p(\mathbf{z}) d\mathbf{z}.$$

To maximize this challenging likelihood, it is common practice to turn to variational inference techniques [60], [61] that introduce a variational distribution $q(\mathbf{z})$ on the random mask $\mathbf{z}$ and optimizes an evidence lower bound (ELBO):

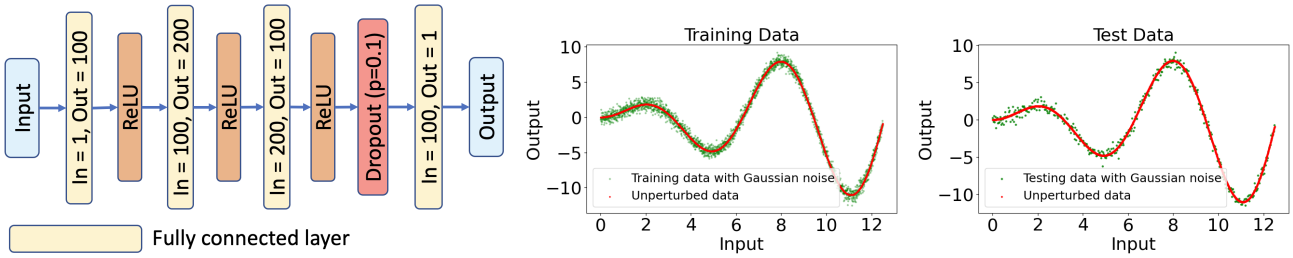
$$\begin{aligned}\mathcal{L}(\mathcal{D}) &= \mathbb{E}_{q(\mathbf{z})}\left[\log \tfrac{\prod_{i=1}^{N} p_{\boldsymbol{\theta}}(y_i|\mathbf{x}_i,\mathbf{z}) p_{\boldsymbol{\eta}}(\mathbf{z})}{q(\mathbf{z})}\right] \\ &= \sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{z})}\left[\log p_{\boldsymbol{\theta}}(y_n|\mathbf{x}_n, \mathbf{z}_n)\right] - \mathrm{KL}(q(\mathbf{z})||p_{\eta}(\mathbf{z})),\end{aligned}$$

where $\mathrm{KL}(q(\mathbf{z})||p_{\boldsymbol{\eta}}(\mathbf{z})) = \mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z}) - \log p(\mathbf{z})]$ represents a regularization term based on Kullback-Leibler (KL) divergence. Whether this KL term is explicitly incorporated distinguishes regular dropout [41], [58] from their Bayesian extensions [4], [62].

Uncertainty Estimation in DNNs with MC-Dropout. As mentioned above, dropout, when used during inference, enables estimation of model (epistemic) uncertainty as proposed by Gal *et al.* [4]. Estimating deep model uncertainty using MC-Dropout has become a widely popular technique, and it is performed via several stochastic forward passes through the network and averaging the outcomes. The variability (standard deviation) in outcome due to the stochastic forward passes can be quantified and interpreted as model prediction (epistemic) uncertainty. Theoretically, this uncertainty is equivalent to performing Bayesian inference in deep Gaussian processes [7].

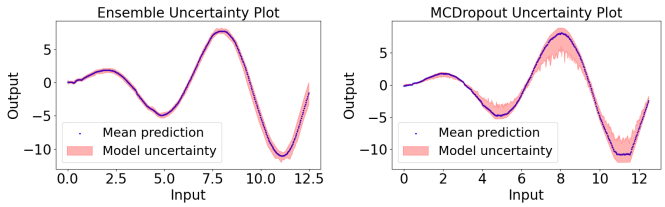
2.7 Demonstration of Ensemble and MC-Dropout Uncertainty in DNNs

Here, we consider a DNN for a regression problem. The model's architecture is shown in Fig. 1a. We add dropout



(a) DNN architecture. (b) Synthetic training data. (c) Synthetic testing data.

Fig. 1: Fig. 1a shows the DNN architecture used for demonstration. Fig. 1b and Fig. 1c show the synthetic training and test data used to demonstrate the MC-Dropout-based uncertainty estimation technique. The synthetic data is generated using the function $f(x) = xsin(x)$ with an added Gaussian noise ($\epsilon \sim \mathcal{N}(0, 0.1)$).



(a) Deep Ensemble (b) MC-Dropout

Fig. 2: These plots illustrate the estimated uncertainty for Ensemble and MCDropout methods. The blue dotted line in Fig. 2a and 2b show the mean prediction, and the light red envelope shows the prediction uncertainty.

before the final layer to capture model uncertainty during inference. The ensemble model does not employ any dropout layer. We generate synthetic training and test data from the function: $f(x) = xsin(x)$. The output of the function $f(x)$ is slightly perturbed by adding a small Gaussian noise ($\epsilon \sim \mathcal{N}(0, 0.1)$) to each data point. The corresponding training data is shown in Fig. 1b, where the green and red dots represent the training and the original data (without perturbation). Fig. 1c shows the test data. We train the model for 1000 iterations using the Adam optimizer [48] with the learning rate set to 0.001, and β_1 and β_2 set to 0.9 and 0.999, respectively. We use MSE as the loss function. To generate the ensemble, we train 50 instances of this DNN without any dropout, resulting in an ensemble of 50 members. We keep the hyper-parameters and training configurations the same as the MC-Dropout method. We randomly shuffle the training data to train each learner and subsequently combine their predictions during inference.

The inference for MC-Dropout is performed by setting the model to evaluation mode and enabling dropout. The model predicts the output $m = 100$ times for each input. The average computed over these samples provides the expected model prediction, and the standard deviation is considered the uncertainty. For ensemble methods, each of the 50 learners infers on each test example. The average computed over these outputs provides the expected model prediction, and the corresponding standard deviation is considered the ensemble prediction uncertainty.

We present the model prediction results for ensemble and MC-Dropout methods in Fig. 2a and 2b, respectively. The blue dots show the expected (mean) prediction. The light red envelope shows the extent of prediction uncertainty. According to Fig. 2a and 2b, the uncertainty estimated by the ensemble method produces a more uniform uncertainty with minor variabilities. However, the uncertainty estimates made by the MC-Dropout increase with the curvature of the data, as seen in Fig. 2b. Notably, the estimated uncertainty is much tighter when the curvature is low. So, MC-Dropout tends to produce more pronounced uncertainty in the regions where the data value changes rapidly compared to the ensemble method.

For the synthetic example, one may be tempted to conclude that the MC-Dropout produces a wider uncertainty band. However, note that the ensemble method reduces the variance without affecting the bias. Therefore, concluding about the characteristics of the uncertainty bounds for both of these methods using only this synthetic problem may not be sufficient. In Section 6, we elaborate on the error rates and uncertainty estimates in real-world data. We justify the difference by showing the flexibility of the MC-Dropout methods and recognizing that there is no need to retrain the model or store the parameters.

3 Prediction Uncertainty in Image-based Deep Visualization Models

3.1 Model Description

Our image-based deep visualization model takes view angles, P_{view} = {Azimuth ($\theta \in [0, 360]$) and Elevation ($\phi \in [-90, 90]$)}, as input and produces a volume-rendered image $\mathcal{I}$ of the data as output. The model architecture is shown in Fig. 3, inspired by insitunet [8]. First, P_{view} is passed through a few fully connected layers, and then the generated latent vector is reshaped into a 4×4 low-resolution image. Finally, the low-resolution image is up-sampled sequentially to produce an image of resolution 128×128. The model uses five residual blocks [63] to perform up-scaling *via* 2D convolution of the image. We use ReLU as the activation function throughout the network except for the last layer, where we use hyperbolic tangent (Tanh) to normalize each output in the range $[-1, 1]$. The structure of a residual block is shown as an inset on the right in Fig. 3. Note that batch normalization is used to stabilize the model during training.

3.2 Quantifying Uncertainty for MC-Dropout Method

As seen in Fig. 3, we augment a 2D dropout layer at each residual block followed by the ReLU activation [64]

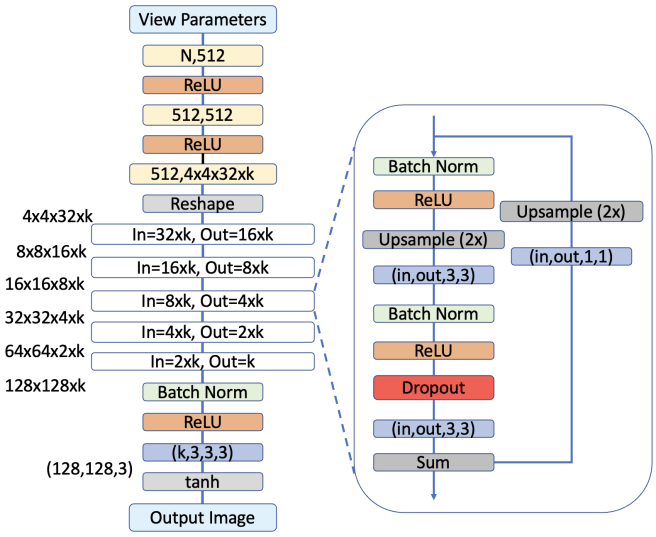


Fig. 3: Architecture of our deep visualization model with the inset on the right showing the structure of a residual block.

to enable the MC-Dropout-based model uncertainty estimation. The dropout also helps in regularization during training. Since we aim to capture the pixel-level prediction uncertainty of the model, we add the 2D dropout layers at each residual block where the 2D convolution operations take place, and the output image is synthesized step-by-step. Note that, during inference, the entire 2D channels will be dropped randomly, and as a result, the variability of the model prediction will be reflected in the output pixel values. This variability will give us the MC-Dropout uncertainty estimates. After the training is done, during inference, the dropout is first enabled. Then, for a given test sample, m stochastic forward passes are performed that produce m predicted images. Now, the average image is treated as the expected model output, and the pixel-wise standard deviation reflects the prediction uncertainty.

3.3 Quantifying Uncertainty for Ensemble Method

We use the same model shown in Fig. 3 to generate a deep ensemble of visualization models. However, in this case, we do not use any dropout layers. We generate an ensemble of 20 members by training each member separately. We use the same training data that is used to train the MC-Dropout model for training ensemble members. We randomly shuffle the training data to train each member while creating the ensemble. After the training of all the ensemble members, for a given test sample view, first, output images are generated from each ensemble member individually. Then, the average image is considered the expected output, and the pixel-wise standard deviation of all the ensemble member outputs is collected to estimate ensemble prediction uncertainty.

Loss Function and Hyperparameters. For both MC-Dropout and Ensemble method, the models are trained on a combined loss function $\mathcal{L} = \mathcal{L}_{mse} + \mathcal{L}_{feat}$. The first part of the loss function is the conventional mean squared error loss ($\mathcal{L}_{mse}$) that computes the pixel-wise differences of the predicted image with the ground truth image. The latter part is the feature reconstruction loss ($\mathcal{L}_{feat}$) [8] computed according to the output from the layer $relu1_2$ of the pre-trained VGG-19 model. The training uses batch size 64 with the Adam optimizer [48] with a learning rate set at 0.0001 and β_1 and β_2, the configurable parameters of Adam, set at 0.9 and 0.999, respectively. For the MC-Dropout model, a dropout probability $\eta = 0.1$ is used, and no dropout is used for training ensemble members. For both the MC-Dropout and Ensemble methods, all the models are trained for 1500 epochs to maintain consistency and comparability.

4 Uncertainty, Error, and Sensitivity Computation Framework

The conceptual framework for pixel-wise uncertainty and error estimation is presented in Fig. 4. During inference, for a given test viewpoint, both MC-Dropout and Ensemble methods generate a set of images. While for MC-Dropout, these images are results of Monte Carlo sampling, for the Ensemble method, each ensemble member produces a predicted image for the same input test viewpoint. These two methods are represented in two sides of Fig. 4, in Section 1 and Section 3. Now, given a set of images for a specific test input, how the pixel-wise uncertainty and error quantities are computed are shown in Section 2, which is the central part of Fig. 4. These operations, demonstrated in Section 2, are identical for both MC-Dropout and Ensemble methods.

4.1 Computation of Pixel-wise Uncertainty

To allow the users to visualize and comprehend the uncertainty in fine-grained detail, we compute prediction uncertainty at every pixel for each RGB color channel separately. The combined pixel-wise uncertainty is then estimated by adding all the channel-wise uncertainties into a single image. The uncertainty in this work is quantified by computing the standard deviation value of channel-wise pixel intensities over all the sample images, as shown in Fig. 4. The bottom half of Section 2 in Fig. 4 depicts this process.

4.2 Computation of Pixel-wise Error and Error Variance

Besides uncertainty, we also compare the predicted images with the ground truth to compute pixel-wise error values. The error values are estimated as the absolute difference in channel-wise intensity values. This error computation is performed for each image for each color channel. Then, the average error image is computed for each channel. Finally, the combined error image is constructed by adding channel-wise average error values. Since, for a specific test sample, both MC-Dropout and Ensemble methods produce multiple predicted images, we also quantify the pixel-wise error variability values by computing pixel-wise error standard deviation for each color channel. Finally, the average and combined error standard deviation image is constructed by averaging and then adding the error standard deviation values of the three color channels. These error standard deviation images reflect the robustness of the error calculation. The top half of Section 2 in Fig. 4 demonstrates this error and error standard deviation computation steps.

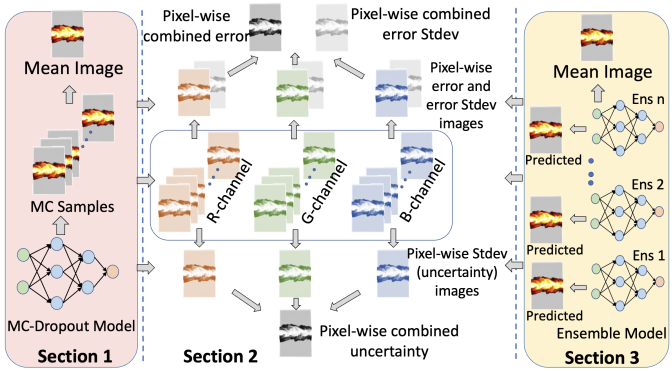


Fig. 4: Uncertainty, Error, and Error standard deviation estimation for both MC-Dropout and Ensemble method. RGB Channel-wise uncertainty and error quantities are computed for both methods.

4.3 Input Space Sensitivity and Its Robustness

Next, we utilize the backpropagation step of our differentiable visualization model to perform a sensitivity analysis of the output image to the perturbations of the input view parameters. We first generate the output image using forward propagation and compute the L_1 norm from the pixel values [8]. We then perform the backpropagation step to estimate the gradient of the L_1 norm. The absolute sum of the estimated gradients for the two view angles gives us the total sensitivity for the given input. For the MC-Dropout method, we repeat the exact computation for 100 Monte Carlo passes during inference. Finally, we compute the mean sensitivity, which reflects the expected sensitivity, and the standard deviation from the samples indicates the robustness of the estimated sensitivity. For the Ensemble method, we perform the above steps for each ensemble member separately and then take the average sensitivity value as the expected sensitivity, and the standard deviation indicates the robustness of the estimated sensitivity values.

5 Interactive Visual Analytics of Uncertainty, Error, and Sensitivity

The proposed interactive visual analytics interface is shown in Fig. 5. The primary goal of the interface is to enable users to pairwise compare and contrast the characteristics of various deep uncertainty estimation methods for deep image-based visualization models. Even though we focus on Ensemble and MC-Dropout-based uncertainty in this work, the proposed tool is not hard-wired to only these two methods. Results from any other uncertainty estimation methods can be easily loaded into this tool for comparison. The interface enables fine-grained pixel-wise analysis of prediction uncertainty, error, and model sensitivity for individual RGB color channels to reveal detailed patterns about these quantities. Since our visualization model takes view angles as inputs and generates corresponding volume-rendered images, we provide the uncertainty, error, and sensitivity pattern for the entire view space so that users can identify and study viewpoints that result in high/low uncertainty or error. This inspection provides the users insights about how prediction uncertainty and error behave in the entire view space when estimated by two different methods. In the following, we discuss various components of our visual analytics tool and their usability.

5.1 Uncertainty and Error Visualization

Our visual analytics interface has three main components: (A) Parallel Coordinates Plot (PCP), (B) Uncertainty and Error heatmap plots, and (C) Image view panel (Fig. 5). The interface presents the uncertainty and error patterns for the entire view space (Azimuth ($\theta \in [0, 360]$) and Elevation ($\phi \in [-90, 90]$)) for MC-Dropout and Ensemble method side-by-side as heatmaps (see Section B in Fig. 5). The x and y-axis of the heatmaps represent azimuth and elevation angles, respectively. Note that our tool allows comparison between two uncertainty methods at a time. If other uncertainty methods are available, users can interactively select any pair of methods using the model selection drop-down at the top. Users can also interactively load the results of different data sets for exploration using the data set selection drop-down. Placing heatmaps side-by-side allows a direct comparison of the patterns of uncertainty and error between the two selected uncertainty estimation methods. We also provide heatmaps of the error standard deviation, indicating the robustness of the error estimates. The top three heatmaps show the uncertainty, error, and error standard deviation plots for the MC-Dropout method, and the bottom three plots show the same plots for the Ensemble method. To generate the heatmaps, we sample the entire view space densely and quantify aggregated uncertainty, error, and error standard deviation for each viewpoint. First, we compute the quantities for three color channels separately and then add them to compute the final uncertainty and error value. Each cell in the heatmap indicates uncertainty/error value for a particular viewpoint. For visualization, the value at each cell is mapped to a color using a perceptually uniform sequential colormap as shown in Fig. 5. The users can interactively investigate these heatmaps for a specific image color channel by selecting the desired color channel from the top right channel selector.

The heatmaps allow interactive lasso selection to select a set of viewpoints for detailed inspection. In Fig. 5, such a lasso selection is highlighted by a yellow-colored region. Note that these heatmaps are linked views; hence, selecting a region in one heatmap automatically highlights the same region on the other heatmaps. The image view panel provides the corresponding ground truth images, predicted mean images, uncertainty, error, and error standard deviation images of the selected viewpoints on the right side (Section C in Fig. 5). Here, the users can visually compare and contrast the estimated pixel-wise uncertainty, error, and error variability for the selected views to gain detailed insight into uncertainty and error characteristics. In Fig. 5, we show the results using the Mixture Fraction (*mixfrac*) variable of Turbulent Combustion data having spatial resolution $360 \times 240 \times 60$. We observe that the regions that show the complex flame structure and its boundaries incur higher uncertainty and error than the other regions. This observation is consistent for both MC-Dropout and Ensemble methods. The users can also investigate the uncertainty, error, and error variability for each color channel on

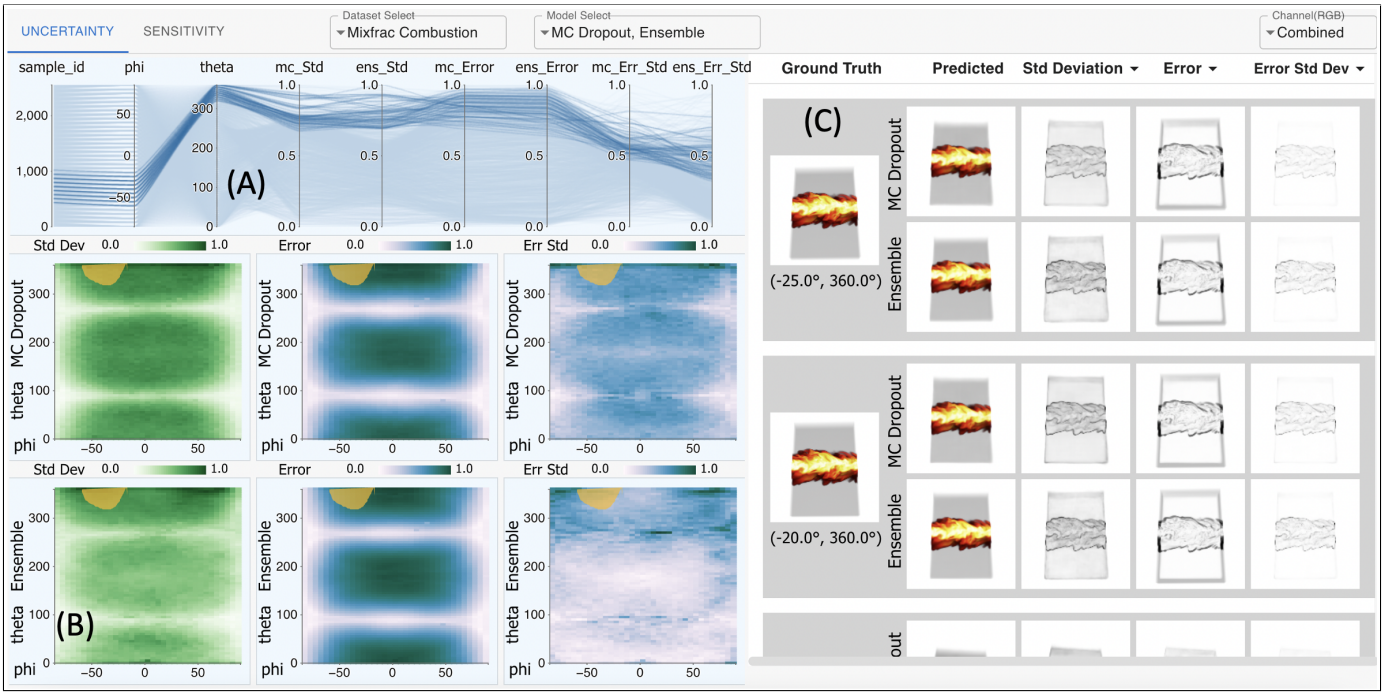


Fig. 5: Interactive Uncertainty analysis interface showing results for Combustion data. The analysis interface allows the users to effectively compare and contrast the uncertainty and error estimates produced by both MC-Dropout and Ensemble methods, where (A) shows PCP, (B) shows uncertainty and error heatmaps, and (C) shows the image view panel.

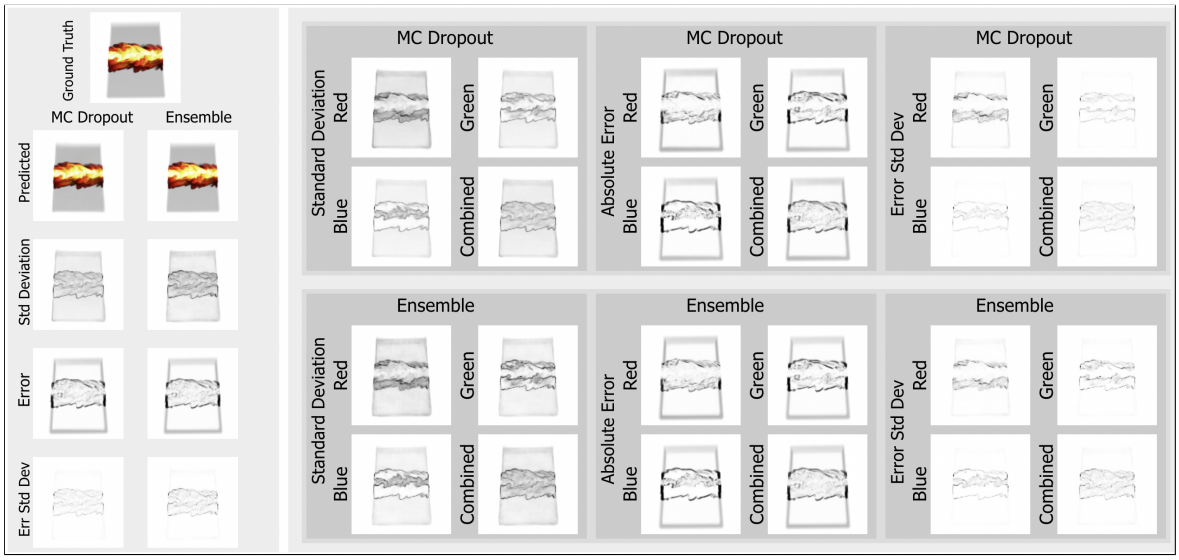
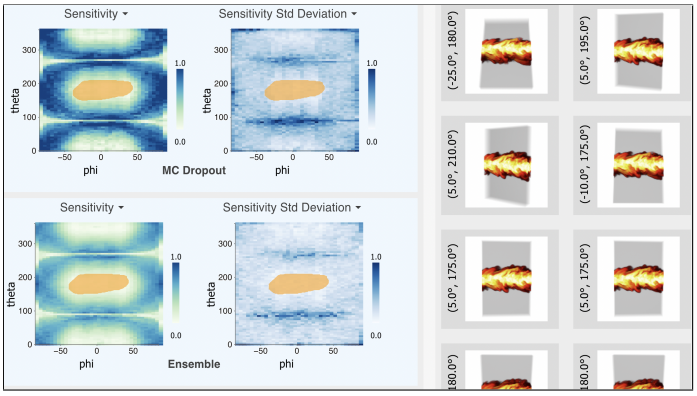


Fig. 6: Popup window for a specific viewpoint shows RGB channel-wise and combined uncertainty and error images to allow detailed comprehension of the predicted results.

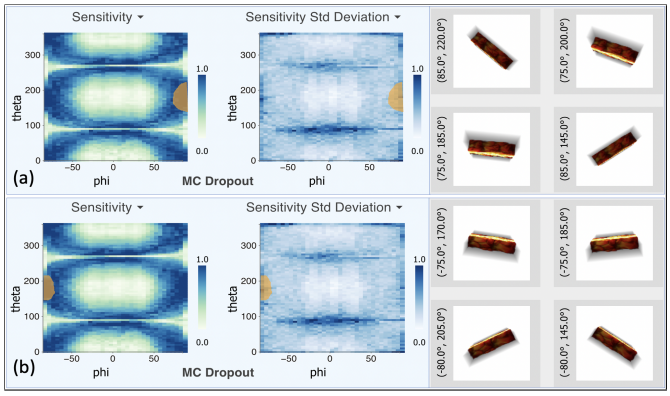
demand by clicking on an image in the Image view panel. Upon clicking, a popup window shows the RGB channel-wise results for the selected view for both MC-Dropout and Ensemble methods. Such a popup window is shown in Fig. 6, where channel-wise images are presented. By investigating the channel-wise images for the selected view, the users can study how uncertainty and error influence each color channel's prediction.

Finally, a PCP is provided (Section A) to allow comparison and correlation study of computed uncertainty, error, and error standard deviation values for the two uncertainty estimation methods. The lasso selection in the heatmaps updates the selection in PCP, and only the selected viewpoints are highlighted, as seen in Fig. 5 (Section A). The PCP maps uncertainty, error, and error standard deviation for the MC-Dropout and Ensemble method as parallel coordinate axes. Users can interactively brush single or multiple axes of interest to query a specific range of views, study the existence of correlation among the selected view, and further inspect the resulting images in the Image view panel.

By inspecting the patterns from the heatmaps, we learn that the uncertainty and error produce similar patterns in



(a) Sensitivity panel of the interface that shows sensitivity heatmaps for both MC-Dropout and Ensemble method. The low-sensitive region at the center of the heatmap is selected (shown in yellow), and corresponding images are shown on the right.



(b) Two highly sensitive regions from the heatmap plot are selected (shown in yellow), and the corresponding images are provided on the right.

Fig. 7: Sensitivity panel of our visual analytics interface is shown in Fig. 7a. The results of the *mixfrac* variable of Combustion data are shown. Fig. 7b highlights images for two different highly-sensitive view space regions.

the view space, meaning that they are correlated. A similar observation is seen from the PCP, too, where the selected viewpoints form a cluster, indicating that the views where the model produces higher uncertainty also incur higher error. Next, we observe that the overall pattern in the uncertainty and error heatmaps for MC-Dropout and Ensemble methods are also correlated. The error variability maps reveal that the error variability is smaller for the Ensemble method; hence, the error standard deviation heatmap is cleaner at the bottom compared to MC-Dropout.

5.2 Sensitivity Visualization

Besides the uncertainty and error estimates, we also compute the input view space sensitivity of the models. The computation of the sensitivity values is discussed in Section 4.3. The sensitivity measure, computed for each view, indicates how rapidly the output images change if the input view changes slightly. Since both MC-Dropout and Ensemble methods involve using multiple sample images to compute mean sensitivity, we further quantify the robustness of the estimated mean sensitivity by measuring the standard deviation of the sensitivity values. Hence, in the sensitivity visualization panel (see Fig. 7a), we present both the sensitivity and sensitivity standard deviation heatmaps. Similar to the uncertainty heatmaps, the users can select a region of interest from these heatmaps, and the corresponding viewpoints are shown on the Image panel. Using this interface, users can study the sensitivity patterns of the trained models for the entire input view space and interactively visualize viewpoints with lower/higher sensitivity.

Fig. 7a shows the sensitivity results for the *mixfrac* variable of Combustion data. The yellow selected region at the center depicts a low-sensitivity region. The corresponding views are shown on the right. We observe that the frontal views of the data that cover a larger screen area produce less sensitivity, i.e., changing the input view angles slightly for these views will not change the output image significantly. In contrast, Fig. 7b shows views of high sensitivity for two different regions in the heatmaps. We find that the side

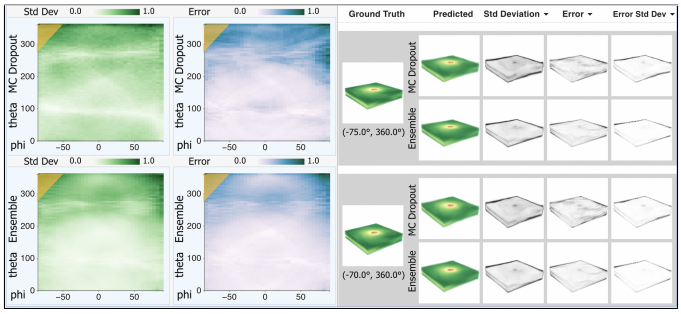


Fig. 8: Visualizing prediction uncertainty for *vel* field of Isabel data. The top two heatmaps show view space uncertainty and error plots for the MC-Dropout, and the bottom two heatmaps show similar plots for the Ensemble method. The upper left corner, the region with high uncertainty, is selected for exploration. Two sets of representative viewpoints from the selected region are shown on the right side.

views of the data are selected where the data dimension is the smallest. Naturally, when such view angles are changed, the images are likely to change rapidly, covering a larger screen space and resulting in a higher sensitivity.

5.3 Visual Analysis using Hurricane Isabel Data

We use the Velocity magnitude (*vel*) field of Hurricane Isabel data with spatial resolution $250 \times 250 \times 50$ to show the results of the proposed method. Fig. 8 shows the results of prediction uncertainty for both MC-Dropout and Ensemble methods. The uncertainty is higher for both methods for higher values of Azimuth angle (θ). To inspect images with high uncertainty, we select a region from the top left corner of the uncertainty map (shown in yellow). Results for the two most uncertain views are shown on the right among the selected viewpoints. We see that the pixels at the boundary of the *vel* field result in the highest uncertainty for both MC-Dropout and Ensemble methods. Next, we investigate the sensitivity heatmap generated by the Ensemble method. The sensitivity heatmap of the MC-Dropout method also shows

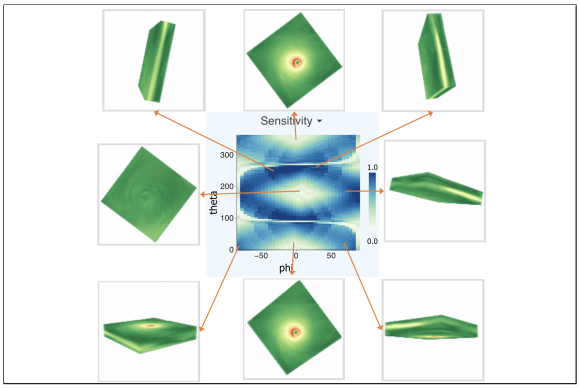


Fig. 9: Visualizing input space sensitivity for *vel* field of Isabel data. Several low and high-sensitivity viewpoints are highlighted. It is observed that the front and back views of the data produce low sensitivity, while the slanted and side views tend to incur high sensitivity.

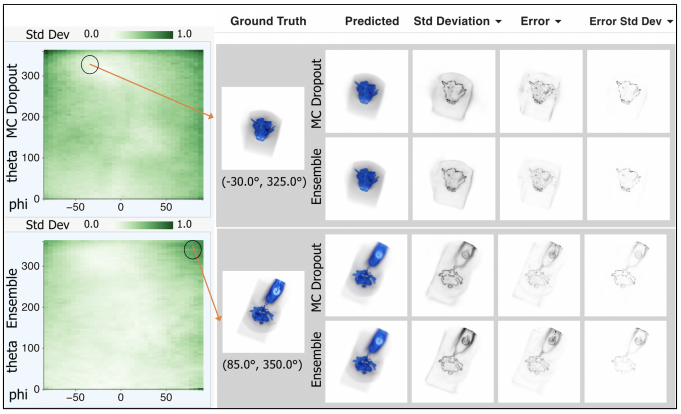


Fig. 10: Visualizing prediction uncertainty for *tev* field of Asteroid data. The top row shows a representative view with low uncertainty, and the bottom row shows an example view with high uncertainty.

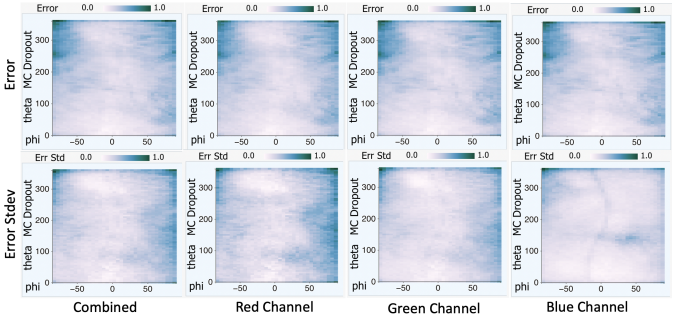


Fig. 11: Visualizing error (top row) and error standard deviation (bottom row) for asteroid data for the complete input view space. The error maps are produced by the MC-Dropout method. It is observed that the error maps are consistent across different color channels. However, the red channel produces maximum error variability while the blue channel incurs minimum error variability.

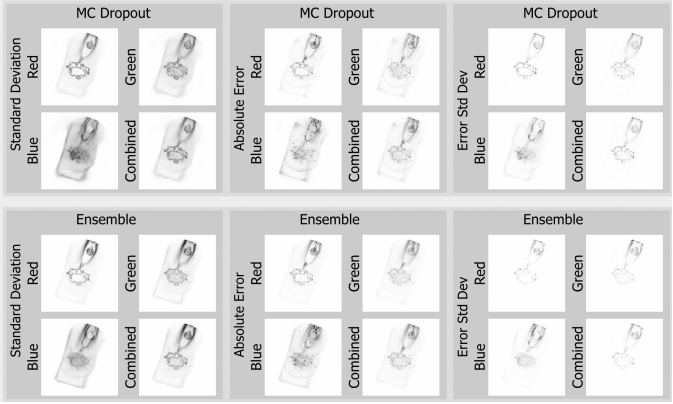


Fig. 12: Visualizing pixel-wise uncertainty, error, and error variability for RGB and combined channels for a representative view for *tev* field of Asteroid data. It is observed that the uncertainty and error vary across different color channels.

the same pattern as the Ensemble method. We depict eight representative views selected from high and low-sensitivity regions of the heatmap. The flat front and back views of the *vel* field produce lower sensitivity, while the slanted and side views produce higher sensitivity.

5.4 Visual Analysis using Asteroid Impact Data

After the Isabel data, we analyze uncertainty and error by using the Temperature (*tev*) field of Asteroid impact data with spatial resolution $300 \times 300 \times 300$. The results of prediction uncertainty are presented in Fig. 10. We observe minor differences in the heatmaps of uncertainty generated by the MC-Dropout and Ensemble method. Images for representative views with high and low uncertainty are shown on the right. The top row shows a viewpoint with low uncertainty when the data is observed from the back side of the asteroid-impacted region. However, the top view of the asteroid impact data produces high uncertainty (the bottom row in the right of Fig. 10) as the data from this view shows complex structures of the *tev* field.

Next, we visualize how the uncertainty and error vary across different color channels of the generated images. In Fig. 12, we show the channel-wise uncertainty, error, and error standard deviation images of a representative view from the asteroid data. The blue channel incurs higher uncertainty than the red and green channels. The pixel-wise error and error standard deviation patterns also vary across the three color channels. We also note that high-level uncertainty, error, and error standard deviation patterns are comparable for both MC-Dropout and Ensemble methods.

Finally, channel-wise error and error standard deviation for the entire view space for the asteroid data are provided in Fig. 11. The top row shows the error, and the bottom row shows the error variability heatmaps. While we observe that the error heatmaps are similar across different channels, the error variability heatmaps demonstrate different characteristics. Notably, the blue channel has the minimum, while the red channel incurs the maximum error standard deviation.

5.5 Implications of the Analysis Results

- Both methods produce similar uncertainty and error heatmaps, and the patterns of higher uncertainty leading to higher errors are consistent for each method. Pearson's

TABLE 1: Pearson's correlation between uncertainty and error values computed on the test set. Observations: (1) Uncertainty and error are positively correlated in both methods, and (2) uncertainty, error, and sensitivity are also positively correlated between the MC-Dropout and Ensemble methods.

	MC-Un, MC-Err	Ens-Un, Ens-Err	MC-Un, Ens-Un	MC-Err, Ens-Err	MC-Sen, Ens-Sen
Isabel	0.835	0.984	0.841	0.919	0.985
Comb	0.9227	0.773	0.878	0.992	0.974
Aster	0.819	0.963	0.863	0.913	0.959

correlation analysis confirms a strong positive correlation between uncertainty, error, and sensitivity within and across the two methods (refer to Table 1). This suggests that the MC-Dropout method can be a reliable choice for resource-constrained applications, with comparable uncertainty and sensitivity characteristics to the Ensemble method. However, it should be noted that the Ensemble method yields higher PSNR (refer to Table 2), indicating a trade-off between accuracy and computational cost.

- The study underscores the importance of providing uncertainty and sensitivity information alongside model predictions to gain experts' trust in the results. Uncertainty, error, and sensitivity are closely related, with higher uncertainty often corresponding to higher errors. In situations where ground truth is unavailable, uncertainty can serve as a valuable indicator for experts, enhancing the credibility of predicted images.
- The analysis also identifies a limitation in deep visualization models, where regions with sharp changes (e.g., edges and high gradient features) tend to exhibit higher error and uncertainty. This highlights the need for further enhancements to improve the accuracy of these visualization models, potentially through the incorporation of GAN-based training frameworks [8].

6 Parameter and Performance Study

We use a GPU server with Nvidia Quadro $P5000$ GPUs for the experimentation. The analysis is done on a MacBook Pro with an Apple M1 Pro chip with 10 CPU and 16 GPU cores and 16GB memory. All models are implemented in PyTorch [65]. The training set consists of $10,000$ randomly sampled viewpoints, and the samples are randomly shuffled to create variability among ensemble members. The evaluations are done on a test set of 512 viewpoints.

Quantitative Evaluation of Prediction Quality and Error. In Table 2, we provide a quantitative evaluation of model prediction quality and error among models with no dropout, MC-Dropout, and Ensemble methods. We perform this evaluation on the test set and report average RGB channel-wise and combined PSNR and Mean squared error (MSE) values. For the MC-Dropout method, we use 100 MC samples and dropout probability $\eta = 0.1$; for the Ensemble method, 20 ensemble members are used. We observe that for all the data sets, the PSNR and MSE for no dropout method are comparable with the MC-Dropout method. However, the MC-Dropout method has the added advantage of uncertainty information over the no-dropout method. Notably, the Ensemble method consistently produces the highest PNSR, making it the best performer.

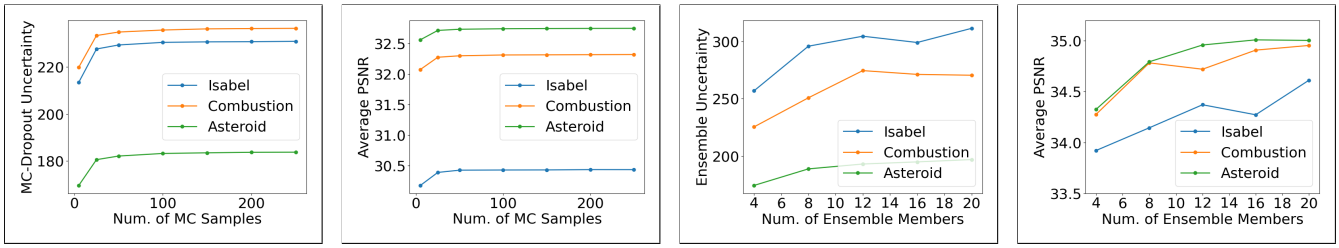
Impact of Different Number of Monte Carlo Samples. The MC-Dropout method requires sampling the model several times to produce the results. To study the impact of the sample numbers on the model prediction, we evaluate how the average prediction uncertainty and average Peak Signal-to-Noise Ratio (PSNR) change by varying the number of MC samples over the test set. In Fig. 13a and Fig. 13b, we show the results as the number of MC samples is increased up to 250. We find that 100 MC samples can produce robust estimates for all three data sets as the uncertainty and PSNR saturate around 100 MC samples.

Impact of Number of Ensemble Members. Since the number of members in the ensemble impacts the total training time, we study how many members would be sufficient to produce robust uncertainty and PSNR values. From Fig. 13c and Fig. 13d, we observe that both average uncertainty and average PSNR, computed over the test set, tend to saturate when 20 members are used.

Impact of Different Dropout Probabilities. In Fig. 14a, we highlight how changing the dropout probability (η) impacts the overall average PSNR values in the test set for the predicted images. For each observation, 100 MC samples are used. It is found that the PSNR value decreases slowly as the dropout probability is increased. We expect this trend to continue for higher dropout probabilities. This experiment (analyses shown in Table 2) and the visualization results presented in the above sections show that a small dropout probability can adequately capture the model uncertainty without compromising the prediction quality. Hence, for all our analyses, we use the fixed dropout probability $\eta = 0.1$.

Visual Comparison of Predicted Images for MC-Dropout and Ensemble Methods Under Different Parameter Configuration. Fig. 14b allows a visual comparison of the predicted (mean) images from the two methods for a specific test viewpoint for the *mixfrac* variable of Combustion data. The top row shows the predicted images when the dropout probabilities are varied from 0.5 to 0.1. The bottom row shows images for that viewpoint when the number of ensemble members increases from 4 to 20. Even though the images look visually similar, there are subtle differences. The PSNR values (provided at the top of each image) convey the quality of each parameter configuration. The ground truth for this test view is shown on the left side at the center. Expectedly, the PSNR improves as the dropout probability declines. Similarly, the PSNR improves for the Ensemble method as the number of ensemble members increases.

Computational and Storage Performance Evaluation. Table 3 presents the MC-Dropout and Ensemble model's computational cost and storage needs. Training all the ensemble members takes significantly longer than training the MC-Dropout model. The storage for all the parameters from all the models is also high. Our experiments show that the inference time is shorter for the Ensemble model than for MC-Dropout. However, this minor lag is not a limiting factor for adopting the MC-Dropout method for practical applications. The flexibility of uncertainty estimation using a single model can easily outweigh such minimal performance differences with the Ensemble method. We perform this evaluation systematically, generating 100 MC samples for each view. We train 20 ensemble members and conduct inference using all the 20 members on the test set.



(a) #MC samp. vs. Uncertainty (b) #MC samp. vs. PSNR (c) #Ens vs. Uncertainty (d) #Ens vs. PSNR

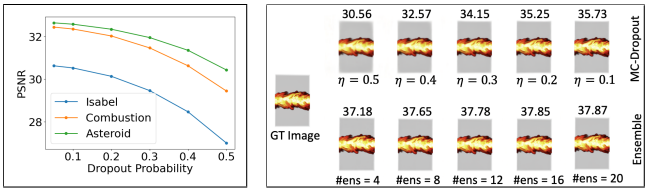
Fig. 13: Fig. 13a and Fig. 13b show how the prediction uncertainty and PSNR are impacted by the number of Monte Carlo samples used. It is observed that the estimated uncertainty and PSNR are saturated when around 100 Monte Carlo samples are used. Fig. 13c and Fig. 13d show the change in estimated uncertainty and PSNR when the number of ensemble members is varied. We observe that ∼20 ensemble members tend to produce consistent prediction uncertainty and PSNR.

TABLE 2: Comparison of channel-wise and average prediction quality for no dropout, MC-Dropout, and Ensemble method for three different data sets. The ensemble method produces the highest quality of predictions among the three test cases, while no dropout and MC-Dropout methods produce comparable quality when dropout probability $\eta = 0.1$ is used.

		Prediction Quality and Error Evaluation							
		Peak Signal to Noise Ratio (PSNR)				Mean Squared Error (MSE)			
		Red Channel	Green Channel	Blue Channel	Average	Red Channel	Green Channel	Blue Channel	Average
Isabel	No dropout	30.48847	31.23577	31.18883	30.97102	0.00125	0.00036	0.00068	0.00077
	MC-Dropout (100 MC samples)	29.89097	30.4523	30.92661	30.4233	0.00142	0.00044	0.00071	0.00085
	Ensemble (20 mems)	34.23113	34.37377	35.18207	34.59566	0.00058	0.00019	0.00031	0.00036
Combustion	No dropout	33.28172	32.80587	32.04885	32.71214	0.00039	0.00061	0.00071	0.00057
	MC-Dropout (100 MC samples)	32.57521	32.42585	31.92816	32.30974	0.00045	0.00061	0.00068	0.00058
	Ensemble (20 mems)	35.11684	35.45715	34.27021	34.94806	0.00025	0.00031	0.0004	0.00032
Asteroid	No dropout	32.13364	34.15641	32.04537	32.77847	0.00068	0.00034	0.00017	0.0004
	MC-Dropout (100 MC samples)	32.03944	34.01588	32.16493	32.74008	0.00068	0.00035	0.00017	0.0004
	Ensemble (20 mems)	34.58787	36.60189	33.78471	34.99149	0.00038	0.00019	0.00012	0.00023

TABLE 3: Training and inference timings and storage requirements for MC-Dropout and Ensemble method.

	MC-Dropout			Ensemble		
	Training time (Hrs)	Avg. inference time per view using 100 MC samples (Secs.)	Storage (MB)	Training time (Hrs)	Avg. inference time per view using 20 Ensemble members (Secs.)	Storage (MB)
Combustion	17.74	0.0771	72	235.42	0.0324	1440
Isabel	17.6	0.0768	72	239.09	0.0313	1440
Asteroid	17.86	0.0852	72	164.08	0.0247	1440



(a) Dropout prob. vs. PSNR (b) Comparison of MC-Dropout vs. Ensemble.

Fig. 14: Fig. 14a depicts how changing the dropout probability impacts the quality of PSNR. Fig. 14b compares mean images by MC-Dropout and Ensemble method.

7 Discussion

Ensemble methods have significantly more computation and memory costs (see Table 3), limiting their usage and deployment in many resource-constrained real-world problems. One can reduce the number of models to reduce computational and memory costs; however, such reduction is often non-trivial. Along this line, Distillation [66] is an approach where the ensemble is reduced to one model by teaching a single network to represent the knowledge of a group of neural networks. In more recent times, similar techniques for estimating uncertainty based on single models have emerged [67]. It is important to emphasize that ensemble learning has become the prevailing approach in contemporary AI systems [68]. Consequently, it is widely adopted as a prominent tool for uncertainty estimation and often serves as a benchmark for our analysis. The evolution of computational infrastructure and the proliferation of data resources have facilitated the emergence of applications in which multiple models collaborate harmoniously to enhance the overall product experience [69]. These models exhibit characteristics that make them amenable to calibration, retraining, maintenance, and optimization to align with specific complementary objectives or cater to distinct domains [70]. It is worth noting that, in such applications, the conventional practice is to train multiple models rather than a single model that can provide predictive uncertainty.

Training a deep ensemble and generating predictions, given a reasonable time budget, requires parallel GPU-based facilities. This potential inconvenience of the deep ensemble may motivate one to find a practical alternative, the MC-Dropout method. Our studies show that although Ensemble methods produce superior prediction quality, the MC-Dropout method results are at par with the Ensemble methods for uncertainty estimation. This parity is achieved without the additional overhead of retraining or an expensive memory footprint. Furthermore, the theoretical connec-

tion of MC-Dropout with the deep Gaussian processes [4] makes it an ideal candidate for deep uncertainty estimation.

8 Conclusion

This work presents a detailed comparative analysis of two deep uncertainty estimation techniques: (1) MC-Dropout and (2) Ensemble method. We propose a new interactive visual analytics tool to compare both methods' prediction uncertainty, error, and model sensitivity. The uncertainty estimation capabilities enrich the usability and credibility of the models, making them more interpretable. By performing a detailed evaluation of these methods, we reveal insights into such techniques when applied to deep volume-rendered image-synthesizing models. In the future, we plan to study uncertainty for other salient rendering parameters and explore other deep uncertainty estimation techniques.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[2] C. Wang and J. Han, "Dl4scivis: A state-of-the-art survey on deep learning for scientific visualization," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2022.

[3] G.-P. Bonneau, H.-C. Hege, C. R. Johnson, M. M. Oliveira, K. Potter, P. Rheingans, and T. Schultz, *Overview and State-of-the-Art of Uncertainty Visualization*. London: Springer London, 2014, pp. 3–27.

[4] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proc. of ICML*, 2016, pp. 1050–1059.

[5] F. K. Gustafsson, M. Danelljan, and T. B. Schon, "Evaluating scalable bayesian deep learning methods for robust computer vision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 318–319.

[6] W. H. Beluch, T. Genewein, A. Nürnberger, and J. M. Köhler, "The power of ensembles for active learning in image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9368–9377.

[7] A. Damianou and N. D. Lawrence, "Deep Gaussian processes," in *Proc. of AISTATS*, 2013, pp. 207–215.

[8] W. He, J. Wang, H. Guo, K.-C. Wang, H.-W. Shen, M. Raj, Y. S. G. Nashed, and T. Peterka, "Insitunet: Deep image synthesis for parameter space exploration of ensemble simulations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 23–33, 2020.

[9] F. Hong, C. Liu, and X. Yuan, "Dnn-volvis: Interactive volume visualization supported by deep neural network," in *2019 IEEE Pacific Visualization Symposium (PacificVis)*, 2019, pp. 282–291.

[10] M. Berger, J. Li, and J. A. Levine, "A generative model for volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 4, pp. 1636–1650, 2019.

[11] Y. Lu, K. Jiang, J. A. Levine, and M. Berger, "Compressive neural representations of volumetric scalar fields," *Computer Graphics Forum*, vol. 40, no. 3, pp. 135–146, 2021.

[12] S. Weiss, P. Hermüller, and R. Westermann, "Fast neural representations for direct volume rendering," *Computer Graphics Forum*, vol. 41, no. 6, pp. 196–211, 2022.

[13] S. Weiss, M. Chu, N. Thuerey, and R. Westermann, "Volumetric isosurface rendering with deep learning-based super-resolution," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 6, pp. 3064–3078, 2021.

[14] S. Weiss, M. IşIk, J. Thies, and R. Westermann, "Learning adaptive sampling and reconstruction for volume visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 7, pp. 2654–2667, 2022.

[15] J. Han and C. Wang, "SSR-TVD: Spatial super-resolution for time-varying data analysis and visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 6, pp. 2445–2456, 2022.

[16] ——, "Tsr-tvd: Temporal super-resolution for time-varying data analysis and visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 205–215, 2020.

[17] S. W. Wurster, H. Guo, H.-W. Shen, T. Peterka, and J. Xu, "Deep hierarchical super resolution for scientific data," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–14, 2022.

[18] J. Han, H. Zheng, D. Z. Chen, and C. Wang, "Stnet: An end-to-end generative framework for synthesizing spatiotemporal super-resolution volumes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 270–280, 2022.

[19] J. Shen, H. Li, J. Xu, A. Biswas, and H.-W. Shen, "Idlat: An importance-driven latent generation method for scientific data," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–11, 2022.

[20] N. Shi, J. Xu, S. W. Wurster, H. Guo, J. Woodring, L. P. Van Roekel, and H.-W. Shen, "Gnn-surrogate: A hierarchical and adaptive graph neural network for parameter space exploration of unstructured-mesh ocean simulations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 6, pp. 2301–2313, 2022.

[21] N. Shi, J. Xu, H. Li, H. Guo, J. Woodring, and H.-W. Shen, "Vdl-surrogate: A view-dependent latent-based model for parameter space exploration of ensemble simulations," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–11, 2022.

[22] J. Han, H. Zheng, Y. Xing, D. Z. Chen, and C. Wang, "V2v: A deep learning approach to variable-to-variable selection and translation for multivariate time-varying data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 1290–1300, 2021.

[23] J. Han and C. Wang, "Surfnet: Learning surface representations via graph convolutional network," *Computer Graphics Forum*, vol. 41, no. 3, pp. 109–120, 2022.

[24] A. T. Pang, C. M. Wittenbrink, S. K. Lodha *et al.*, "Approaches to uncertainty visualization," *The Visual Computer*, vol. 13, no. 8, pp. 370–390, 1997.

[25] K. Potter, J. Krüger, and C. Johnson, "Towards the visualization of multi-dimensional stochastic distribution data," in *Proceedings of The International Conference on Computer Graphics and Visualization (IADIS) 2008*, 2008.

[26] K. Potter, P. Rosen, and C. R. Johnson, *Uncertainty Quantification in Scientific Computing: 10th IFIP WG 2.5 Working Conference, WoCoUQ 2011, Boulder, CO, USA, August 1-4, 2011, Revised Selected Papers*, 2012, ch. From Quantification to Visualization: A Taxonomy of Uncertainty Visualization Approaches, pp. 226–249.

[27] K. Brodlie, R. Allendes Osorio, and A. Lopes, *Expanding the Frontiers of Visual Analytics and Visualization*. Springer London, 2012, ch. A Review of Uncertainty in Data Visualization, pp. 81–109.

[28] S. Liu, J. Levine, P. Bremer, and V. Pascucci, "Gaussian mixture model based volume visualization," in *2012 IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, 2012, pp. 73–77.

[29] T. M. Athawale, B. Ma, E. Sakhaee, C. R. Johnson, and A. Entezari, "Direct volume rendering for nonparametric models of uncertainty," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 1797–1807, Feb 2021.

[30] K. Pöthkow and H.-C. Hege, "Positional uncertainty of isocontours: Condition analysis and probabilistic measures," *IEEE Trans. on Vis. and Comp. Graphics*, vol. 17, pp. 1393–1406, 2011.

[31] K. Pöthkow, B. Weber, and H.-C. Hege, "Probabilistic marching cubes," in *Proceedings of the 13th Eurographics / IEEE - VGTC Conference on Visualization*, ser. EuroVis'11, 2011, pp. 931–940.

[32] R. T. Whitaker, M. Mirzargar, and R. M. Kirby, "Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2713–2722, 2013.

[33] T. M. Athawale, C. R. Johnson, S. Sane, and D. Pugmire, "Fiber uncertainty visualization for bivariate data with parametric and nonparametric noise models," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–11, 2022.

[34] C. Gillmann, P. Arbelaez, J. T. Hernandez, H. Hagen, and T. Wischgoll, "An uncertainty-aware visual system for image pre-processing," *Journal of Imaging*, vol. 4, no. 9, 2018. [Online]. Available: https://www.mdpi.com/2313-433X/4/9/109

[35] C. Gillmann, D. Saur, T. Wischgoll, and G. Scheuermann, "Uncertainty-aware visualization in medical imaging - a survey," *Computer Graphics Forum*, vol. 40, no. 3, pp. 665–689, 2021.

[36] A. Kamal, P. Dhakal, A. Y. Javaid, V. K. Devabhaktuni, D. Kaur, J. Zaientz, and R. Marinier, "Recent advances and challenges in

uncertainty visualization: a survey," *Journal of Visualization*, vol. 24, no. 5, pp. 861–890, 2021.

[37] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.

[38] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," *arXiv preprint arXiv:1505.05424*, 2015.

[39] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in neural information processing systems*, 2017, pp. 6402–6413.

[40] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, "Deep double descent: where bigger models and more data hurt*," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2021, no. 12, dec 2021.

[41] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[42] Y. You, J. Hseu, C. Ying, J. Demmel, K. Keutzer, and C.-J. Hsieh, "Large-batch training for lstm and beyond," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2019.

[43] D. Choi, C. J. Shallue, Z. Nado, J. Lee, C. J. Maddison, and G. E. Dahl, "On empirical comparisons of optimizers for deep learning," *CoRR*, vol. abs/1910.05446, 2019. [Online]. Available: http://arxiv.org/abs/1910.05446

[44] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *Advances in Neural Information Processing Systems*, 2018, pp. 3179–3189.

[45] M. Raghu, K. Blumer, R. Sayres, Z. Obermeyer, B. Kleinberg, S. Mullainathan, and J. Kleinberg, "Direct uncertainty prediction for medical second opinions," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5281–5290.

[46] Y. Ma, N. J. Foti, and E. B. Fox, "Stochastic gradient MCMC methods for hidden M arkov models," in *Proc. of ICML*, 2017, pp. 2265–2274.

[47] A. Hasanzadeh, E. Hajiramezanali, K. Narayanan, N. Duffield, M. Zhou, and X. Qian, "Semi-implicit graph variational auto-encoders," in *Proc. of NIPS*, 2019, pp. 10 712–10 723.

[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[49] X. Ma, "Apollo: An adaptive parameter-wised diagonal quasi-newton method for nonconvex stochastic optimization," 2021. [Online]. Available: https://openreview.net/forum?id=5B8YAz6W3eX

[50] M. S. Ayhan and P. Berens, "Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks," in *Medical Imaging with Deep Learning Conference*, 2018.

[51] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren, "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks," *Neurocomputing*, vol. 338, pp. 34–45, 2019.

[52] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.

[53] M. Leutbecher and T. N. Palmer, "Ensemble forecasting," *Journal of computational physics*, vol. 227, no. 7, pp. 3515–3539, 2008.

[54] W. S. Parker, "Ensemble modeling, uncertainty and robust predictions," *Wiley Interdisciplinary Reviews: Climate Change*, vol. 4, no. 3, pp. 213–223, 2013.

[55] A. Renda, M. Barsacchi, A. Bechini, and F. Marcelloni, "Comparing ensemble strategies for deep learning: An application to facial expression recognition," *Expert Systems with Applications*, vol. 136, pp. 1–11, 2019.

[56] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," *arXiv preprint arXiv:1906.02530*, 2019.

[57] A. Vyas, N. Jammalamadaka, X. Zhu, D. Das, B. Kaul, and T. L. Willke, "Out-of-distribution detection using an ensemble of self supervised leave-out classifiers," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 550–564.

[58] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[59] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Proc. of Neurips*, 2015, pp. 2575–2583.

[60] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

[61] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[62] Y. Gal, J. Hron, and A. Kendall, "Concrete dropout," in *In Proc. of Neurips*, 2017, pp. 3581–3590.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[64] S. Park and N. Kwak, "Analysis on the dropout effect in convolutional neural networks," in *Asian Conference on Computer Vision*, 2016.

[65] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. of NIPS*, 2019, pp. 8024–8035.

[66] S. Reich, D. Mueller, and N. Andrews, "Ensemble distillation for structured prediction: Calibrated, accurate, fast - choose three," *CoRR*, vol. abs/2010.06721, 2020. [Online]. Available: https://arxiv.org/abs/2010.06721

[67] J. Thiagarajan, R. Anirudh, V. S. Narayanaswamy, and T. Bremer, "Single model uncertainty estimation via stochastic data centering," in *Advances in Neural Information Processing Systems*, vol. 35. Curran Associates, Inc., 2022, pp. 8662–8674.

[68] Z. Hu, J. Zhang, H. Wang, S. Liu, and S. Liang, "Leveraging relational graph neural network for transductive model ensemble," in *Proc. of KDD*, 2023, p. 775–787.

[69] OpenAI, "Gpt-4 technical report," 2023.

[70] A. C. et. al., "Palm: Scaling language modeling with pathways," 2022.

Soumya Dutta is an Assistant Professor in the Computer Science department at the Indian Institute of Technology Kanpur (IITK). He received his Ph.D. degree in Computer Science from the Ohio State University in May 2018. His research interests are Machine Learning for Visual Computing, Uncertainty Visualization, xAI, Big Data Analytics, and HPC. Contact him at soumyad@cse.iitk.ac.in.

Faheem Nizar is a Senior Undergraduate student at the Indian Institute of Technology Kanpur (IITK). He is currently majoring in Electrical Engineering. His research interests are in Machine Learning, Deep Learning, Data Analytics and Visualisation. He has also worked with Long-Distance wireless communication networks. Contact him at nfaheem20@iitk.ac.in.

Ahmad Amaan is a Senior Undergraduate student at the Indian Institute of Technology Kanpur (IITK). He is currently majoring in Mechanical Engineering with minors in computer systems and machine learning. His research interests include system design, image processing, and machine learning for visualization. Contact him at aamaan20@iitk.ac.in.

Ayan Acharya holds the position of Senior Software Engineer at Linked Inc. in Sunnyvale, USA. His research interests primarily revolve around deep learning, large-scale optimization, and Bayesian non-parametrics. He completed his doctoral studies at the University of Texas at Austin, where he specialized in the development of efficient inference algorithms for graphical models. For further inquiries or contact, you can reach him at aacharya@utexas.edu.