

Uncertainty-Aware Deep Neural Representations for Visual Analysis of Vector Field Data

Atul Kumar, Siddharth Garg, and Soumya Dutta

Abstract—The widespread use of Deep Neural Networks (DNNs) has recently resulted in their application to challenging scientific visualization tasks. While advanced DNNs demonstrate impressive generalization abilities, understanding factors like prediction quality, confidence, robustness, and uncertainty is crucial. These insights aid application scientists in making informed decisions. However, DNNs lack inherent mechanisms to measure prediction uncertainty, prompting the creation of distinct frameworks for constructing robust uncertainty-aware models tailored to various visualization tasks. In this work, we develop uncertainty-aware implicit neural representations to model steady-state vector fields effectively. We comprehensively evaluate the efficacy of two principled deep uncertainty estimation techniques: (1) Deep Ensemble and (2) Monte Carlo Dropout, aimed at enabling uncertainty-informed visual analysis of features within steady vector field data. Our detailed exploration using several vector data sets indicate that uncertainty-aware models generate informative visualization results of vector field features. Furthermore, incorporating prediction uncertainty improves the resilience and interpretability of our DNN model, rendering it applicable for the analysis of non-trivial vector field data sets.

Index Terms—Implicit Neural Network, Uncertainty, Monte Carlo Dropout, Deep Ensemble, Vector Field, Visualization, Deep Learning.

1 INTRODUCTION

The indisputable success of deep neural networks (DNNs) [44] has resulted in numerous applications of it in the scientific visualization domain [73]. Analysis of intricate vector fields using DNNs has shown promising results with applications such as generating super-resolution flow fields [26, 31], reconstructing flow fields from streamlines [23, 28], flow map reconstruction [63], and predicting flow lines [33]. While these DNN-based approaches produce state-of-the-art results, learning a direct neural representation of the vector data is yet to be explored. Furthermore, the existing models do not quantify what it does not know or how confidently the predictions are generated. Such missing knowledge, if estimated and conveyed to the experts, can significantly help them to make informed decisions about their data [9, 19]. Consequently, DNN-based flow data visual analytics will become more trustworthy when they can assess their prediction uncertainty. However, the literature survey reveals that a direct vector data modeling approach augmented with uncertainty estimation capability using DNNs is still missing – a gap that this work attempts to bridge.

Among various neural architectures used to analyze flow data, we study the efficacy of implicit neural representations (INRs) for directly learning vector fields, i.e., a INR will predict the vector components at any queried location in the spatial domain. The choice of INR over other neural architectures is motivated by the recent success of INRs in producing state-of-the-art results for scientific data [32, 48, 63, 69]. Since DNNs inherently do not provide prediction uncertainty, we employ deep uncertainty quantification techniques to obtain uncertainty estimates along with the predicted vectors from our INR model so that the downstream flow field analysis tasks can effectively leverage such information. Hence, we further focus on designing uncertainty-informed flow data visualization techniques to intuitively and effectively communicate the prediction uncertainty to domain experts.

Various factors specific can influence the choice of deep uncertainty estimation methods. While the deep learning community has developed several uncertainty quantification methods, our focus is on selecting methods that can be easily integrated into visualization models with

minimal architecture modifications, facilitating smooth adoption of such methods for uncertainty-aware visual analysis. Literature survey indicates that Deep Ensembles often excel in producing accurate predictions and estimating uncertainty by leveraging the variation among ensemble member predictions [5, 27, 39]. However, Deep Ensembles demand prohibitively excessive training time and resource utilization since multiple ensemble members need to be trained. In contrast, single-model-based uncertainty estimation methods are computationally cheap and are gaining importance in the research community [1, 19, 70]. Among these methods, Monte Carlo Dropout (MCDropout) stands out as a suitable choice for our purposes since it only requires the addition of dropout layers to an existing model and theoretically equivalent to approximate inferencing in deep Gaussian processes [14, 19]. Therefore, in this work, we compare and contrast the characteristics of uncertainty estimates generated by Deep Ensemble and MCDropout methods for enabling uncertainty-aware visual analytics of steady-state flow fields.

We analyze the effectiveness and advantages of uncertainty-aware implicit neural representations (INRs) of steady-state flow fields coupled with two deep uncertainty estimation techniques: (1) Deep Ensemble and (2) MCDropout method. We conduct a comprehensive assessment of the predictive accuracy of such uncertainty-aware INRs by (a) evaluating the quality of reconstructed vector fields when the INR learns the vector representation directly and (b) assessing the effectiveness in analyzing and visualizing flow features through streamline and critical point visualizations. For each task, we devise uncertainty visualization methods to convey uncertainty information to the experts. We compare and contrast the characteristics of uncertainties derived from these methods and illustrate how such information can assist domain scientists in making informed decisions and developing confidence in the model outputs. We validate our techniques using several 2D and 3D vector field data sets to showcase the efficacy of our approach in flow field analysis and visualization. Hence, our contributions are twofold:

• Atul Kumar and Soumya Dutta are with the department of Computer Science and Engineering, and Siddharth Garg is with the Electrical Engineering department at IIT Kanpur, India. Soumya Dutta is the corresponding author. Contact him at soumyad@cse.iitk.ac.in.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx

1. We propose developing uncertainty-aware INRs for vector field data to enhance the robust analysis of intricate flow features, emphasizing the significance of visualizing prediction uncertainty to improve how domain scientists interpret model predictions.
2. We extensively compare and contrast between two principled deep uncertainty estimation techniques: (1) Deep Ensemble, and (2) MCDropout, demonstrating their suitability for conducting uncertainty-informed flow feature analysis and visualization.

2 RESEARCH BACKGROUND AND UNCERTAINTY ESTIMATION IN DEEP NEURAL NETWORKS

2.1 Deep Learning in Scientific Data Visualization

Deep learning has found diverse applications in scientific visualization. Lu et al. [48] and Weiss et al. [76] have introduced methods aimed at generating concise neural representations of scientific data. Hong et al. [38], He et al. [36], and Berger et al. [7] have investigated the visualization of scalar field data, while Weiss et al. have explored isosurface visualization [75] and volume visualization [77]. Spatiotemporal super-resolution volume generation has emerged as another research focus [20, 29, 30, 79]. Novel models for domain knowledge-aware latent space generation have been proposed [65]. Moreover, DNNs have been employed as substitutes for generating visualizations and exploring parameter spaces in ensemble data [36, 66, 67]. Analysis of flow field features using CNNs was proposed in [80]. Reconstruction of vector data from 3D streamlines was developed by Han et al. [28]. More recently, Berenjkoub et al. highlighted how vortex boundaries could be extracted using CNNs [6]. In a recent work, Dutta et al. [15] demonstrated how visualizing prediction uncertainty can provide insights about the model’s robustness for view synthesizing tasks. For a comprehensive overview of flow data visualization using deep learning methods, please refer to [46] and for a broader set of applications of deep learning applications in scientific visualization please refer to [73].

2.2 Uncertainty Visualization

Pang et al. provide one of the earliest summaries of uncertainty visualization techniques [53]. Potter et al. focus on visualizing spatial probability distributions, preceded by a taxonomy of uncertainty visualization methods [59, 60]. Brodlie et al. [10] introduce visualization techniques enhanced with tools for uncertainty estimation. Liu et al. employ flickering as a method to depict uncertainty in volume data [47], while Athawale et al. delve deeper into uncertainty visualization in volume rendering using non-parametric models [2]. Uncertainty visualization techniques tailored for isocontouring methods have received significant attention in research. Pöthkow et al. devise a method to compute the level crossing probability between adjacent points, which is further refined to calculate the probability for each cell [57, 58]. Whitaker et al. [78] explore uncertainty visualization in ensembles of contours. Streamline uncertainty in ensemble field [17] and pathline uncertainty in time-varying field [11] is also explored. Otto et al. [51] study uncertainty in 2D vector fields. Bonneau et al. conduct a comprehensive survey of various uncertainty visualization techniques [9]. Recently, Gillmann et al. provide a summary of uncertainty visualization methods geared towards image processing [21].

2.3 Uncertainty Estimation in Deep Neural Networks

The uncertainty [8, 43] of a DNN can be categorized into two broad types – *data or aleatoric uncertainty* and *model or epistemic uncertainty*. Data uncertainty is attributed to errors and noise during data acquisition. Model (epistemic) uncertainty can arise for different reasons. Firstly, the DNNs produce a compressed representation of large-scale data. Such compression often results in prediction error and associated uncertainty. Secondly, DNNs are often fine-tuned carefully using learning rate variation, regularization, etc. Different decisions for such configurations lead to different learned model representations, and analysis using such models can lead to uncertainties. The data uncertainty can be addressed by improving data collection method and by improving the data quality. In contrast, the epistemic uncertainty is inherent to the model and needs to be studied in detail.

2.3.1 Techniques for Modeling Uncertainty in DNNs

Deterministic methods. Deterministic models can be equipped with uncertainty estimation capabilities by explicitly training a network to quantify uncertainties [64].

Bayesian Methods. Bayesian neural networks [22] employ prior distributions on the model parameters of DNNs to compute epistemic uncertainty [43]. Training such networks require stochastic gradient MCMC [49], and variational inference [34] methods.

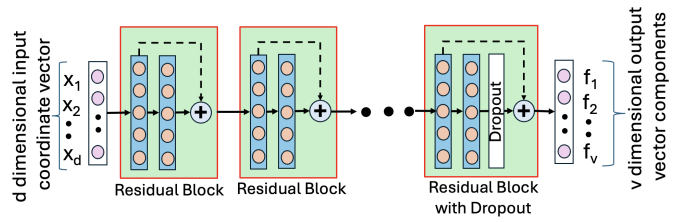


Fig. 1: The schematic of the MCDropout-enabled INR model which uses a residual block-based MLP architecture. A dropout layer is added at the last residual block to generate uncertainty estimates during inference time. The INR architecture for the Ensemble method is identical to this, except there is no dropout layer.

Test-time Augmentation. Test time augmentation methods perform data augmentation during the inference time and then estimate the uncertainty from the variability in the predicted results [3, 74].

Deep Evidential Regression. In deep evidential regression, the network learns parameters as well as hyperparameters to the corresponding evidential distributions [1]. Then finally, the evidential distributions are used to estimate model uncertainty.

Uncertainty via Stochastic Data Centering. Here, the authors show that when an ensemble of DNNs is trained with data sets that are shifted by a constant bias, model uncertainty becomes the variability across such ensemble members’ predictions [70].

2.3.2 Ensemble Method

In ensemble-based methods, several models work harmoniously to produce predictions that are of superior quality compared to the predictions from any individual model [62]. By doing this, ensemble methods improve the generalization error, and by estimating the variability among ensemble member predictions, the model uncertainty can be estimated robustly. Following this principle, Lakshminarayanan et al. propose Deep Ensembles [43] for DNNs. Conceptually, Deep Ensemble learning can be shown as an approximation of *Bayesian averaging* [61]. In Bayesian averaging, the final model prediction is formulated as: $\text{prediction} = \int \mathcal{P}_w(x) \pi(w|\mathcal{D})$. Here $\mathcal{P}_w(x)$ is the prediction associated with sample x and $\pi(w|\mathcal{D})$ represents the posterior probability distribution with \mathcal{D} being the training data. In reality, estimation of this integral is extremely difficult, and it is also found that to estimate this integral sufficiently accurately, exploration of all the modes of $\pi(w|\mathcal{D})$ is not required. This observation imply that non-weighted averaged predictions generated by a set of ensemble members can be considered an approximation of the above expression. Note that shuffling of the training data and a random parameter initialization during the training process introduces a good variety in each learned ensemble member to predict the uncertainty robustly, an approach followed in this work to generate Deep Ensembles. Subsequent research works on Deep Ensembles [5, 27, 52, 71] show that ensemble methods often outperform other uncertainty estimation techniques and are more immune to changes in data distribution.

2.3.3 MCDropout Method

Dropout [19, 37] is a regularization technique that prevents models from overfitting during training and is achieved by randomly masking a subset of the weights during training. However, Gal et al. [19] discovered that adding dropout in a DNN with arbitrary depth and non-linear activations makes it theoretically equivalent to an approximate Bayesian inference in deep Gaussian processes. Then activating dropout at test time is equivalent to sampling from the Bayesian posterior distribution, $p(\mathbf{W} | \mathbf{X}, \mathbf{Y})$, where \mathbf{W} denotes the model’s weights, \mathbf{X} is the training data and \mathbf{Y} is the target output. Finally, the mean of these sampled predictions is considered as the expected model output, and by measuring the variance in these sampled predictions, model uncertainty can be estimated [19]. Such probabilistic predictions are derived by collecting Monte Carlo (MC) samples from the dropout-enabled trained model, known as Monte Carlo Dropout (MCDropout) method, by performing multiple forward passes during inference.

3 UNCERTAINTY-AWARE NEURAL REPRESENTATION FOR VECTOR FIELD DATA

3.1 Implicit Neural Representation

Implicit neural representations with periodic activation functions have been identified as a promising solution for learning representations of coordinate-based data sets, where the mapping from any input coordinate in the data domain to the corresponding output quantity values is learned. Sitzmann et al. [68] in their work depicted that a feed-forward neural network with sinusoidal activation function, termed as SIREN (sinusoidal representation network), can be used to build such INRs [68]. Several variations of this SIREN have recently been used to solve many challenging problems in the scientific data visualization domain and obtain state-of-the-art results [32, 48, 63, 69]. The success of these recent research efforts has prompted us to build our uncertainty-aware model using SIRENs as the base architecture. Besides analyzing the uncertainty estimates produced by the Deep Ensemble and MCDropout methods using a SIREN-based model, we study the accuracy such a network can achieve in representing intricate steady vector fields.

3.2 Model Architecture

Using an implicit neural network, we aim to learn the function that represents the mapping from the input data coordinate domain to the corresponding vector value space. To achieve this, we build our base model as a multilayer perceptron consisting of d neurons as inputs, followed by l hidden layers and an output layer containing v neurons. To enrich the model’s learning capability and train a deep network in a stable fashion, we enhance the base SIREN architecture by incorporating residual blocks and skip connections [35] as was suggested in [48]. Our model learns a function that takes a d dimensional coordinate vector as input and predicts a v dimensional vector output where v is the number of components in the vector field data. Essentially, the network learns a function $\mathcal{F}(\theta) : \mathbb{R}^d \mapsto \mathbb{R}^v$, where θ represents the parameters of the neural network. When the size of the flow data is large, then using a single DNN to learn all the vector components jointly also allows us to produce a compact neural representation of the flow data set since the model parameters are shared across the vector components. Hence, the model size will be sufficiently smaller than the raw flow data set.

3.3 Uncertainty Quantification Using MCDropout Method

The model architecture used for the MCDropout method is shown in Fig. 1. We observe that a post-activation dropout layer is added at the last residual block to build a dropout-enabled model that can be used conveniently during inference to estimate prediction uncertainty. The addition of the dropout layer also helps in regularization during training. Theoretically, if one wants to simulate a fully Bayesian network, a dropout layer must be added at each residual block [41]. However, such a large number of dropout layers often acts as a strong regularizer and hinders the learning process, resulting in lower accuracy during test time [41]. We observe similar behavior when the dropout layer is added to every residual block. Hence, following Kendall et al.’s suggestion [41], we build a model by adding dropout at the deepest residual block, giving us robust uncertainty estimates and a high-quality vector field reconstruction. The impact of using different numbers of dropout layers has been studied and presented later in Table 7.

As outlined in Section 2.3.3, inference involves generating a set of Monte Carlo samples through multiple forward passes of the trained model when dropout is enabled. In our approach, we produce m instances of the vector field and subsequently calculate the average vector field, serving as the predicted vector field. The grid-pointwise standard deviation, computed using vector components estimated by m forward passes, denotes the prediction uncertainty. We separately compute standard deviation for each vector component and then add the standard deviation values to obtain the final prediction uncertainty.

3.4 Uncertainty Quantification Using Ensemble Method

A schematic of our INR architecture is presented in Fig. 1. To build a Deep Ensemble [43] of INRs, we use this model architecture without the dropout layer. To produce an ensemble model comprising m

Table 1: Description of data sets used in the experimentation.

Data set	Dimensionality	Spatial Resolution
Heated Cylinder (T=750)	2D	150 × 450
Heated Cylinder (T=1500)	2D	150 × 450
Fluid (T=750)	2D	512 × 512
Hurricane Isabel (T=25)	3D	250 × 250 × 50
Tornado	3D	128 × 128 × 128
Turbine	3D	151 × 71 × 56
Tangaroa (T=150)	3D	300 × 180 × 120

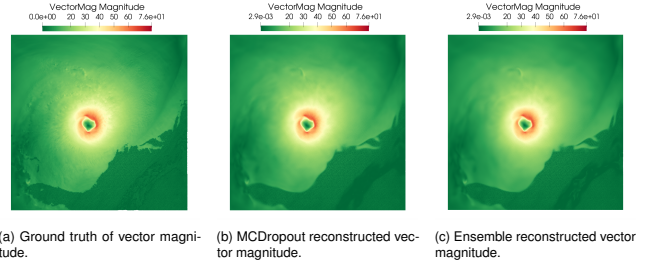


Fig. 2: Volume visualization of the magnitude of reconstructed vector fields using MCDropout and Ensemble methods for Hurricane Isabel data set. The ground truth is shown in Fig. 2a. We observe that both methods produce visually comparable results.

members, each member is trained separately, utilizing samples from the entire vector field data. During every ensemble creation, the training data is randomly shuffled for each member so that variability is introduced to each member. After the ensemble members’ training is finished, each member generates a realization of the entire vector field by inferring vector values at the grid locations. Then, the averaged vector field represents the expected vector field, and the standard deviation computed at each grid point from the predicted vector values indicates the epistemic uncertainty of the ensemble model. We separately compute standard deviation for each vector component and then add the values to quantify the final uncertainty.

Loss Function and Hyperparameters. For both MCDropout and Ensemble methods, we utilize 14 residual blocks for 3D and 10 for 2D vector field data sets. Each hidden layer comprises 120 neurons for 3D and 100 neurons for 2D data sets. The training uses conventional mean squared error loss (\mathcal{L}_{mse}). In our work, we focus on using a consistent network architecture and hyperparameter combination for both uncertainty estimation methods to produce comparable results across multiple data sets. Hence, by empirical experimentation, we identify a suitable learning rate and batch size combination that produce stable, consistent, and high-quality results across all data sets. We employ a batch size of 2048 with the Adam optimizer [42], setting the learning rate at $5e-5$ and the two Adam optimizer coefficients β_1 and β_2 to their default values at 0.9 and 0.999, respectively. The learning rate is decayed by a factor of 0.1 if the loss value does not decrease for 10 consecutive epochs. During MCDropout method, a training dropout probability of $\eta = 0.05$ and a testing dropout probability of $\eta = 0.1$ is applied consistently, while no dropout is used for training ensemble models. Both MCDropout and Ensemble methods undergo training for 500 epochs. Further discussion on the hyperparameter selection and impact of network architecture on reconstruction quality and uncertainty estimates are provided in Section 6.

4 UNCERTAINTY-INFORMED FLOW FIELD VISUALIZATION

We conduct a thorough study of our models using six vector field data sets. The dimensionality and spatial resolution of these data sets are reported in Table 1. We use a GPU server with NVIDIA GeForce GTX 1080Ti GPUs with 12GB GPU memory for all the experimentation. All the models are implemented in PyTorch [54]. The Heated Cylinder data set [25] and Fluid data set [40] are generated using Gerris flow solver [55]. Hurricane Isabel data was produced by the Weather Research and Forecast model, courtesy of NCAR and NSF. Turbine data set [12] and Tornado data [13] set are made available by Dr. Jen-Ping Chen and Dr. Roger Crawfis, respectively, at the Ohio State

University. Tangaroa data set [56] is a simulation of an incompressible 3D flow around a CAD model of the research vessel Tangaroa.

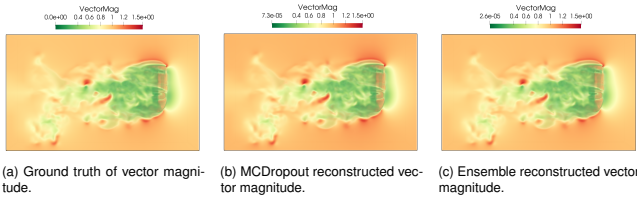


Fig. 3: Volume visualization of the magnitude of reconstructed vector fields using MCDropout and Ensemble methods for Tangaroa data set. The ground truth is shown in Fig. 3a. We observe that both methods produce visually comparable results.

4.1 Visual Analysis of Reconstructed Field, Prediction Uncertainty, and Error

We reconstruct the entire vector field to thoroughly assess the model’s reconstruction quality, prediction uncertainty, and error. Through experimentation, we ascertain that utilizing 100 Monte Carlo (MC) samples for the MCDropout method and 30 ensemble members for the Ensemble method yields a robust estimation of vectors, error, and uncertainty. The determination of these numbers of MC samples and ensemble members is based on comparing the reconstruction quality across various quantities of MC samples and ensemble members. Detailed results can be found in Table 3 and Table 4. Thus, unless specified otherwise, we consistently employ 100 MC samples for MCDropout and 30 ensemble members for the Ensemble method in all presented results.

To derive the final vector field, we calculate the average vectors at each grid point, where the averaging process entails computing the mean over 100 vectors for the MCDropout method and over 30 vectors for the Ensemble method. We compute the error at each grid point by contrasting the predicted value with the ground truth vectors to estimate fine-grained prediction error. This error computation involves assessing each vector component individually and then adding the component-wise errors, i.e., the L1 norm, to obtain the total error at a grid point. Similarly, we compute the component-wise uncertainty at each grid point (by computing the component-wise standard deviation) for both MCDropout and Ensemble methods, subsequently estimating the total uncertainty by summing the component-wise uncertainty values.

This section focuses on a qualitative comparison, presenting volume visualizations of the reconstructed vector magnitude fields, estimated prediction uncertainty, and error values for the Isabel and Tangaroa data sets. Fig. 2 and Fig. 3 display the volume rendering of ground truth, MCDropout, and Ensemble predicted vector magnitude fields for Isabel and Tangaroa data sets, respectively. Both methods accurately reconstruct the vector components for these data sets. Subsequently, Fig. 4 and Fig. 5 present volume renderings of predicted error and uncertainty fields for the Isabel and Tangaroa data sets. It is noted that error and uncertainty do not exhibit strong spatial correlation. While the error patterns for MCDropout and Ensemble methods appear visually similar, spatial uncertainty visualizations differ. In the Isabel data set, regions with higher prediction uncertainty correspond to vortex regions in the vector field for both MCDropout and Ensemble methods (see Fig. 4c and Fig. 4d). However, in the Tangaroa data set, the relatively higher uncertainty valued regions are confined to smaller spatial domain for the Ensemble method (Fig. 5d). For MCDropout method, we see that the moderately uncertain regions are widespread (Fig. 5c), compared to the Ensemble method.

4.2 Visual Analysis of Flow Features Using Uncertainty-Aware Streamlines

Streamlines serve as valuable tools for visualizing flow patterns, identifying recirculation regions, and pinpointing stagnation points, aiding in understanding fluid behavior across engineering and scientific domains. To enhance the effectiveness of flow feature visualization, we employ both MCDropout and Ensemble methods with our model’s predictions. These methods not only generate streamlines but also allow

us to quantify uncertainty in predicted vector values at each step of the integration process. To generate streamlines, we use RK-4 integration and trace the streamline in both forward and backward direction. Using the MCDropout approach, we generate n realizations of the same streamline from a given seed point (with $n=100$ in our study) through MC sampling. To estimate the expected streamline, we compute the mean streamline by averaging the coordinates at each integration step. However, due to variability among the MC streamline samples, some streamlines may go out of bounds. In such instances, we calculate the mean using only the existing streamlines at those integration steps. Our analysis method also supports the computation of the median streamline as an alternative to the mean streamline to mitigate the impact of outliers. The uncertainty at each step of the streamline is determined by the standard deviation computed from the n samples following similar strategy as used to compute the mean streamline. Similarly, employing the Ensemble method involves generating n realizations of the streamline (where $n = 30$ due to our use of 30 ensemble members), considering the mean streamline as the final result, and computing pointwise standard deviation as the uncertainty estimate.

Utilizing models sensitive to uncertainty helps measure and communicate uncertainty associated with streamlines, aiding users in making informed decisions during flow feature analysis [18]. This uncertainty information builds user confidence in the model’s predictions and highlights segments where predictions may lack confidence. We visualize uncertainty in streamlines by rendering the mean (averaged) streamline as a stream-tube, with the stream-tube diameter scaled by uncertainty estimates and colored by uncertainty values. Fig. 6 shows this stream-tube visualization, where Fig. 6a displays the ground truth streamline without uncertainty, and Fig. 6b shows the uncertainty-aware visualization. Higher uncertainty is observed at the streamline’s ends, with the seed location marked by a green sphere. Next, we qualitatively study the uncertainty characteristics estimated by MCDropout and Ensemble methods through several case studies.

Fluid (2D) Data Set. We use 100 uniformly randomly generated seed points to generate streamlines for the Fluid data set using MCDropout and Ensemble methods. The results are shown in Fig. 7. The ground truth streamlines are shown in Fig. 7a, and streamlines from MCDropout and Ensemble methods are provided in Fig. 7b and Fig. 7c respectively. The streamlines are colored using prediction uncertainty, and the thickness of the stream tubes also varies according to the prediction uncertainty. It is observed that both MCDropout and Ensemble methods can generate accurate streamlines when compared against ground truth streamlines (Fig. 7a). Furthermore, by comparing Fig. 7b and Fig. 7c, we also learn that both MCDropout and Ensemble methods estimate comparable uncertainty estimates across the entire data set.

Tornado Data Set. Next, we conduct a similar study using the Tornado data set to study the generated uncertainty patterns. In this study, streamlines on 100 uniformly randomly generated seed points are shown, where the ground truth streamlines are shown in Fig. 8a. We find that both MCDropout and Ensemble methods produce visually accurate streamlines. We also notice that while the uncertainty patterns are generally comparable, several streamlines demonstrate different uncertainty patterns between MCDropout and Ensemble methods as seen from Fig. 8b and Fig. 8c respectively.

Hurricane Isabel Data Set. To study the uncertainty-aware streamlines in the Hurricane Isabel data set, we generate eight streamlines by seeding around the vortex core region in the data set. The ground truth streamlines are shown in Fig. 9a. Fig. 9b and Fig. 9c depict the uncertainty-aware streamlines generated by MCDropout and Ensemble methods respectively. Again, we observe a comparable uncertainty pattern for both uncertainty estimation methods.

Turbine Data Set. Finally, uncertainty-aware streamline visualization for the Turbine data set is presented in Fig. 10. We show a visualization of a selected streamline, generated by the MCDropout method, in Fig. 10a. Visualization of the same generated by the Ensemble method is given in Fig. 10b. The rotor blades in the Turbine data set are shown as a context using gray surface rendering, and the ground truth streamline (in red color) is overlaid for visual comparison. Similar to the previous case studies, we observe comparable uncertainty

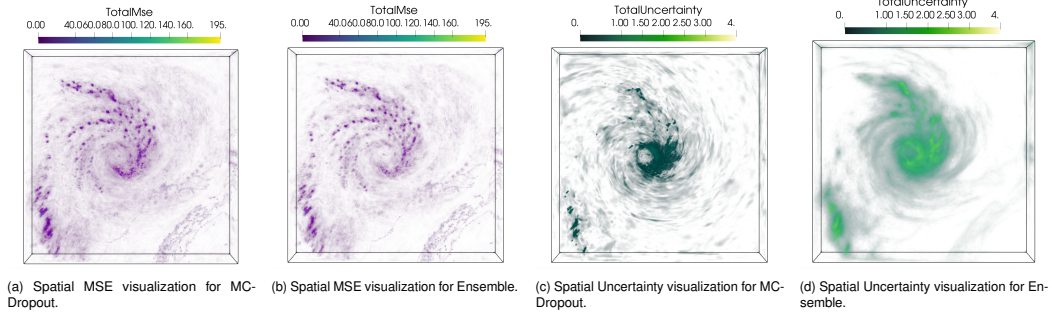


Fig. 4: Visualization of uncertainty and error fields for Hurricane Isabel data set. The MSE and prediction uncertainty is estimated at each grid point between the predicted and ground truth vectors for MCDropout and Ensemble methods. Fig. 4a and Fig. 4b show the rendering of MSE fields, and Fig. 4c and Fig. 4d present the rendering of uncertainty fields. We observe that the locations with higher MSE correspond to similar spatial regions for both methods. In contrast, the vortex region is detected as a region with higher prediction uncertainty for both methods.

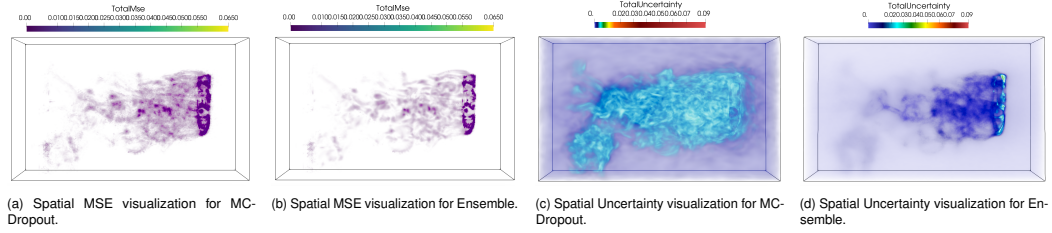


Fig. 5: Visualization of uncertainty and error fields for Tangaroa data set. The MSE and prediction uncertainty are estimated at each grid point between the predicted and ground truth vectors for MCDropout and Ensemble methods. Fig. 5a and Fig. 5b show the rendering of MSE fields, and Fig. 5c and Fig. 5d present the rendering of uncertainty fields. Results indicate that regions with higher MSE align spatially for both methods, while areas with higher prediction uncertainty are more widespread for MCDropout compared to the Ensemble method.

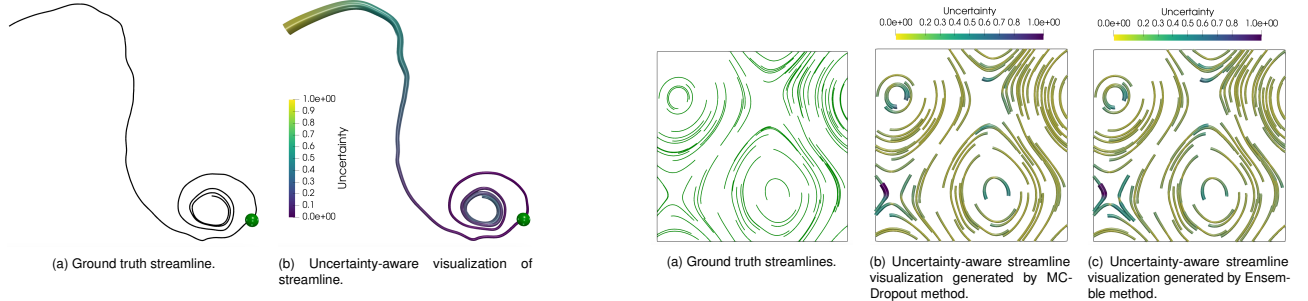


Fig. 6: Fig. 6a shows the ground truth. Fig. 6b shows the uncertainty-aware streamline from the Ensemble method, where the streamline is visualized as a stream tube, colored with uncertainty, and the diameter of the tube is varied using prediction uncertainty.

Fig. 7: Uncertainty-aware streamline visualization for Fluid 2D data. The streamlines are generated using 100 uniformly randomly generated seeds. Fig. 7a shows the ground truth streamlines, Fig. 7b and Fig. 7c show streamlines generated by MCDropout and Ensemble method, respectively. The streamlines are colored using uncertainty values and rendered using stream tube visualization, where the diameter of the tube is varied using the prediction uncertainty values.

estimates from the MCDropout and Ensemble methods.

4.3 Visual Analysis of Flow Features Using Uncertainty-Aware Critical Points

Critical points play an important role in studying vector field characteristics as it helps identify points where the flow behavior undergoes significant changes. These points, including saddles, sources, centers, and sinks, provide valuable insights into the dynamics of the vector field. By observing critical points, users determine flow patterns, locate stagnation points, and comprehend the overall flow behavior. Comprehensive understanding of robustness of critical points can lead to the accurate flow features study [16, 24, 72] and flow data compression [45].

In this study, we evaluate the accuracy of uncertainty-aware neural models in predicting critical point locations and quantify the spatial variability of these predictions using the MCDropout and Ensemble methods. Initially, we compute the mean vector field using both techniques to ensure the robustness of the estimated critical points. From this mean-field, we identify the critical point locations. To assess the spatial variability of each critical point's location derived from the mean field, we analyze data from individual vector field realizations: 100

fields for the MCDropout and 30 for the Ensemble method. For each method, first, we identify critical points in each vector field realization. Then, we create a new scalar field where the scalar value at each grid point reflects the cumulative contribution of all critical points from all the field realizations. Here, we do not distinguish between critical points detected from different MC samples (or ensemble members). Each critical point deposits its contribution to all the grid points. The contribution of a critical point to a grid point is calculated as the inverse Euclidean distance between the critical point and the grid point. Hence, the contribution of a critical point to the nearby grid points will be higher and gradually fall off for far away grid points. Formally, for a grid point P , we compute its scalar value as follows: $v_P = \sum_{i=1}^N \frac{1}{D(P, C_i)}$, where v_P denotes the value at grid point P and $D(P, C_i)$ is the Euclidean distance between point P and critical point C_i with N being the total number of critical points detected across all realizations.

Therefore, a grid point with a higher value will indicate a relatively

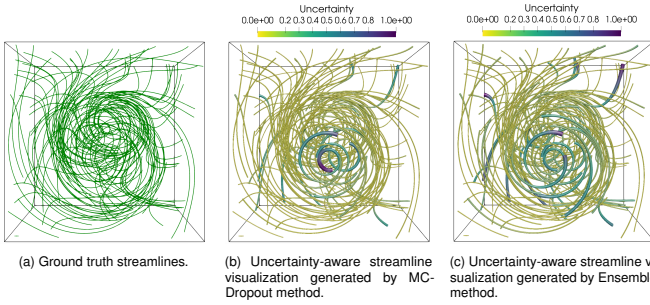


Fig. 8: Uncertainty-aware streamline visualization for Tornado data. The streamlines are generated using 100 uniformly randomly generated seeds. Fig. 8a shows the ground truth streamlines, Fig. 8b and Fig. 8c show streamlines generated by MCDropout and Ensemble method, respectively. The streamlines are colored using uncertainty values and rendered using stream tube visualization, where the diameter of the tube is varied using the prediction uncertainty values.

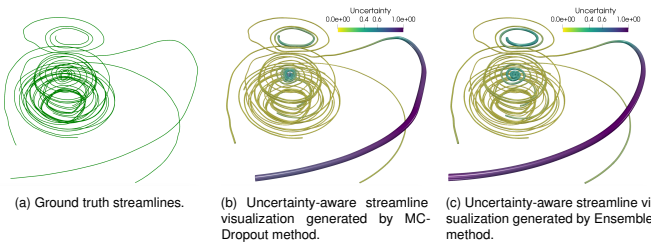


Fig. 9: Uncertainty-aware streamline visualization for Hurricane Isabel data. The streamlines are generated using 8 seed points placed around the vortex region in the data. Fig. 9a shows the ground truth streamlines, Fig. 9b and Fig. 9c show streamlines generated by MCDropout and Ensemble method respectively. The streamlines are colored using uncertainty values and rendered using stream tube visualization, where the diameter of the tube is varied using the prediction uncertainty values.

large number of critical points near that grid point generated from all the realizations. Now, consider a spatial region in this newly constructed scalar field around a given mean critical point extracted from the mean field. Suppose we observe relatively high scalar values for the grid points in this region. In that case, it will indicate that when extracted from each realization, this critical point’s location is scattered within that region, resulting in higher scalar values for most grid points. Conversely, if the detected critical points across all realizations are almost identical with minimal variability, they will only contribute heavily to a few nearby grid points. Thus, by examining the spatial spread of relatively high-valued regions around each mean critical point in this scalar field, we can determine whether the detected critical points across all the field realizations are concentrated (i.e., high certainty) or spatially scattered (i.e., low certainty).

In Fig. 11 and Fig. 12, we show the results of the critical point analysis using two time steps ($T=750$ and $T=1500$) of the Heated Cylinder data set. We show the ground truth critical points with the vector field rendered using surface LIC visualization in Fig. 11a and Fig. 12a for time step 750 and 1500, respectively. The saddles and centers are shown using red and blue dots. The results obtained from the MCDropout and Ensemble methods are shown in Fig. 11b and Fig. 11c for time step 750. We observe that both methods correctly predict all the critical points from the reconstructed vector fields. The green shade in the background highlights the spatial variability that we computed as a scalar field, as discussed above. We observe that the MCDropout method produces high prediction variability for a critical point indicated by the bright green spot around it. However, the Ensemble method makes highly confident predictions for all the critical points.

Similarly, the visualization of critical points for time step 1500 for the Heated Cylinder data is depicted in Fig. 12. Here, we observe that the MCDropout method fails to detect a critical point at the boundary of

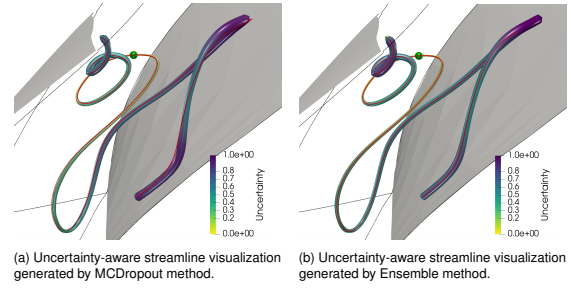


Fig. 10: Uncertainty-aware streamline visualization for Turbine data. We show an uncertainty-informed streamline of a selected seed point. The ground truth (colored in red) is overlaid with the uncertain stream-tube as a reference, and the seed is shown as a green sphere. By comparing the streamline generated by the MCDropout method (Fig. 10a) and the Ensemble method (Fig. 10b), it is observed that both the methods produce visually similar and comparable uncertainty.

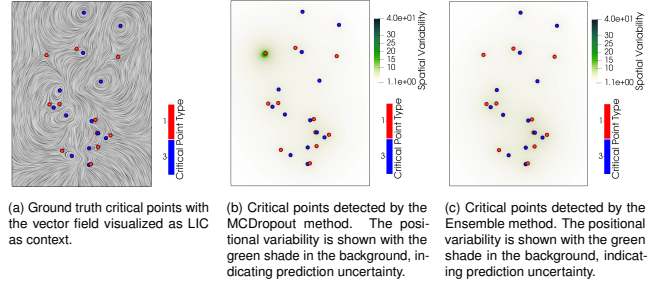


Fig. 11: Result of critical point analysis for Heated Cylinder data set ($T=750$). Critical points detected by MCDropout and Ensemble methods are shown in Fig. 11b and Fig. 11c, respectively. The green highlighted regions in the background show the variability in the predicted critical point’s locations, with brighter green indicating higher variability. The ground truth is shown in Fig. 11a. We observe that both methods detect all the critical points, and the MCDropout method produces higher prediction variability than the Ensemble method.

the data set (see red dotted region in Fig. 12b). This shows a potential limitation of computing the final predicted flow field as the average over multiple instances of predicted flow fields. Investigating individual predicted sampled fields and visualizing the median flow field can help recover such missing critical points. In contrast, the Ensemble method successfully retains all the critical points (see Fig. 12c) when compared against the ground truth (Fig. 12a). Both methods show similar spatial variability, as indicated by the bright green spots around several critical points. A quantitative comparison on the accuracy of the predicted critical points is provided later in Section 5.

5 QUANTITATIVE EVALUATION AND PARAMETER STUDY

Table 2: Comparison of reconstruction quality (PSNR), error (RMSE), and model storage overhead between MCDropout and Ensemble methods.

Data Set	Deep Ensemble (30 Members)			MCDropout (100 MC samples)		
	Model Size (KB)	PSNR (dB) \uparrow	RMSE \downarrow	Model Size (KB)	PSNR (dB) \uparrow	RMSE \downarrow
Heated Cylinder ($T=750$)	24364	57.799	0.00209	812	55.32	0.00278
Heated Cylinder ($T=1500$)	24364	58.417	0.00251	812	54.913	0.00375
Fluid	24364	76.897	0.00019	812	73.953	0.00027
Hurricane Isabel	48612	53.602	0.51843	1620	52.176	0.61607
Tornado	48608	72.832	0.00493	1620	69.5	0.00721
Turbine	48604	53.361	0.00811	1620	51.545	0.01053
Tangaroa	48572	68.665	0.00152	1620	65.008	0.00232

Reconstruction Quality and Prediction Error. Table 2 shows a comparison between the Ensemble and MCDropout methods concerning model size, PSNR, and RMSE. The Ensemble method outperforms MCDropout in terms of both PSNR and RMSE. However, achieving this superior reconstruction quality with the Ensemble method requires nearly 30 times more storage space and training time compared to

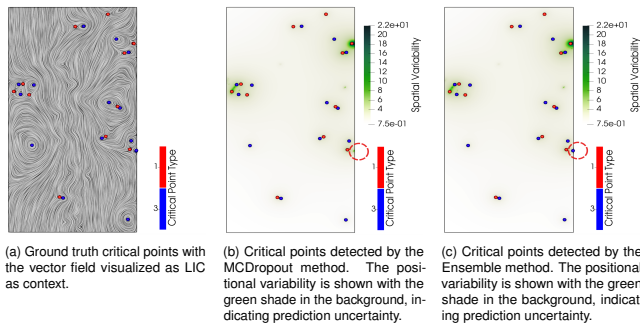


Fig. 12: Result of critical point analysis for Heated Cylinder data set (T=1500). Critical points detected by MCDropout and Ensemble methods are shown in Fig. 12b and Fig. 12c, respectively. The green highlighted regions in the background show the variability in the predicted critical point’s locations, with brighter green indicating higher prediction variability. The ground truth critical points are shown in Fig. 12a. We further observe that the MCDropout method fails to detect a critical point at the boundary, as shown by the red dotted circle in Fig. 12b, which is successfully detected by the Ensemble method.

Table 3: PSNR (dB) values when the varying number of MC samples are used for computation in the MCDropout method. PSNR increases slowly as the number of MC samples increases. Here, we use 100 MC samples to produce stable reconstruction and uncertainty estimates.

Data Set	PSNR (\uparrow) for diff. #MCSamples		
	50	100	150
Heated Cylinder (T=750)	55.182	55.32	55.37
Heated Cylinder (T=1500)	54.781	54.913	54.955
Fluid	73.29	73.953	74.204
Hurricane Isabel	52.175	52.176	52.176
Tornado	69.497	69.5	69.501
Turbine	51.519	51.545	51.553
Tangaroa	65.003	65.008	65.01

MCDropout, as it involves training 30 ensemble members. It is worth noting that the PSNR of the Ensemble method is computed using predictions from 30 ensemble members, while for MCDropout, it is calculated based on 100 Monte Carlo (MC) samples. The rationale behind the chosen number of samples is elaborated on subsequently.

Impact of Different Number of MC Samples on PSNR Value for MCDropout Method. Table 3 demonstrates the PSNR value of the vector field obtained by averaging different numbers of MC samples. There is a slight gain of PSNR as we increase the number of MC samples up to 150. After that, the gains in PSNR are not as significant to justify considering more number of samples. Hence, to find a good trade-off between computation time and prediction quality, we use 100 MC samples for all experiments involving the MCDropout method.

Impact of Different Number of Ensemble Members on PSNR Value for Ensemble Method. The effect of different numbers of ensemble members on PSNR is depicted in Table 4. The number of ensemble members required to produce robust prediction is much lower than in the MCDropout method. It can be observed that PSNR gains are insignificant when increasing the number of ensemble members beyond 15 up to 30. Hence, at around 15 ensemble members, the PSNR becomes primarily saturated. However, for consistency, we use

Table 4: PSNR (dB) values when varying numbers of ensemble members are used for the Ensemble method. We observe that PSNR increase is almost insignificant beyond 15 members upto 30.

Data Set	PSNR (\uparrow) for diff. #Ensemble members			
	15	20	25	30
Heated Cylinder (T=750)	57.203	57.347	57.595	57.799
Heated Cylinder (T=1500)	58.193	58.312	58.305	58.417
Fluid	76.795	76.85	76.734	76.897
Hurricane Isabel	53.626	53.589	53.57	53.602
Tornado	72.735	72.81	72.899	72.832
Turbine	53.366	53.433	53.301	53.361
Tangaroa	68.443	68.536	68.556	68.665

Table 5: Streamline error analysis for MCDropout and Ensemble method using Chamfer and Hausdorff distance metrics. We generate streamlines from 100 uniformly randomly selected seed points for each data set and report their average Chamfer and Hausdorff distances when compared against ground truth streamlines. We observe that, on average, the Ensemble method produces more accurate streamlines.

Data Set	Deep Ensemble		MCDropout	
	Avg. Chamfer Distance \downarrow	Avg. Hausdorff Distance \downarrow	Avg. Chamfer Distance \downarrow	Avg. Hausdorff Distance \downarrow
Heated Cylinder (T=750)	0.00104	0.00095	0.00776	0.004
Heated Cylinder (T=1500)	0.00062	0.00062	0.0037	0.00189
Fluid	0.00453	0.00469	0.01261	0.01314
Hurricane Isabel	1.816	1.95277	2.54561	2.86682
Tornado	0.00261	0.00296	0.0096	0.00938
Turbine	0.00167	0.00127	0.00215	0.00342
Tangaroa	0.00024	0.00036	0.0004	0.00063

Table 6: Average RMSE in predicted critical point locations are reported for MCDropout and Ensemble methods. We observe that the Ensemble method produces more accurate critical points.

Data Set	Avg. RMSE (\downarrow) for Deep Ensemble	Avg. RMSE (\downarrow) for MCDropout
Heated Cylinder (T=750)	0.01866	0.02896
Heated Cylinder (T=1500)	0.03234	0.03481
Fluid	0.22243	0.24127

30 ensemble members for the experiments in the work.

Streamline Error Analysis for MCDropout and Ensemble Methods. We examine the precision of streamlines generated by both MCDropout and Ensemble methods, utilizing averaged mean streamlines for comparison. To conduct a thorough quantitative assessment, we generate mean streamlines from 100 randomly uniformly distributed seed points for each method. Subsequently, we gauge the streamline error by measuring Chamfer distance [4] and Hausdorff distance between the predicted averaged streamlines and the ground truth counterparts. It’s noteworthy that for the MCDropout method, we aggregate results over 100 Monte Carlo (MC) samples, while for the Ensemble method, we utilize streamlines predicted by 30 ensemble members. The average Chamfer distance and Hausdorff distance values computed over the 100 streamlines for each data set are presented in Table 5. It is evident that the Ensemble method consistently yields more precise streamlines compared to the MCDropout method across all data sets, as evidenced by the lower Chamfer and Hausdorff distance values.

Accuracy Analysis of The Predicted Critical Points for MCDropout and Ensemble Methods. To quantitatively assess the precision of the identified critical point locations, we calculate the average root mean squared error (RMSE) between the predicted and the corresponding ground truth critical points for both MCDropout and Ensemble methods. The outcomes are documented in Table 6 for the Heated Cylinder and Fluid data sets. It is apparent that, on average, the Ensemble method outperforms the MCDropout method in accurately determining critical points using the reconstructed vector field.

Impact of Varying Number of Dropout Layers on MCDropout Model Performance. In [19], Gal et al. demonstrate that incorporating a dropout layer after each hidden layer in a DNN renders the model theoretically equivalent to conducting inference in a fully Bayesian neural network. However, Kendall et al. [41] argue that adding such dropout layers after every layer could hamper the model’s learnability,

Table 7: We report how the reconstruction quality (PSNR) changes when different numbers of Dropout layers are used for the MCDropout method. Three configurations are studied: Dropout used (1) only at the last residual block; (2) at the last half of the residual blocks; (3) at all the residual blocks. We observe that PSNR decreases with an increasing number of Dropout layers as it acts as a strong regularizer. Hence, Dropout only at the last residual block is used.

Data Set	PSNR (\uparrow) with different Dropout Layer Location		
	Last Res. block	Last half Res. blocks	All Res. blocks
Hurricane Isabel	52.176	51.686	49.595
Turbine	51.545	50.16	39.274

Table 8: PSNR (dB) of 100 MC samples with different Dropout probabilities during inference.

Data Set	PSNR (\uparrow) for different test-time Dropout probabilities (100 MC Samples)					
	0.05	0.1	0.2	0.3	0.4	0.5
	Heated Cylinder ($T=750$)	55.391	55.32	55.171	54.935	54.678
Heated Cylinder ($T=1500$)	54.983	54.913	54.758	54.564	54.306	53.984
Fluid	74.352	73.953	73.139	72.282	71.328	70.283
Hurricane Isabel	52.176	52.176	52.175	52.173	52.172	52.169
Tornado	69.501	69.5	69.497	69.493	69.487	69.479
Turbine	51.556	51.545	51.514	51.472	51.424	51.347
Tangaroa	65.011	65.008	65.002	64.994	64.983	64.968

Table 9: Training time (Hours) for MCDropout and Ensemble (30 ensemble members) for 500 epochs with a batch size of 2048 and inference time (seconds) for generating final vector field using MCDropout (100 MC samples) and Ensemble (30 ensemble members) methods.

Data set	Training time (Hours)		Inference time (Seconds)	
	MCDropout	Ensemble (30 members)	MCDropout (100 MC samples)	Ensemble (30 members)
	Heated Cylinder ($T=750$)	0.27	7.12	2.84
Heated Cylinder ($T=1500$)	0.27	7.12	2.85	1.83
Fluid	0.64	18.06	12.33	3.99
Hurricane Isabel	7.03	218.06	182.14	54.48
Tornado	4.73	151.8	122.13	37.99
Turbine	1.6	59.46	32.39	11.51
Tangaroa	14.86	462.68	389.89	114.13

as numerous dropout layers act as potent regularizers. This aligns with our observations of decreased reconstruction quality when applying dropout at each residual block in the MCDropout model. Kendall et al. [41] further propose to use fewer dropout layers, approximating a partial Bayesian network, to achieve robust uncertainty estimation. They suggest that a dropout layer after the final layer is sufficient to produce high prediction accuracy and robust uncertainty estimates. Consequently, we use dropout layer at the last residual block to ensure high-quality predictions and robust uncertainty estimates. In Fig. 13, we show results of uncertainty estimates for a streamline selected from Isabel data set. We present uncertainty visualization using model with dropout added at the (1) last residual block, (2) last half of the blocks, and (3) all residual blocks. We observe that all of these configurations produce similar uncertainty estimates. To assess the detailed impact of varying the number of dropout layers, we conduct experiments on the Isabel and Turbine data sets. The results, detailed in Table 7, indicate a decrease in PSNR with increasing number of dropout layers.

Impact of Varying Dropout Probabilities on MCDropout Model Performance. Table 8 depicts a study examining the quality of vector field reconstruction (measured by PSNR) averaged over 100 MC samples across various test-time dropout probabilities. The MCDropout model is trained with a dropout rate of 0.05. Notably, it is observed that PSNR remains largely consistent up to a dropout probability of 0.2, consistent with findings by Gal et al. [19]. They posit that once a model has reached convergence, minimal variations in dropout probability do not significantly impact the predicted quality. However, as dropout is increased to as high as 0.5, the prediction quality gradually diminishes, as evidenced by lower PSNR values.

Consistent Hyperparameter Selection. In this work, we seek to identify a hyperparameter combination that ensures stable training of implicit neural models using both MCDropout and Ensemble methods

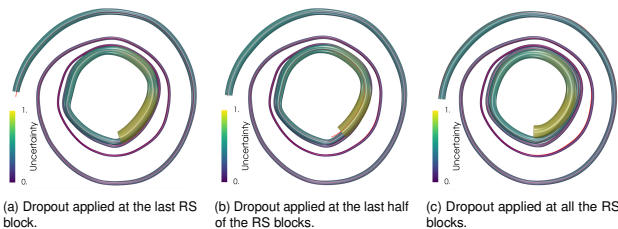


Fig. 13: Comparing uncertainty patterns when dropout is added at the last (Fig. 13a), last half (Fig. 13b), and all residual blocks (Fig. 13c). The uncertainty pattern remains identical for these configurations. Ground truth streamline is overlaid in red as reference.

Table 10: Evaluating reconstruction quality under different hyperparameter combinations for MCDropout method.

Learning Rate	Turbine PSNR (100 MC samples)			Tornado PSNR (100 MC samples)			Isabel PSNR (100 MC samples)		
	Batch Size			Batch Size			Batch Size		
	1024	2048	4096	1024	2048	4096	1024	2048	4096
0.0001	20.791	51.854	50.9	21.094	21.103	67.757	27.1	27.364	27.604
0.0005	19.498	19.957	20.074	19.076	19.716	19.414	25.598	25.726	25.881
1e-5	46.338	41.776	43.662	67.893	67.135	65.879	52.286	51.973	51.446
5e-5	51.934	51.545	50.43	71.036	69.5	69.703	30.888	52.176	52.195

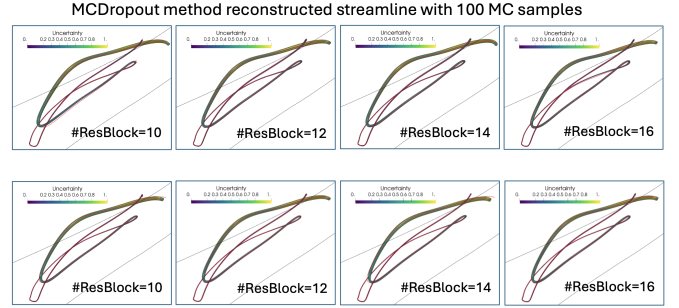


Fig. 14: Visualization of a representative streamline from Turbine data set under varying numbers of residual blocks for MCDropout and Ensemble methods. It is observed that the estimated prediction uncertainty is robust and is not influenced when the number of residual blocks is varied.

across multiple flow data sets. Since both methods share the same base network architecture, we perform the hyperparameter selection experiment using the MCDropout method as it is a single-model based approach and requires less number of model training compared to the expensive ensemble method. After determining a stable hyperparameter combination, we apply these same parameters to train the ensemble models, aiming for stable, consistent, and comparable results. Table 10 presents the reconstruction quality, measured by PSNR, for 12 different hyperparameter combinations with varying learning rates and batch sizes on the Turbine, Isabel, and Tornado data sets. The combination of a learning rate of $5e-5$ and a batch size of 2048 consistently produces high-quality, stable PSNR values across all three data sets. Consequently, we adopt this hyperparameter combination for subsequent training of both MCDropout and Ensemble models. Our results show that this combination yields consistent and comparable uncertainty-aware neural representations across all data sets.

Table 11: Reconstruction quality with varying network depth.

#Res Blocks	MCDropout (PSNR (\uparrow) for 100 MC samples)				Ensemble (PSNR (\uparrow) for 15 ensemble members)			
	10	12	14	16	10	12	14	16
Tornado	67.21	68.79	69.50	72.21	71.58	73.05	72.70	37.08
Turbine	50.89	50.68	51.55	49.97	53.84	54.96	53.80	55.43
Isabel	51.66	51.74	52.18	28.39	52.87	52.68	53.69	35.24

Impact of Varying Model Architecture on Accuracy and Uncertainty Estimates. We vary the network depth by adjusting the number of residual blocks, while keeping the number of neurons per layer fixed. The learning rate and batch size are maintained at $5e-5$ and 2048, respectively, to ensure comparability of the results. From Table 11, we observe that PSNR values exhibit minimal variation across different numbers of residual blocks. However, an exception occurs with 16 residual blocks: the PSNR drops for the Tornado data set using the Ensemble method and for the Isabel data set using the MCDropout method. This drop is due to unstable training, which would require further hyperparameter tuning. Additionally, we examine the impact of varying the number of residual blocks on the estimated prediction uncertainty. We find that prediction uncertainty remain unaffected by changes in the model architecture. In Fig. 14, we present a representative streamline from the Turbine data under different numbers of residual blocks for both methods. It is evident that the prediction uncertainty is robust and remains consistent regardless of the number of residual blocks. However, for a much shallower network, we anticipate that the quality of reconstruction and uncertainty estimates will gradually degrade as the network will have less learning ability.

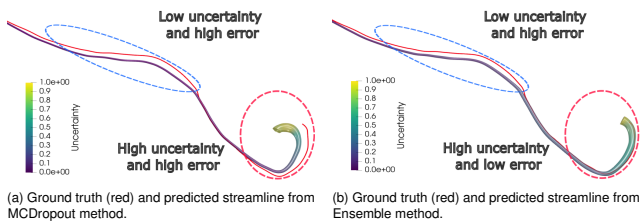


Fig. 15: The importance of error and uncertainty information for vector field feature analysis using streamlines generated by a DNN model.

Training and Inference Time for MCDropout and Ensemble Methods. Table 9 shows training and inference times for the two methods. The Ensemble approach has faster inference time since it uses predictions from only 30 members, whereas MCDropout generates 100 MC samples, resulting in longer inference times. However, Ensemble training is significantly more time and resource consuming because it requires training multiple models (30 in this study), unlike MCDropout, which needs to train a single model. This significant difference in training time often makes MCDropout preferable for timely results with robust uncertainty estimates.

6 DISCUSSION

Uncertainty-agnostic vs. Uncertainty-aware DNNs. We advocate the use of DNNs equipped to quantify uncertainty when analyzing vector fields. We find that uncertainty-informed neural networks offer valuable insights into the reliability of their predictions. By effectively conveying such uncertainty to experts, they can make well-informed decisions about the data features. This integration of uncertainty is crucial in fostering trust in the use of DNN-predicted results for scientific research. Particularly when dealing with large vector fields, scientists often have to rely on the model’s outputs due to the impracticality of handling full-resolution data. In such cases, where ground truth data may be unavailable, estimating errors becomes challenging. However, uncertainty-aware DNNs can still provide reliable uncertainty estimates, offering experts greater confidence in interpreting predicted results. Next, we discuss the domain expert feedback to highlight the implications of our proposed techniques.

Domain Expert Feedback. We interviewed a computational fluid dynamics scientist to collect expert feedback. The expert immediately liked the idea of using uncertainty-aware neural models and agreed that visualizing prediction uncertainty on predicted flow features is critical and can provide meaningful insights about the prediction’s quality and help build trust in using deep learning techniques for scientific research. The expert found our uncertainty-informed streamline visualization intuitive and easy to comprehend. The expert was also intrigued to see that these neural models can be both under-confident and over-confident, as highlighted in Fig. 15, and agreed that access to such results is critical for verifying and validating hypotheses from flow data. Next, the expert suggested that developing such uncertainty-aware models for time-varying flow data would be very useful. Finally, the expert observed that complex and turbulent regions tend to incur higher prediction uncertainty, which was expected since predicting intricate flow features is a more complicated task than predicting less turbulent flow. This is because of the highly nonlinear and multiscale nature of turbulent flows. However, the expert further noted that large-scale ocean and upper atmospheric flows tend to be non-turbulent, and hence the proposed method can serve as an effective uncertainty-aware compression method for such large-scale flow data.

Mean vs. Median Streamline. Formally, both MCDropout and Ensemble methods consider the expected model output to be the mean and so we visualize the mean streamlines. However, to mitigate outlier sensitivity, our method also allows visualization of the median streamline, which is less affected by outliers. Experiments show that mean and median streamlines are nearly identical, with minor differences. In Fig. 16, results from the Isabel data show that for both methods, mean (green) and median (blue) streamlines generally overlap, though the zoomed inset reveals slight differences for the Ensemble method.

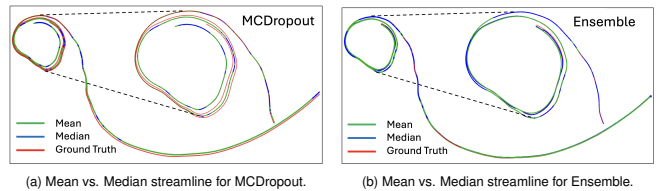


Fig. 16: We show representative mean and median streamlines for MCDropout and Ensemble method. We observe that mean and median streamlines are almost identical while the zoomed view on the right shows minor differences between them for the Ensemble method.

Error vs. Uncertainty. Our experiments also reveal instances where the models may exhibit high confidence (low uncertainty) in predicting a segment of a streamline, yet this segment can have high errors. This poses a potential limitation for uncertainty-aware methods, as users might trust predictions guided by high model confidence, only to find the results partially erroneous. This observation is illustrated in Fig. 15a and Fig. 15b, where we present a streamline from the Isabel data set. Both MCDropout and Ensemble methods predict a segment of the streamline (marked by blue dotted region) with low uncertainty, but when compared against the ground truth streamline (depicted in red), we discover high prediction errors. Conversely, we identify another segment of the streamline (highlighted by the red dotted region), where the MCDropout method exhibits high uncertainty, and the segment also shows high error. However, for the Ensemble method, this segment demonstrates low error despite the method yielding high uncertainty. These findings suggest that while uncertainty can aid in communicating the confidence of predicted results, it may also lead to situations where the model is confident but the result is erroneous. Therefore, additional methods of model explainability will be necessary for such cases.

MCDropout vs. Ensemble Method. This work investigates two approaches for estimating uncertainty using an INR to analyze steady-state vector fields. We use Deep Ensemble, a widely accepted standard [39, 50] despite its extensive training time, and explore single-model-based MCDropout method to mitigate computational challenges. MCDropout method is chosen for its theoretical elegance and ease of integration into existing DNN models with minimal adjustments [19]. Our findings show that both MCDropout and Ensemble methods produce comparable uncertainty patterns, but the Ensemble method offers slightly more accurate predictions at the cost of significantly longer training times (Table 9). Therefore, in scenarios where resources are limited and immediate uncertainty estimation is crucial, MCDropout proves to be a dependable alternative to the Ensemble method.

Extension to Time-varying Data. A natural extension of this work is to perform uncertainty-aware flow analysis on time-varying data. Time-varying data sets will be significantly larger and more challenging to model. Instead of directly modeling the time-varying vector fields, a computationally viable alternative could be to use flowmap-based representations [63] or directly learning particle tracing results [33]. Additionally, due to the high computational cost, training an Ensemble for uncertainty estimation may not be feasible and MCDropout and other single-model methods will be preferable.

7 CONCLUSIONS AND FUTURE WORK

This paper emphasizes the importance of understanding uncertainty by applying two uncertainty estimation methods. While in this work, we study the impact of uncertainty on tasks such as vector data prediction, streamline generation, and critical point detection, in the future, we plan to conduct a comprehensive topological analysis and thoroughly study flow map characteristics to evaluate reconstruction quality further using the proposed methods. Other future research endeavors include exploring alternative deep uncertainty estimation techniques and expanding our method to handle time-varying data. Insights from uncertainty estimates help identify areas needing explicit training and recognize model limitations in specific data regions. In critical scenarios, a model’s confidence in its predictions is crucial, fostering greater trust when the model acknowledges its uncertainties.

ACKNOWLEDGMENTS

We acknowledge the anonymous reviewers for their insightful comments. We thank Prof. Nairita Pal from IIT Kharagpur, India for providing constructive expert feedback for the proposed work and suggesting potential future research directions. We also extend our gratitude to Dr. Ayan Acharya for the valuable discussions and suggestions, which helped in improving the paper.

REFERENCES

- [1] A. Amini, W. Schwarting, A. Soleimany, and D. Rus. Deep evidential regression. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20*, article no. 1251, 11 pages. Curran Associates Inc., Red Hook, NY, USA, 2020. 1, 2
- [2] 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*, 27(2):1797–1807, Feb 2021. 2
- [3] 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. 2
- [4] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: two new techniques for image matching. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'77*, 5 pages, p. 659–663. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1977. 7
- [5] 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*, pp. 9368–9377, 2018. 1, 2
- [6] M. Berenkjoub, G. Chen, and T. Günther. Vortex boundary identification using convolutional neural network. In *2020 IEEE Visualization Conference (VIS)*, pp. 261–265, 2020. doi: 10.1109/VIS47514.2020.00059 2
- [7] M. Berger, J. Li, and J. A. Levine. A generative model for volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 25(4):1636–1650, 2019. 2
- [8] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015. 2
- [9] 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*, pp. 3–27. Springer London, London, 2014. 1, 2
- [10] K. Brodlie, R. Allendes Osorio, and A. Lopes. *Expanding the Frontiers of Visual Analytics and Visualization*, chap. A Review of Uncertainty in Data Visualization, pp. 81–109. Springer London, 2012. 2
- [11] C.-M. Chen, A. Biswas, and H.-W. Shen. Uncertainty modeling and error reduction for pathline computation in time-varying flow fields. In *2015 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 215–222, 2015. doi: 10.1109/PACIFICVIS.2015.7156380 2
- [12] J.-P. Chen, M. D. Hathaway, and G. P. Herrick. Prestall Behavior of a Transonic Axial Compressor Stage via Time-Accurate Numerical Simulation. *Journal of Turbomachinery*, 130(4):041014, 08 2008. doi: 10.1115/1.2812968 3
- [13] R. Crawfis. Tornado data set generator, 2003. 3
- [14] A. Damianou and N. D. Lawrence. Deep Gaussian processes. In *Proc. of AISTATS*, pp. 207–215, 2013. 1
- [15] S. Dutta, F. Nizar, A. Amaan, and A. Acharya. Visual analysis of prediction uncertainty in neural networks for deep image synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 2024. (In press). doi: 10.1109/TVCG.2024.3406959 2
- [16] G. Favelier, N. Faraj, B. Summa, and J. Tierny. Persistence atlas for critical point variability in ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):1152–1162, 2019. doi: 10.1109/TVCG.2018.2864432 5
- [17] F. Ferstl, K. Bürger, and R. Westermann. Streamline variability plots for characterizing the uncertainty in vector field ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):767–776, 2016. doi: 10.1109/TVCG.2015.2467204 2
- [18] F. Ferstl, K. Bürger, and R. Westermann. Streamline variability plots for characterizing the uncertainty in vector field ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):767–776, 2016. doi: 10.1109/TVCG.2015.2467204 4
- [19] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proc. of ICML*, pp. 1050–1059, 2016. 1, 2, 7, 8, 9
- [20] H. Gao, L. Sun, and J.-X. Wang. Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *Physics of Fluids*, 33(7):073603, 07 2021. doi: 10.1063/5.0054312 2
- [21] C. Gillmann, P. Arbelaez, J. T. Hernandez, H. Hagen, and T. Wischgoll. An uncertainty-aware visual system for image pre-processing. *Journal of Imaging*, 4(9), 2018. 2
- [22] E. Goan and C. Fookes. *Bayesian Neural Networks: An Introduction and Survey*, pp. 45–87. Springer International Publishing, Cham, 2020. doi: 10.1007/978-3-030-42553-1_3 2
- [23] P. Gu, J. Han, D. Z. Chen, and C. Wang. Reconstructing unsteady flow data from representative streamlines via diffusion and deep-learning-based denoising. *IEEE Computer Graphics and Applications*, 41(6):111–121, 2021. doi: 10.1109/MCG.2021.3089627 1
- [24] D. Günther, J. Salmon, and J. Tierny. Mandatory critical points of 2d uncertain scalar fields. In *Computer Graphics Forum*, vol. 33, pp. 31–40. Wiley Online Library, 2014. 5
- [25] T. Günther, M. Gross, and H. Theisel. Generic objective vortices for flow visualization. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 36(4):141:1–141:11, 2017. 3
- [26] L. Guo, S. Ye, J. Han, H. Zheng, H. Gao, D. Z. Chen, J.-X. Wang, and C. Wang. Ssr-vfd: Spatial super-resolution for vector field data analysis and visualization. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 71–80, 2020. doi: 10.1109/PacificVis48177.2020.8737 1
- [27] 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*, pp. 318–319, 2020. 1, 2
- [28] J. Han, J. Tao, H. Zheng, H. Guo, D. Z. Chen, and C. Wang. Flow field reduction via reconstructing vector data from 3-d streamlines using deep learning. *IEEE Computer Graphics and Applications*, 39(4):54–67, 2019. doi: 10.1109/MCG.2018.2881523 1, 2
- [29] J. Han and C. Wang. Tsr-tvd: Temporal super-resolution for time-varying data analysis and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):205–215, 2020. 2
- [30] J. Han and C. Wang. SSR-TVD: Spatial super-resolution for time-varying data analysis and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(6):2445–2456, 2022. 2
- [31] J. Han and C. Wang. Tsr-vfd: Generating temporal super-resolution for unsteady vector field data. *Computers & Graphics*, 103:168–179, 2022. doi: 10.1016/j.cag.2022.02.001 1
- [32] J. Han and C. Wang. Coordnet: Data generation and visualization generation for time-varying volumes via a coordinate-based neural network. *IEEE Transactions on Visualization and Computer Graphics*, 29(12):4951–4963, 2023. doi: 10.1109/TVCG.2022.3197203 1, 3
- [33] M. Han, J. Li, S. Sane, S. Gupta, B. Wang, S. Petruzza, and C. R. Johnson. Interactive visualization of time-varying flow fields using particle tracing neural networks. In *2024 IEEE 17th Pacific Visualization Conference (PacificVis)*, pp. 52–61, 2024. doi: 10.1109/PacificVis60374.2024.00015 1, 9
- [34] A. Hasanzadeh, E. Hajiramezani, K. Narayanan, N. Duffield, M. Zhou, and X. Qian. Semi-implicit graph variational auto-encoders. In *Proc. of NIPS*, pp. 10712–10723, 2019. 2
- [35] 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*, pp. 770–778, 2016. 3
- [36] 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*, 26(1):23–33, 2020. 2
- [37] 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. 2
- [38] F. Hong, C. Liu, and X. Yuan. Dnn-volvis: Interactive volume visualization supported by deep neural network. In *2019 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 282–291, 2019. 2
- [39] Z. Hu, J. Zhang, H. Wang, S. Liu, and S. Liang. Leveraging relational graph neural network for transductive model ensemble. In *Proc. of KDD*, p. 775–787, 2023. 1, 9
- [40] J. Jakob, M. Gross, and T. Günther. A fluid flow data set for machine

- learning and its application to neural flow map interpolation. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Scientific Visualization)*, 2020. 3
- [41] A. Kendall, V. Badrinarayanan, and R. Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*. BMVA Press, 2017. 3, 7, 8
- [42] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3
- [43] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pp. 6402–6413, 2017. 2, 3
- [44] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015. 1
- [45] X. Liang, S. Di, F. Cappello, M. Raj, C. Liu, K. Ono, Z. Chen, T. Peterka, and H. Guo. Toward feature-preserving vector field compression. *IEEE Transactions on Visualization and Computer Graphics*, 29(12):5434–5450, 2023. doi: 10.1109/TVCG.2022.3214821 5
- [46] C. Liu, R. Jiang, D. Wei, C. Yang, Y. Li, F. Wang, and X. Yuan. Deep learning approaches in flow visualization. *Advances in Aerodynamics*, 4(1):17, 2022. 2
- [47] 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)*, pp. 73–77, 2012. 2
- [48] Y. Lu, K. Jiang, J. A. Levine, and M. Berger. Compressive neural representations of volumetric scalar fields. *Computer Graphics Forum*, 40(3):135–146, 2021. doi: 10.1111/cgf.14295 1, 2, 3
- [49] Y. Ma, N. J. Foti, and E. B. Fox. Stochastic gradient MCMC methods for hidden Markov models. In *Proc. of ICML*, pp. 2265–2274, 2017. 2
- [50] OpenAI. Gpt-4 technical report, 2023. 9
- [51] M. Otto, T. Germer, H.-C. Hege, and H. Theisel. Uncertain 2d vector field topology. *Computer Graphics Forum*, 29(2):347–356, 2010. doi: 10.1111/j.1467-8659.2009.01604.x 2
- [52] 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. 2
- [53] A. T. Pang, C. M. Wittenbrink, S. K. Lodha, et al. Approaches to uncertainty visualization. *The Visual Computer*, 13(8):370–390, 1997. 2
- [54] A. Paszke et al. Pytorch: An imperative style, high-performance deep learning library. In *Proc. of NIPS*, pp. 8024–8035, 2019. 3
- [55] S. Popinet. Free computational fluid dynamics. *ClusterWorld*, 2(6), 2004. 3
- [56] S. Popinet, M. Smith, and C. Stevens. Experimental and numerical study of the turbulence characteristics of airflow around a research vessel. *Journal of Atmospheric and Oceanic Technology*, 21(10):1575–1589, 2004. doi: 10.1175/1520-0426(2004)021<1575:EANSOT>2.0.CO;2 4
- [57] K. Pöthkow and H.-C. Hege. Positional uncertainty of isocontours: Condition analysis and probabilistic measures. *IEEE Trans. on Vis. and Comp. Graphics*, 17:1393–1406, 2011. 2
- [58] K. Pöthkow, B. Weber, and H.-C. Hege. Probabilistic marching cubes. In *Proceedings of the 13th Eurographics / IEEE - VGTC Conference on Visualization, EuroVis’11*, 10 pages, pp. 931–940, 2011. 2
- [59] 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. 2
- [60] 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*, chap. From Quantification to Visualization: A Taxonomy of Uncertainty Visualization Approaches, pp. 226–249, 2012. 2
- [61] R. Rahaman and A. H. Thiery. Uncertainty quantification and deep ensembles. In *Neural Information Processing Systems*, 2020. 2
- [62] O. Sagi and L. Rokach. Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4):e1249, 2018. 2
- [63] S. Sahoo, Y. Lu, and M. Berger. Neural flow map reconstruction. *Computer Graphics Forum*, 41(3):391–402, 2022. doi: 10.1111/cgf.14549 1, 3, 9
- [64] M. Sensoy, L. Kaplan, and M. Kandemir. Evidential deep learning to quantify classification uncertainty. In *Advances in Neural Information Processing Systems*, pp. 3179–3189, 2018. 2
- [65] 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. 2
- [66] 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. 2
- [67] 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*, 28(6):2301–2313, 2022. 2
- [68] V. Sitzmann, J. N. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020. 3
- [69] K. Tang and C. Wang. Str-inr: Spatiotemporal super-resolution for multivariate time-varying volumetric data via implicit neural representation. *Computers & Graphics*, 119:103874, 2024. doi: 10.1016/j.cag.2024.01.001 1, 3
- [70] 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, pp. 8662–8674. Curran Associates, Inc., 2022. 1, 2
- [71] 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*, pp. 550–564, 2018. 2
- [72] B. Wang, P. Rosen, P. Skraba, H. Bhatia, and V. Pascucci. Visualizing robustness of critical points for 2d time-varying vector fields. *Computer Graphics Forum*, 32(3pt2):221–230, 2013. doi: 10.1111/cgf.12109 5
- [73] 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. 1, 2
- [74] 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*, 338:34–45, 2019. 2
- [75] 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*, 27(6):3064–3078, 2021. 2
- [76] S. Weiss, P. Hermüller, and R. Westermann. Fast neural representations for direct volume rendering. *Computer Graphics Forum*, 41(6):196–211, 2022. 2
- [77] S. Weiss, M. Işık, J. Thies, and R. Westermann. Learning adaptive sampling and reconstruction for volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(7):2654–2667, 2022. 2
- [78] 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*, 19(12):2713–2722, 2013. 2
- [79] 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. doi: 10.1109/TVCG.2022.3214420 2
- [80] T. B. L. Yi. Cnn-based flow field feature visualization method. *International Journal of Performability Engineering*, 14(3):434, 10 pages, 2018. doi: 10.23940/ijpe.18.03.p4.434444 2