Uncertainty Visualization of Critical Points of 2D Scalar Fields for Parametric and Nonparametric Probabilistic Models

Tushar M. Athawale (ID), Zhe Wang (ID), David Pugmire (ID), Kenneth Moreland (ID), Qian Gong (ID), Scott Klasky (ID), Chris R. Johnson (ID), and Paul Rosen (ID)

Fig. 1: Parametric (a-c) vs. nonparametric (d) models for uncertainty visualization of local minima of the magnitude of velocity field vectors in the Red Sea ensemble simulations [56] that correspond to eddy-like fe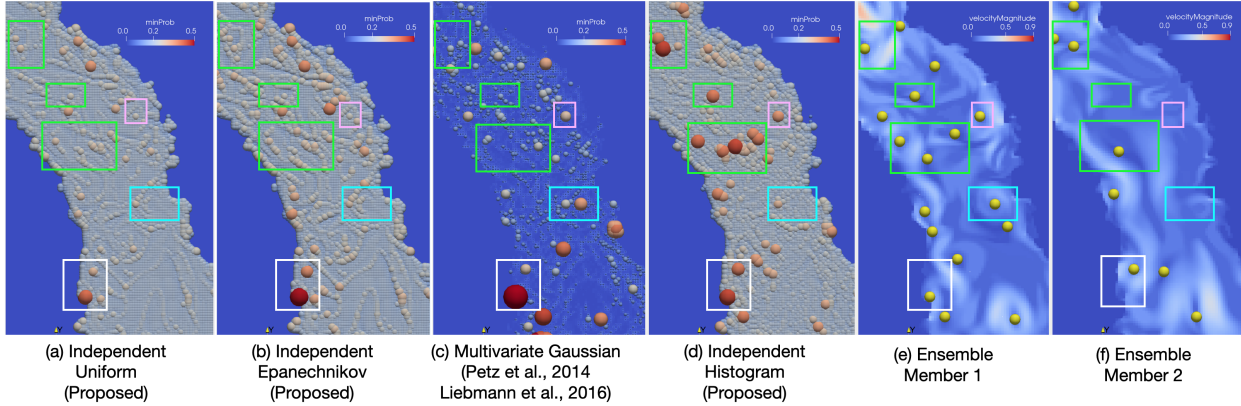atures. Images (e-f) show two randomly picked ensemble members with critical points rendered in yellow. The proposed nonparametric models show new high-probability features (larger spheres in red and orange) in green boxes that are not highlighted in parametric models. The results of nonparametric models can be trusted more because of the ability of nonparametric models to capture the shapes of multimodel and skewed distributions [8,53], unlike the restricted shape assumptions for the parametric models. The pink box shows a feature that is revealed by the multivariate Gaussian and histogram models but not by others. The cyan boxes show high-probability critical points in the multivariate Gaussian model that are not displayed as prominently in other models. The white boxes show two critical points that consistently have high probability across all models. All visualizations are computed efficiently and visualized in ParaView [2] using VTK-m [42] as a backend.

Abstract—This paper presents a novel end-to-end framework for closed-form computation and visualization of critical point uncertainty in 2D uncertain scalar fields. Critical points are fundamental topological descriptors used in the visualization and analysis of scalar fields. The uncertainty inherent in data (e.g., observational and experimental data, approximations in simulations, and compression), however, creates uncertainty regarding critical point positions. Uncertainty in critical point positions, therefore, cannot be ignored, given their impact on downstream data analysis tasks. In this work, we study uncertainty in critical points as a function of uncertainty in data modeled with probability distributions. Although Monte Carlo (MC) sampling techniques have been used in prior studies to quantify critical point uncertainty, they are often expensive and are infrequently used in production-quality visualization software. We, therefore, propose a new end-to-end framework to address these challenges that comprises a threefold contribution. First, we derive the critical point uncertainty in closed form, which is more accurate and efficient than the conventional MC sampling methods. Specifically, we provide the closed-form and semianalytical (a mix of closed-form and MC methods) solutions for parametric (e.g., uniform, Epanechnikov) and nonparametric models (e.g., histograms) with finite support. Second, we accelerate critical point probability computations using a parallel implementation with the VTK-m library, which is platform portable. Finally, we demonstrate the integration of our implementation with the ParaView software system to demonstrate near-real-time results for real datasets.

Index Terms—Topology, uncertainty, critical points, probabilistic analysis

◆

1 INTRODUCTION

Topological data analysis (TDA) is increasingly used in scientific visualizations because of its ability to concisely convey position and scale of important data features in complex datasets. The application of TDA can be found in diverse domains, including combustion science [12],

- *Tushar M. Athawale, Zhe Wang, David Pugmire, Kenneth Moreland, Qian Gong, and Scott Klasky are with the Oak Ridge National Laboratory. E-mail:{athawaletm, wangz, pugmire, morelandkd, gongq, klasky }@ornl.gov*
- *Chris R. Johnson and Paul Rosen are with the University of Utah E-mail: {crj, prosen}@sci.utah.edu*

molecular dynamics [43], and hydrodynamics [44]. Critical points are fundamental topological descriptors of scalar fields and form the basis of many topological visualization techniques, including persistent diagrams [17], contour trees [14], and Morse complexes [18,27]. Critical points denote the domain position where the gradient of a field vanishes (see technical details in Sec. 3). Uncertainty inherent in data arising from instrument/simulation/model errors [13], however, creates uncertainty regarding critical point positions. Ignoring uncertainty in critical

point positions, therefore, can lead to errors in topological visualizations and analysis. Thus, it is necessary to quantify and visually convey uncertainty in critical points to prevent misinformation. In this paper, we study uncertainty in critical points as a function of uncertainty in the underlying data modeled with probability distributions.

A significant number of studies have investigated uncertainty in critical points arising from uncertain scalar fields. Petz et al. [49] and Liebmann and Scheuermann [38] modeled noise in data with multivariate Gaussian distribution to derive critical point probabilities. The former work utilized Monte Carlo (MC) sampling of the original space, and the latter work utilized MC sampling of informative subspaces (referred to as patch sampling in their work) for deriving critical point probabilities. Günther et al. [25] investigated spatial bounds in the domain where at least one critical point is guaranteed to exist for the uniform noise assumption. Mihai and Westermann [41] derived confidence intervals for gradient field and Hessian to visualize likely critical point positions and their type in the domain. Vietinghoff et al. [61] derived the critical point probability using the Bayesian inference and derived confidence intervals [62]. Recently, Vietinghoff et al. developed a novel mathematical framework [63] that quantified uncertainty in critical points by analyzing the variation in manifestation of the same critical points occurring across realizations of the ensemble.

Inspired by these advances, we propose a novel closed-form theoretical framework for uncertainty quantification and visualization of critical point uncertainty. In particular, we analytically derive the critical point probability per grid position in a regular grid for independent parametric and nonparametric noise models with finite support. Although the previously proposed multivariate Gaussian noise models [38, 49] can better handle the noise correlation compared to independent noise models, they have two shortcomings. First, they resort to the MC sampling approach for deriving the critical point probability, which can be computationally expensive depending on the number of samples and grid resolution, and they converge slowly to the true answer. Second, because of the restricted shape assumptions for the (parametric) Gaussian distributions, they can be less robust to the data outliers than the nonparametric noise models [6, 53]. We address these two shortcomings by developing a closed-form formulation and algorithm that enhance the efficiency of results compared to MC sampling. Further, we showcase how the nonparametric models can provide more robust results compared to multivariate Gaussian and other parametric models.

Another important challenge associated with uncertainty visualization algorithms is the additional computational costs they bear, which prevents their integration with production-level software, e.g, VisIt [15] and ParaView [2]. The problem of the added cost is amplified for expensive MC sampling methods (e.g., in the case of multivariate Gaussian models). It is, therefore, important to research techniques that facilitate the integration of uncertainty visualization with production-level tools to make them usable and accessible to a broader scientific community. Recently, Wang et al. [65] provided a parallel and platform-portable implementation of isosurface uncertainty visualization using the VTK-m library [42]. They also showcased the integration of their VTK-m implementation with ParaView. The Topology Toolkit (TTK) by Tierny et al. [59] provided efficient implementation of mandatory critical points [25] that is usable in ParaView. Motivated by these works, we present a VTK-m parallel implementation of our closed-form critical point probability computations that is integrable with ParaView.

To summarize, our contributions are threefold. First, we propose a theoretical framework for uncertainty computation of critical points in uncertain 2D scalar fields. In particular, we propose an algorithm for deriving local minimum, local maximum, and saddle probability in closed form when data uncertainty is modeled as independent parametric (uniform, Epanechnikov) and nonparametric (histogram) distributions with finite support. Second, we evaluate our algorithms by demonstrating their enhanced accuracy and comparing their performance with the conventional MC models. We showcase the increased robustness of our proposed nonparametric models to data outliers compared to parametric models through results on a synthetic dataset. We present the utility of our methods through experiments on real datasets. Lastly, we implement our algorithms using the VTK-m library to present accelerated computation of critical point uncertainty and demonstrate their usability in ParaView for broader community access.

2 RELATED WORK

The research in uncertainty visualization dates back to early 2000 when Pang [47] and Johnson [32,33] recognized the need for quantifying and depicting uncertainty in visualization. Since then, multiple advances in uncertainty visualization have been documented in multiple survey reports, including those by Brodlie et al. [13], Potter et al. [52], and Kamal et al. [35]. Uncertainty visualization specific to ensemble data and topology was discussed in recent survey reports by Wang et al. [64] and Yan et al. [69], respectively.

Multiple new techniques have been derived to portray uncertainty in scalar, vector, tensor, and multivariate data. Uncertainty visualization of scalar field data covers a range of algorithms, including isosurfaces of univariate data [8, 21, 28, 51, 65], multivariate surfaces [5, 55], direct volume rendering [6, 16, 39, 57], topological merge trees [70], contour trees [68], persistence diagrams [60], and Morse complexes [7]. Although not to the extent of scalar fields, vector-field uncertainty has been explored to gain insight into important data features, including critical point uncertainty [45, 46], streamlines [20, 30], and Finite-Time Lyapunov Exponents [26]. A few techniques have been developed to compute and visualize uncertainty in tensor field data captured with high angular resolution diffusion imaging (HARDI) [31] and diffusion tensor imaging (DTI) [58]. These prior contributions mainly included Monte Carlo sampling, Bayesian statistics, closed-form solutions, empirical ensemble analysis, and low-dimensional embedding techniques to understand uncertainty in data features. The methods proposed in this paper model uncertain 2D scalar data as a probabilistic field and derive closed-form solutions to understand critical point uncertainty.

A variety of noise models have been previously explored to visualize uncertainty in scientific data. Statistically independent Gaussian [50] and uniform [4, 25] distributions have been used to model uncertainty in scalar fields and study their impact on features, such as level-sets and critical points. This work was later extended to multivariate Gaussian noise models [11, 28, 36, 38, 51] to capture the correlation among uncertain data and avoid overestimation of feature probabilities in the final visualization caused by the data independence assumption. Nonparametric models (e.g., histograms, kernel density estimation [48], Gaussian mixture models) have been used to show enhancements in visualization quality over parametric models for various visualization techniques, including level-sets [8, 53], direct volume rendering [6, 39], and fiber surfaces [5], because of their higher robustness to outliers. Nonparmaetric models, however, possess extra computational cost compared to parametric models. Recently, copula-based models have been explored to capture the correlation between both parametric and nonparametric models for uncertainty visualization [29]. In this paper, we present our methods by modeling data uncertainty with independent parametric and nonparametric noise models with finite support.

Effective presentation of uncertainty is another important research challenge. Visual attribute-mapping methods, e.g., colormapping [54] and point movement [24] proportional to uncertainty, have been proposed to convey uncertainty in 3D surfaces. We utilize the elevation map technique proposed by Petz et al. [49] to render critical point uncertainty for the climate dataset in Sec. 5. The novel use of glyphs has been previously proposed to convey uncertainty in vector [66] and tensor [34] field data. We utilize sphere glyphs to show critical point uncertainty for the Red Sea dataset in Fig. 1. Animation techniques have been proposed for uncertain volume rendering [40]. Effective quantification and visualization of uncertainty for 2D and high-dimensional data still remains a big challenge for the visualization community.

3 BACKGROUND AND PROBLEM SETTING

We briefly define critical points. Let $\mathbb{M} \subset \mathbb{R}^2$ be a 2D domain with a boundary discretized as a regular grid (we further ignore the boundary condition for most of our discussion). Let $f : \mathbb{M} \to \mathbb{R}$ be a function; ∇f denotes its gradient. A point $p \in \mathbb{M}$ is considered *critical* if $\nabla f(p) = 0$; otherwise it is *regular*. We will assume that all critical points are nondegenerate, i.e., f is a Morse function. A critical point is categorized

into three types, local minimum (l_{min}), local maximum (l_{max}), and saddle (l_s). In particular, if $f(p)$ is smaller than the function value of all of its neighbors, then the point p is a local minimum. Similarly, if $f(p)$ is greater than the function value of all of its neighbors, then the point p is a local maximum. If $f(p)$ is smaller than the function value of one neighbor and greater than the one for the next neighbor in alternating fashion with the neighbors visited sequentially in clockwise or counterclockwise manner, then the point p is a saddle.
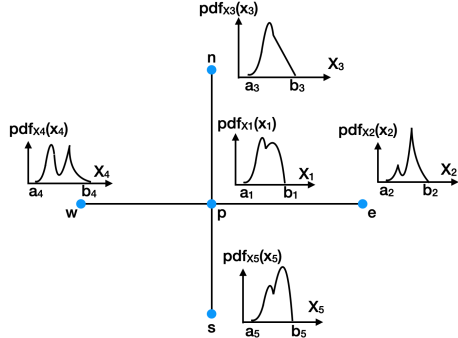

Fig. 2: Depiction of a problem setting. The probability distributions at a grid vertex p and its neighbors e, n, w, and s represent uncertainty in data. Our aim is to compute the probability of point p to be critical when distributions are represented with parametric and nonparametric models.

Uncertainty in data, however, creates uncertainty regarding whether a point is critical or not. In this paper, we calculate the probability for a domain point to be critical when uncertainty in data is modeled as probability distributions. Specifically, the methods in this paper consider uncertain data at four neighbors of a domain point along the two coordinate axes directions in a regular grid to compute the probability of the domain point to be critical. Although several applications consider six- or eight-pixel neighborhoods depending on the domain triangulation for critical point visualization, we plan to research these cases with a higher number of neighbors in the future.

Fig. 2 depicts the problem setting, which is also used to introduce notation. Let p be a point for which we want to compute the probability of it being critical. Let $X_1 \sim \text{Pdf}_{X_1}(x_1)$ denote a random variable with parametric or nonparametric noise distribution Pdf_{X_1} over the support $x_1 \in [a_1, b_1]$ at a point p. Let e, n, w, and s be the four neighbors of a point p in the east, north, west, and south directions with random variables $X_2 \sim \text{Pdf}_{X_2}(x_2)$, $X_3 \sim \text{Pdf}_{X_3}(x_3)$, $X_4 \sim \text{Pdf}_{X_4}(x_4)$, and $X_5 \sim \text{Pdf}_{X_5}(x_5)$, respectively, that denote uncertainty in data. For each random variable X_i, $x_i \in [a_i, b_i]$ with $a_i < b_i$. Our work presents all derivations for the independent noise models with noise distribution over a finite support, i.e., the random variables X_i with $i \in \{1, \ldots, 5\}$ are assumed to be independent and bounded by a finite support $[a_i, b_i]$. As the local data are not always independent in real datasets and identifying bounds $[a_i, b_i]$ can be challenging considering noisy data acquisition processes [13], we discuss the ramifications of our independent noise and finite support assumptions in Sec. 6. Because of the data independence assumption, the joint probability density Pr_{joint} of the random variables is the product of their individual probability densities, i.e., $\text{Pdf}_{joint} = \prod_{i=1}^{i=5} \text{Pdf}_{X_i}(x_i)$. Let $dx = \prod_{i=1}^{i=5} dx_i$. Given these data, our goal is to find the probability of point p being a local minimum $\text{Pr}(p = l_{min})$, local maximum $\text{Pr}(p = l_{max})$, and saddle $\text{Pr}(p = l_s)$.

4 METHODS

We describe the mathematical formulation and our algorithm for critical point probability computation in closed form for independent parametric and nonparametric noise models with finite support.

4.1 Critical Point Probability (Two-Pixel Neighborhood)

For simplicity, we describe our derivations and approach for computing the probability of point p to be critical for 1D uncertain scalar fields. Our methods for the 1D case generalize to uncertain 2D scalar fields, as described in Sec. 4.2. For the 1D case, we consider only the two neighbors with random variables X_2 and X_3 of a 1D point p with its

associated random variable X_1. The rest of the problem settings are similar to the 2D case described earlier in Sec. 3.

4.1.1 Local Minimum Probability

The probability of point p being a local minimum, $\text{Pr}(p = l_{min})$, can be computed by integrating the joint probability Pr_{joint} over its support where random variable X_1 is simultaneously smaller than all neighboring random variables (i.e., X_2 and X_3). Mathematically, the local minimum probability can be represented as follows:

$$\text{Pr}(p = l_{min}) = \text{Pr}[(X_1 < X_2) \text{ and } (X_1 < X_3)]$$
$$= \int_{x_1=a_1}^{x_1=b_{min}} \int_{x_2=max(x_1,a_2)}^{x_2=b_2} \int_{x_3=max(x_1,a_3)}^{x_3=b_3} (\text{Pdf}_{joint}) \, dx, \quad (1)$$
$$\text{where } b_{min} = min(b_1, b_2, b_3)$$

Equation (1) represents the core integration formula for the computation of local minimum probability at a domain position p. We describe our approach for the computation of local minimum probability in three parts. (1) We explain the integral limits and piecewise simplifications of the core integration formula in Eq. (1). (2) We describe our piecewise integration approach to efficiently compute the formula in Eq. (1). (3) We show a running illustration of our piecewise integration approach to compute the local minimum probability.

Limits a_1 and b_{min} of the outer integral in Eq. (1): The outer integral of Eq. (1) indicates the portion of data range of a random variable X_1 (i.e., $[a_1, b_1]$) that can result in point p being a local minimum. In particular, the portion $[a_1, b_{min}]$ (with $a_1 < b_{min}$) of random variable X_1 can result in point p being a local minimum, where b_{min} denotes the minimum among b_1, b_2, and b_3. In contrast, the data range $[b_{min}, b_1]$ for $b_{min} \neq b_1$ cannot result in a point p as a local minimum (l_{min}) because it will be always greater than the random variables X_2 or X_3 depending on if $b_{min} = b_2$ or $b_{min} = b_3$, respectively. Thus, mathematically, for any value $x_1 \geq b_{min}$, $\text{Pr}(p = l_{min}) = 0$. In the case $b_{min} < a_1$, then $Pr(p = l_{min}) = 0$ because there is at least one random variable between X_2 and X_3 that will be always smaller than X_1.

Limits $max(x_1, a_i)$ and b_i of the inner integrals in Eq. (1): The two inner integrals in Eq. (1) integrate the joint distribution Pdf_{joint} over its support where random variables X_2 and X_3 are simultaneously greater than $x_1 \in [a_1, b_{min}]$ in the outer integral. The inner integral lower limits are $max(x_1, a_i)$ for $i \in \{2, 3\}$. The maximum is taken because the support of a random variable X_i is restricted to $[a_i, b_i]$. Thus, for $x_1 < a_i$ in any inner integral, the entire support $[a_i, b_i]$ with $i \in \{2, 3\}$ will always be greater than x_1, and the inner integral does not depend on the value of x_1. In contrast, for $x_1 > a_i$, the inner integration depends on the value of x_1 because x_1 assumes values in support of distributions. It is guaranteed that the upper limit of inner integrals in Eq. (1) is greater than their respective lower limit, i.e., $b_i \geq max(x_1, a_i)$, for two reasons. First, for any random variable X_i, we assume $a_i < b_i$. Second, the maximum value of x_1 is equal to $b_{min} = min(b_1, b_2, b_3)$ based on the outer integral (see the previous paragraph), and it cannot exceed the upper limits b_2 and b_3 in inner integrals. Depending upon whether the $max(x_1, a_i)$ is equal to x_1 or a_i, the integral in Eq. (1) can be simplified and computed differently, which necessitates the evaluation of the integral in Eq. (1) in a piecewise manner, as described next.

Piecewise simplification of Eq. (1): The core integration formula in Eq. (1) can be simplified differently for different subsets of the range of the outer integral (i.e., $[a_1, b_{min}]$). We refer to each subset as a piece P. For a piece P, where $x_1 < a_2$ and $x_1 < a_3$, the inner integrals in Eq. (1) attain the range $x_2 \in [a_2, b_2]$ and $x_3 \in [a_3, b_3]$. In other words, the inner integrals do not depend on x_1 when $x_1 < a_2$ and $x_1 < a_3$. Thus, the integration over the entire support of random variables X_2 and X_3 simplifies Eq. (1) to the integration over a marginal distribution of X_1 for the piece P, i.e., $\int_{x_1 \in P} \text{Pdf}_{X_1}(x_1) \, dx_1$.

For a piece P, where $x_1 \in [a_2, b_2]$ and $x_1 < a_3$, the inner integrals in Eq. (1) attain the range $x_2 \in [x_1, b_2]$ and $x_3 \in [a_3, b_3]$. In this case, only the first inner integral related to the range of random

Upper limits and pdfs depend on the intervals starting before id_{a1}, i.e., $[a_2,b_2]$ in a given example

Each new piece P_i adds a new inner integral (here new piece is $P_2=[a_3,b_{min}=b_2]$)

$$I_{P_1} = \int_{x_1=a_1}^{x_1=a_3} \int_{x_2=x_1}^{x_2=b_2} \mathbf{pdf}_{X_1}(x_1)\mathbf{pdf}_{X_2}(x_2)\, \mathbf{dx_1 dx_2}$$

$$I_{P_2} = \int_{x_1=b_{min}}^{x_1=b_{min}} \int_{x_2=x_1}^{x_2=b_2} \int_{x_3=x_1}^{x_3=b_3} \mathbf{pdf}_{X_1}(x_1)\mathbf{pdf}_{X_2}(x_2)\mathbf{pdf}_{X_3}(x_3)\, \mathbf{dx_1 dx_2 dx_3}$$

(a) <u>Task 1</u>: Determine pieces P_i of the outer integral in Eq. (1)

(b) <u>Task 2</u>: Determine integral over a piece $P_1=[a_1,a_3]$

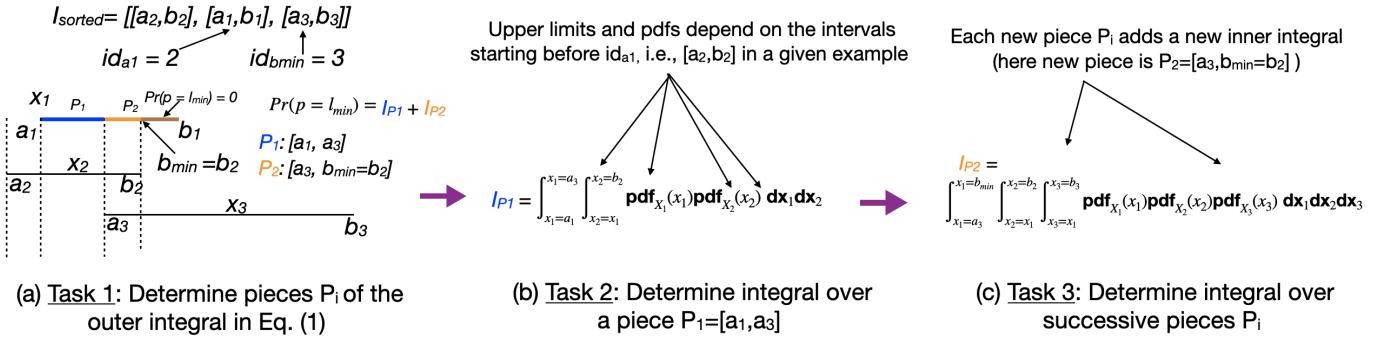(c) <u>Task 3</u>: Determine integral over successive pieces P_i

Fig. 3: Illustration of our three-step approach for critical point uncertainty computation for the 1D case. (1) Determine the range of random variable X_1 for which critical point probability is nonzero (i.e., $[a_1, b_{min} = b_2]$) and determine its pieces. The range for which critical point probability is zero is shown in brown in (a). Each new start point $a_i \in [a_1, b_{min}]$ creates a new piece. Here, a_3 results in two pieces $P_1 = [a_1, a_3]$ and $P_2 = [a_3, b_{min}]$. (2) Compute integration for piece P_1 (i.e., I_{P_1}) depending on which intervals overlap it. Since the interval $[a_2, b_2]$ overlaps with P_1, I_{P_1} corresponds to integral over a joint distribution of X_1 and X_2, as depicted in (b). (3) Update integrals for next pieces depending on observed start points (e.g., inclusion of random variable X_3 in the integral I_{P_2} in (c) based on the start point a_3) and sum all piecewise integrals to compute the local minimum probability.

variable X_2 depends upon x_1. Thus, Eq. (1) simplifies as the integration over the joint distribution of X_1 and X_2 for the piece P, i.e., $\int_{x_1 \in P} \text{Pdf}_{X_1}(x_1)\text{Pdf}_{X_2}(x_2)\,dx_2\,dx_1$. In summary, various pieces of the outer integral in Eq. (1) can be simplified differently based on whether the inner integrals depend on x_1 or not.

Approach for computing local minimum probability: The computation of Eq. (1) depends on the ordering of start points a_i (i.e., $a_1, a_2,$ and a_3) and b_{min}. Thus, in 1D case, there are $4! = 24$ permutations of a_i and b_{min}. These permutations increase fast for 2D/high-dimensional cases. We, therefore, devise an efficient algorithm that computes the piecewise integrals on the fly depending on the observed permutation of a_i and b_{min} without needing to go through all permutations.

We now describe our approach for the computation of the local minimum probability at a domain position p [i.e., $\Pr(p = l_{min})$]. Our approach comprises three main tasks. **Task 1: Determination of pieces P_i of the outer integral in Eq. (1) needed for performing piecewise integration.** Initially, we compute b_{min}. If $b_{min} < a_1$, then there are no pieces and $\Pr(p = l_{min}) = 0$. If $b_{min} > a_1$, then we sort intervals representing uncertain data ranges (i.e., $[a_i, b_i]$) based on their start points a_i and keep them in the array named I_{sorted}. We note the index of the interval $[a_1, b_1]$ in I_{sorted} (referred to as id_{a_1}) and the index of interval in I_{sorted} from the end that contains b_{min} (referred to as $id_{b_{min}}$), as a_1 and b_{min} constitute limits of the outer integral in Eq. (1). Any start points a_i contained in the range of indices id_{a_1} and $id_{b_{min}}$ determine the pieces P_i for integration. This process generalizes to any ordering of a_i to determine the pieces p_i of the outer integral in Eq. (1).

Next, we compute the integration for piece P_1 denoted as I_{P_1}. **Task 2: Integration over piece P_1 of the outer integral range $[a_1, b_{min}]$.** The integration for piece P_1 depends on the intervals that started before a_1 because the inner integral in Eq. (1) depends on x_1 for a random variable X_i (with $i \in 2, 3$) started before a_1, as $max(x_1, a_i)$ is equal to x_1. Task 2, therefore, corresponds to finding the intervals that started before id_{a_1}, which also determines the upper limits of inner integrals for piece P_1 depending on observed order of intervals.

The computation of the integral in Eq. (1) for piece P_1 (as well as any arbitrary piece P_i) simplifies to one of the three types of integration formulae, which we call integration templates. Generally, the simplification depends on the number of intervals overlapping a piece P_i, as explained earlier. If there is no overlap with a piece, then the integral in Eq. (1) simplifies to the integration of the probability distribution of random variable X_1 (Template 1) over a piece. If only one random variable (X_2 or X_3) is overlapping with a piece, then the integral in Eq. (1) simplifies to the integral over the joint distribution of X_1 and a random variable corresponding to the overlapping interval (Template 2). If both random variables X_2 and X_3 are overlapping with a piece, then the integral in Eq. (1) corresponds to the integral over the joint distribution of all random variables (Template 3).

Having determined the integration for piece P_1, we compute integration for successive pieces. **Task 3: Integration over piece P_i with**

$i > 1$. Essentially, each new start point a_i observed between the outer integral limits a_1 and b_{min} of Eq. (1) creates a new piece. Generally speaking, each new start point a_i results in a different simplification of Eq. (1) because $max(x_1, a_i)$ in Eq. (1) becomes equal to x_1 at each new start point. Thus, encountering a new start point adds one inner integral in a simplified form compared to the piece before encountering a new start point. Finally, the integration of all pieces is summed to compute the local minimum probability at a point p, i.e., $\Pr(p = l_{min})$.

Illustration of Local Minimum Probability Computation: Figure 3 illustrates our method for computing the local minimum probability for a domain point p. As shown for the example in Fig. 3, $a_2 < a_1 < a_3 < b_2 < b_1 < b_3$. Initially, we determine the range of a random variable X_1 that can result in point p being a local minimum. As shown in Fig. 3a, each value in the range $[x_1 = a_1, x_1 = (b_{min} = b_2)]$ has a nonzero probability of being simultaneously smaller than the neighboring random variables (i.e., X_2 and X_3). On the contrary, the range $[x_1 = (b_{min} = b_2), x_1 = b_1]$ is always greater than the random variable X_2, and therefore, cannot result in point p as a local minimum.

In Task 1, we determine the pieces P_i of the range $[x_1 = a_1, x_1 = (b_{min} = b_2)]$. As depicted in Fig. 3, the array I_{sorted} has intervals ordered by a_i, where $a_2 < a_1 < a_3$. For this I_{sorted}, $id_{a_1} = 2$ and $id_{b_{min}} = 3$. Since a_3 is a start point in interval indexed by $id_{b_{min}}$, it divides the outer integral range $[a_1, b_{min}]$ in Eq. (1) into two pieces (depicted in blue and orange in Fig. 3).

In Task 2, we determine the simplification of the formula in Eq. (1) for piece P_1. The simplification for piece $P_1 = [a_1, a_3]$ in Fig. 3 (denoted by blue) depends on the number of intervals that started before a_1. As observed in Fig. 3 (and I_{sorted}), the interval $[a_2, b_2]$ starts before a_1. Since $x_1 < a_3$ in piece P_1, the formula in Eq. (1) integrates random variable X_3 over its entire support and simplifies to the integration over the joint distribution of random variables X_1 and X_2 shown in Fig. 3b.

In Task 3, we determine the simplification of the formula in Eq. (1) for successive pieces formed by each new start point $a_i \in [a_1, b_{min}]$. In Fig. 3, the start point $a_3 \in [a_1, b_{min}]$ results in a new piece P_2 shown in orange. All inner integrals for piece P_2 stay the same as piece P_1 except for one newly added inner integral with limits $[x_1, b_3]$, as shown in Fig. 3c, because $x_1 = max(x_1, a_3)$ for piece P_2, unlike the piece P_1 in which $a_3 = max(x_1, a_3)$. Thus, we make such updates to inner integrals for each new piece corresponding to a new start point $a_i \in [a_1, b_{min}]$.

Time complexity: Our presented approach for local minimum probability computation initially sorts all the intervals based on start points a_i with $i \in 1, 2, 3$ and b_{min} to determine pieces for integration (Task 1), which is a constant time operation. Task 2 and Task 3 comprise a single loop, which runs a maximum of three times corresponding to the three entries of a sorted interval array I_{sorted}. Each loop iteration computes the integral template (simplification of the formula in Eq. (1)) on the fly depending on the observed data in array I_{sorted} in constant time. The algorithm is, therefore, linear time complexity with the number of input intervals (here, three) and extremely efficient.

4.1.2 Local Maximum Probability

Having derived a probabilistic framework for computation of local minimum probability (Sec. 4.1.1), the computation of the local maximum probability $Pr(p = l_{max})$ at a domain point p is fairly straightforward. Computation of the local maximum probability corresponds to computing $Pr[(X_1 > X_2)$ and $(X_1 > X_3)]$ for the 1D case, which is equivalent to computing $Pr[(-X_1 < -X_2)$ and $(-X_1 < -X_3)]$. This negation format is equivalent to Eq. (1). Thus, we create negated random variables $X_1' = -X_1$, $X_2' = -X_2$, and $X_3' = -X_3$. We then apply our proposed local minimum probability computation algorithm (Sec. 4.1.1) to these new random variables X_i' for computing the local maximum probability.

4.1.3 Saddle Probability

The probability of point p being a saddle, $Pr(p = l_s)$, can be computed by integrating the joint probability Pdf_{joint} over its support where the random variable X_1 is simultaneously smaller than X_2 and greater than X_3 (and the other way around). Mathematically, the saddle probability can be represented as follows:

$$Pr(p = l_s) = (t_1 = Pr[(X_1 < X_2) \text{ and } (X_1 > X_3)]) + \\ (t_2 = Pr[(X_1 > X_2) \text{ and } (X_1 < X_3)])$$

The term t_1 in the equation above can be written as follows:

$$t_1 = Pr[(X_1 < X_2) \text{ and } (X_1 > X_3)]$$
$$= \int_{x_1=a_{max}^{alt}}^{x_1=b_{min}^{alt}} \int_{x_2=max(x_1,a_2)}^{x_2=b_2} \int_{x_3=a_3}^{x_3=min(x_1,b_3)} (Pdf_{joint}) \, dx, \quad (2)$$
$$\text{where } a_{max}^{alt} = max(a_1, a_3), b_{min}^{alt} = min(b_1, b_2)$$

Equation (2) represents the core integration formula for the computation of the saddle probability at a domain point p. We derive our closed-form computations and algorithm only for the term $t_1 = Pr[(X_1 < X_2)$ and $(X_1 > X_3)]$. The term t_2 can be computed by creating negated random variables $X_i' = -X_i$ with $i \in \{1, 2, 3\}$ and plugging X_i' in place of X_i into the derivation for term t_1.

The algorithm to compute the saddle probability is similar to the three tasks of the local minimum probability computation in Sec. 4.1.1. In Task 1, the algorithm first determines the data range $[a_{max}^{alt}, b_{min}^{alt}]$ that qualifies for point p being a saddle (similar to the range determination for local minimum probability computation, illustrated in Fig. 3a). The algorithm then divides this range into pieces P_i depending on the ordering of points $[a_{max}^{alt}, b_{min}^{alt}, a_2, b_3]$ that correspond to the limits of integrals in Eq. (2). In Task 2, the integral of piece P_1 is computed depending on how a_2 and b_3 are ordered with respect to the lower limit a_{max}^{alt}. In Task 3, the integrals of successive pieces are computed depending on the order in which a_2 and b_3 appear until the upper limit b_{min}^{alt}. All integrals fit into one of the templates that represent simplification of the core integration in Eq. (2). All piecewise integrals are finally added to compute the term t_1.

4.2 Critical Point Probability (Four-Pixel Neighborhood)

Our methods for the two-pixel neighborhood case (Sec. 4.1) generalize to the four-pixel neighborhood case (depicted in Fig. 2) straightforwardly. Here, we briefly discuss a few new specifics related to the four-pixel neighborhood case. The detailed algorithms and illustrations are provided in the supplementary material.

Local Minimum Probability: The core integration formula for the computation of local minimum probability at a domain position p in the case of a four-pixel neighborhood takes a form similar to Eq. (1).

$$Pr(p = l_{min})$$
$$= Pr[(X_1 < X_2) \text{ and } (X_1 < X_3) \text{ and } (X_1 < X_4) \text{ and } (X_1 < X_5)]$$
$$= \int_{x_1=a_1}^{x_1=b_{min}} \int_{x_2=max(x_1,a_2)}^{x_2=b_2} \cdots \int_{x_5=max(x_1,a_5)}^{x_5=b_5} (Pdf_{joint}) \, dx, \quad (3)$$
$$\text{where } b_{min} = min(b_1, b_2, b_3, b_4, b_5)$$

Computation of the integral in Eq. (3) has a workflow similar to the two-pixel neighborhood case. Initially, pieces of the range $[a_1, b_{min} = min(b_1 \ldots b_5)]$ are determined based on the ordering of integral limits, i.e., a_1, a_2, a_3, a_4, a_5, and b_{min}. Based on these six points of interest, there are $6! = 720$ possible permutations. We compute piecewise simplification of Eq. (3) (templates) on the fly depending on the observed ordering of the six points without needing to go through all permutations. Finally, all piecewise integrals are summed to compute the local minimum probability.

Local Maximum Probability: Similar to the two-pixel neighborhood case, the local maximum probability in the four-pixel neighborhood case is computed by negating the random variables followed by the application of our local minimum probability computation algorithm to the negated random variables.

Saddle Probability The core integration formula for computing probability of point p being a saddle, $Pr(p = l_s)$, in the case of four-pixel neighborhood is mathematically represented as follows:

$$Pr(p = l_s)$$
$$= (t_1 = Pr[(X_1 < X_2) \text{ and } (X_1 > X_3) \text{ and } (X_1 < X_4) \text{ and } (X_1 > X_5)]) + \\ (t_2 = Pr[(X_1 > X_2) \text{ and } (X_1 < X_3) \text{ and } (X_1 > X_4) \text{ and } (X_1 < X_5)])$$

The term t_1 in the equation above takes a form similar to the two-pixel neighborhood case in Eq. (2) and can be written as follows:

$$t_1 = Pr[(X_1 < X_2) \text{ and } (X_1 > X_3) \text{ and } (X_1 < X_4) \text{ and } (X_1 > X_5)]$$
$$= \int_{x_1=a_{max}^{alt}}^{x_1=b_{min}^{alt}} \int_{x_2=max(x_1,a_2)}^{x_2=b_2} \int_{x_3=a_3}^{x_3=min(x_1,b_3)} \int_{x_4=max(x_1,a_4)}^{x_4=b_4} \int_{x_5=a_5}^{x_5=min(x_1,b_5)} \cdots$$
$$(Pdf_{joint}) \, dx, \text{ where } a_{max}^{alt} = max(a_1, a_3, a_5), b_{min}^{alt} = min(b_1, b_2, b_4)$$
$$\quad (4)$$

Again, the saddle probability can be efficiently computed using piecewise integration with piece limits determined by ordering of the integral limits $a_{max}^{alt}, b_{min}^{alt}, a_2, a_4, b_3, b_5$ in Eq. (4). We compute piecewise simplification of Eq. (4) (templates) on the fly depending on the observed ordering of the six points and sum them up to compute the saddle probability, similar to the two-pixel neighborhood case.

4.3 Parametric Noise Models

In this section, we derive critical point probability computations for the uniform and Epanechnikov distributions. The derivation for the uniform noise model acts as a building block for our histogram model derivation in Sec. 4.4.1. The uniform distribution is given by $Pdf_X(x) = \frac{1}{b-a}$, where $x \in [a, b]$. Although we do not have a closed-form solution for the Gaussian noise model, similar to the Gaussian, the Epanechnikov model gives more weight to the data mean, has a bell-like shape, and is smoother than the uniform noise distribution (see Fig. 4). The Epanechnikov distribution, therefore, can yield better results than the uniform noise model, as presented later in Sec. 5.2. The Epanechnikov distribution is given by $Pdf_X(x) = \frac{3}{2*(b-a)}[1 - (\frac{x-m}{w})^2]$, where $m = (a+b)/2$, $w = (b-a)/2$, and $x \in [a, b]$. These distribution formulae can be plugged into the integration formulae for critical point probability computations (i.e., Eq. (3) and Eq. (4)) and their simplifications (templates) to compute results for these two types of distributions.
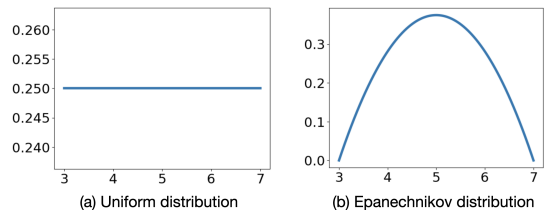

Fig. 4: Epanechnikov distribution (b) gives more weight to the mean unlike the uniform distribution (a), and hence, can provide enhanced visualization compared to the uniform noise model.

The derivation of integral templates for the uniform and Epanechnikov kernel can be cumbersome because of high-order functions resulting from integrals. For example, since the Epanechnikov is an order-2 kernel, the integration templates can result in formulae of order-15 functions because of the five integrals in the case of four-pixel neighborhood without simplification (Sec. 4.2). Thus, we use the Wolfram Alpha software [67] for deriving the integral templates[1]. Note that these high-order functions may produce numerical instabilities for large or fractional data values. Thus, the dataset range needs to be properly scaled to ensure stable computations with the Epanechnikov kernel.

4.4 Nonparametric Noise Models

For nonparametric models, we do not assume a particular distribution type for data. Instead, we derive a histogram with a user-specified number of bins at each grid vertex to derive critical point probabilities. Our derivations for the parametric models act as building blocks for derivations of nonparametric models. We propose closed-form and semianalytical (mix of MC and closed-form) solutions.

4.4.1 Closed-Form Formulation

Each histogram can be considered as a weighted combination of nonoverlapping uniform distributions. Mathematically, the $\text{pdf}_X(x)$ using a histogram can be written as $\text{pdf}_X(x) = w_i \sum_{i=1}^{i=h} K_b(x - x_i)$, where $\sum_{i=1}^{i=h} w_i = 1$, K_b denotes a uniform kernel K with width b, x_i denotes the bin center, and h denotes the number of histogram bins, and $x \in [x_1 - b/2, x_h + b/2]$. Thus, our derivation and algorithm for the uniform distribution (Sec. 4.3) can be leveraged for critical point probability computation with the histogram noise model. Let K_{b_i}, K_{b_j}, K_{b_k}, K_{b_l}, and K_{b_m} denote the uniform kernels of histograms for random variables X_1, X_2, X_3, X_4, and X_5, respectively. The critical point probability can be computed by going through all combinations of the uniform kernels across five random variables and summing the critical point probability for each possible combination weighted by its probability. Mathematically, the local minimum probability at a grid point p can be computed as follows:

$$\Pr(p = l_{min}) = w \sum_{i=1}^{i=h} \cdots \sum_{m=1}^{m=h} \Pr(p = l_{min})_{i,j,k,l,m}, \qquad (5)$$

where $\Pr(p = l_{min})_{i,j,k,l,m}$ denotes the probability of each kernel K_{b_i} of random variable X_1 being simultaneously smaller than kernels $K_{b_j}, \ldots, K_{b_m}$ of random variables $X_2, \ldots, X_5$, respectively, weighted by the probability of choosing kernels denoted by $w = w_i w_j w_k w_l w_m$. We verify the correctness of our formulation through quantitative and qualitative evaluation in our results.

Nonparametric models can increase the robustness of visualization to outliers [6, 8, 53] because they do not assume any particular shape of the distribution. We show the increased robustness of visualizations under uncertainty with the proposed nonparametric models compared to parametric models in the results, Sec. 5. However, the increased quality of nonparametric models comes at the cost of an increased number of computations. The time complexity of computing the local minimum probability is $O(h^5)$, as observed from Eq. (5), where h is the number of histogram bins. Our histograms method in Eq. (5) can be extended to a more general kernel density estimation [48], in which each noise sample is assigned a kernel. However, the time complexity can grow sharply with an increase in the sample count, and KDE can quickly become impractical for use in visualization. Thus, we restrict our nonparametric methods to histograms with a user-specified number of bins. To accelerate the performance of nonparametric methods, we propose two solutions. First, we present a more efficient semianalytical approach (Sec. 4.4.2), which provides an approximate but reliable solution at a greater speed. Second, we accelerate our critical point uncertainty computation using a parallel implementation (Sec. 4.5).

---

[1] All integral templates and code are available at `https://github.com/tusharathawale/UCV/tree/exp_critical_point_noplugin`.

4.4.2 Semianalytical Solution

We propose a mix of MC and closed-form formulation to get approximate but reliable solutions at a greater speed. For our method, we draw c samples from $\text{pdf}_{X_1}(x_1)$ at a point p. For each sample, the probability of that sample being critical can be computed in closed form. Specifically, we compute the probability of a sample being smaller (or greater) than neighboring random variables in closed form. For example, in order to compute $\Pr(\text{sample} < X_2)$, we first find the histogram bin of $\text{Pdf}_{X_2}(x_2)$ to which the sample belongs. We then integrate the bin density for values greater than the sample (similar to a uniform distribution) and sum it with the integration of densities of subsequent bins. Note that integration of subsequent bins is precomputed efficiently using the prefix sum (scan) method and, therefore, the integration computation time does not depend on the number of bins. Let $\Pr_2, \ldots, \Pr_5$ denote $\Pr(\text{sample} < X_2), \ldots, \Pr(\text{sample} < X_5)$, respectively. Because of the independent assumption, the local minimum probability for the sample corresponds to a product $\Pr_{sample} = \Pr_2 \Pr_3 \Pr_4 \Pr_5$. Saddle and local maximum probabilities for the sample can be computed in a similar manner. Let $\Pr_{total}$ denote the sum of $\Pr_{sample}$ for all c samples. Thus, the probability of the point p being critical can be found as the ratio of $\Pr_{total}$ and c. The computational complexity of this semianalytical method is proportional to the number of samples (i.e., c) drawn from a single distribution $\text{Pdf}_{X_1}(x_1)$. This method, therefore, provides faster results compared to the exponential closed-form solution in Sec. 4.4.1.

4.5 Integration with VTK-m and ParaView

Motivated by Wang et al.'s work [65], we integrate the critical point uncertainty code with the VTK-m software [42]. Since the computation of critical point probability depends only on the local neighbors and is independent of data at other pixels, it is embarrassingly parallel. We therefore implemented our code using VTK-m. The advantage of VTK-m is that it allows optimized access to neighbors, and it is platform portable. With our VTK-m implementation, we showcase significant speed-up in critical point uncertainty computation on various architectures, including AMD, NVIDIA, and Intel processors. Further, we export our VTK-m code as a plugin for the use in ParaView software. With our plugin, the uncertainty of critical points can be visualized in ParaView in near-real time and can be combined with the other ParaView filters for better analysis of uncertainty.

5 RESULTS AND DISCUSSIONS

5.1 Validation and Performance of Proposed Algorithms

We validate the correctness of our proposed closed-form computations and algorithms through qualitative and quantitative comparison with respect to the conventional MC sampling approach. We also demonstrate the performance improvements of the proposed methods over MC sampling. Figure 5 demonstrates the correctness and enhanced performance of our algorithms through experiments on a synthetic Ackley function [1] sampled on a uniform grid. In particular, the uncertain data is synthetically generated by injecting random uniform noise at each grid position to produce an ensemble of 50 members. At each grid vertex, the minimum and maximum values are computed from the ensemble to estimate the range of a uniform distribution. The local minimum probability $Pr(p = l_{min})$ and saddle probability $Pr(p = l_s)$ are then computed at each grid vertex p using the MC sampling method and our proposed algorithms. The results are visualized in Fig. 5.

Columns Fig. 5a-b show the results of MC sampling. Column Fig. 5c shows the results obtained with our closed-form formulation and algorithms. Column Fig. 5d visualizes the difference image between column Fig. 5c and column Fig. 5b. Column Fig. 5e visualizes the convergence of MC solutions to our closed-form solution by plotting the root mean squared error (RMSE) between the MC and closed-form solutions. The white isocontours (isovalues 0.1 and 0.01 for local minimum and saddle probability results, respectively) shown in Fig. 5a-c enclose the possible critical point positions. As observed in Fig. 5a-c, the isocontour structure in MC results converges to our closed-form solutions as we increase the number of MC samples from 100 to 2000.
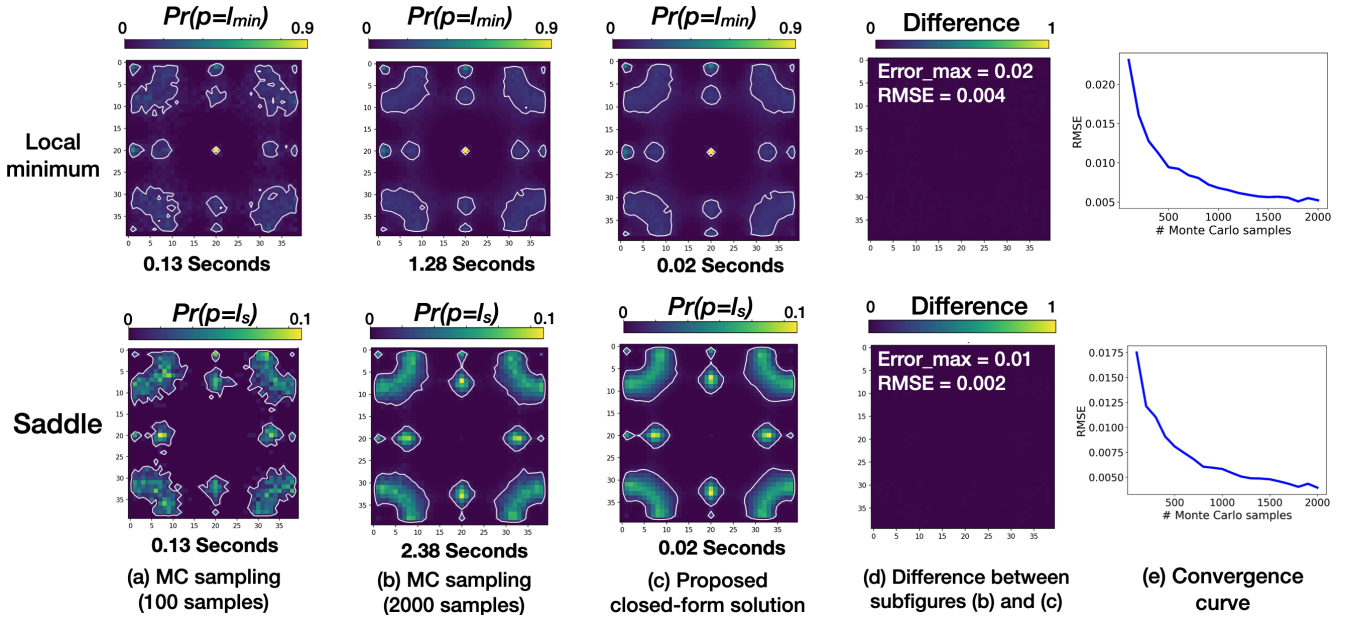
Fig. 5: The qualitative and quantitative proof of correctness and enhanced performance of our proposed closed-form computations (column c) with the MC sampling approach (columns a and b) as the baseline. The results are shown for the uniform noise model. The solution obtained with 2000 MC samples converges to closed-form computations with our algorithms (see the difference images in column d and convergence curves in column e), thereby confirming their correctness. Our methods provide $64\times$ and $119\times$ speed-up with respect to the MC sampling approach with 2000 samples.

In MC sampling, uniform distributions are sampled to estimate the local minimum/saddle probability. As observed in Fig. 5a, drawing 100 samples per grid vertex not only causes more computation time but also results in a lower accuracy compared to our closed-form results in Fig. 5c. As we increase the number of MC samples to 2000 in Fig. 5b, the accuracy increases (i.e., results converge), but the computational performance reduces. Specifically, compared to the MC sampling method with 2000 samples, our closed-form solution provides a $64\times$ speed-up for local minimum probability and $119\times$ speed-up for the saddle probability computation with a comparable accuracy. The timings reported are for the Python serial implementation on a quad-core Intel I7 processor. Note that the MC sampling approach for a saddle takes on average more time compared to MC sampling for the local minimum because of the computation of two terms, t_1 and t_2, for a saddle (see Sec. 4.2). The RMSE and maximum probability difference ($Error\_max$) between the MC and our closed-form solutions shown in Fig. 5d and a convergence curve in Fig. 5e confirm the correctness of our derivations and algorithms. Similar convergence curves and results for the Epanechnikov and histogram (closed-form and semianalytical) models are reported in the supplement.

5.2 Comparison of Parametric Vs. Nonparametric Models

We present a comparison of parametric and nonparametric models in terms of accuracy and performance for computation of critical point probability. Here, we show the results for the synthetic data modeled as a Gaussian mixture model, which is a common data model used in scientific and topological analysis [10,60,70]. In particular, we generate the ensemble of 50 members, in which each member comprises a mixture of two Gaussians. As shown in Fig. 6, 40 members correspond to a Gaussian mixture, in which peaks (yellow) are oriented in NW-SE direction. These 40 members are considered as the ground truth, and the
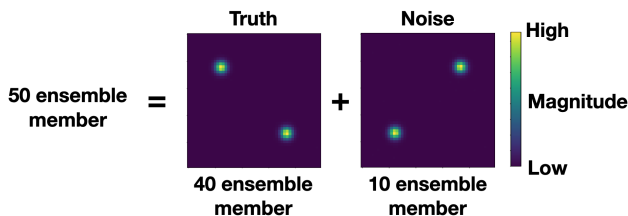

Fig. 6: Gaussian mixture data for computation of critical point probability.

variation across them corresponds to small noise randomly added. The remaining 10 members are a $90°$ rotated version of the first 40 members oriented in NE-SW direction. These rotated members represent the noisy data or outliers. In other words, the peaks represented by these rotated members are not the true peaks.

We apply parametric (independent uniform, Epanechnikov, Gaussian, and multivariate Gaussian) and nonparametric (histogram with 5 bins) noise models to the Gaussian mixture ensemble (Fig. 6) for visualization of the local maximum probability (i.e., $Pr[p = l_{max}]$). The results in Fig. 7 demonstrate the quality and performance of various noise models. Figure 7a visualizes the result for the independent uniform noise model using our closed-form computational algorithm (Sec. 4.2). At each pixel, the data range is computed based on the minimum and maximum data values observed across the ensemble. The closed-form uniform noise model is the fastest, but shows all critical points in NW-SE-NE-SW directions equally likely with moderate probability. This is not a desirable result, as the critical points in the NE-SW directions correspond to noise or outliers (see Fig. 6) and should not result in important features in a probabilistic visualization.

The quality of results is improved overall in the case of independent Gaussian and Epanechnikov noise models, as observed in Fig. 7b and Fig. 7c, respectively. For both models, we determine the distribution range based on the sample mean and standard deviation per pixel across the ensemble members. Since the Gaussian model does not have a closed-form solution, we performed MC sampling with 1000 samples to compute the local maximum probability, which is computationally expensive compared to the uniform noise model. On the other hand, the Gaussian noise model highlights the true critical points (NW-SE) better with respect to local neighbors compared to the outlier critical points (NE-SW). However, both true and outlier critical points get moderate probability values assigned. Our proposed closed-form result for the Epanechnikov model in Fig. 7c is similar to the result for the Gaussian model in Fig. 7b, obtained in reduced time with about $2.48\times$ speed-up. The Epanechnikov model enhances the results compared to the uniform model because of its similar characteristics to the Gaussian model, i.e., greater mean weight, a bell-like shape, and more smoothness, compared to the uniform model (see Sec. 4.3). The multivariate Gaussian model in Fig. 7d from the previous work [38,49] prominently highlights both the true and outlier peaks with high probabilities (bright yellow). Capturing data correlation assigns high probability to the outlier critical points and counter intuitively exhibits less robustness to outliers.

Figure 7e and Fig. 7f visualize the results for the independent non-
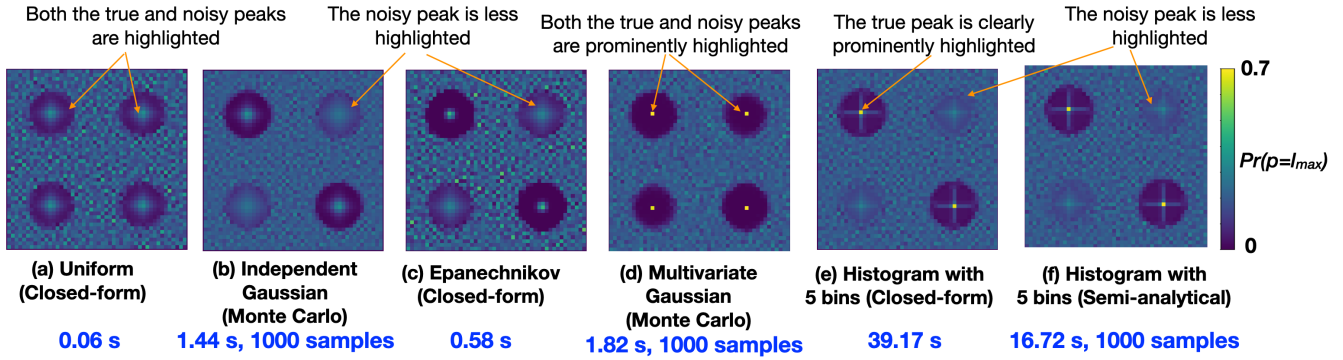
Fig. 7: A comparison of parametric vs. nonparametric noise models in terms of quality and performance of local maximum probability (i.e., $Pr[p = l_{max}]$) computation for the ensemble dataset shown in Fig. 6. Our closed-form uniform noise model (a) is the fastest, but it exhibits less accurate results in that it equally highlights both the true and noisy peaks with moderate probability. The independent Gaussian noise (b) model without a closed-form solution and the independent Epanechnikov model with a closed-form solution (c) highlight the true peaks better with respect to the local neighbors, but with moderate probability and slightly more compute time. The multivariate Gaussian model (d) from previous work [38, 49] highlights both the true and noisy peaks with high probabilities (bright yellow). The proposed closed-form histogram method (e) exhibits the most robust result to outliers clearly highlighting the true local maximum positions with high probability (bright yellow), but it is the slowest in performance. The proposed semianalytical histogram method (f) reduces the time while maintaining sufficient accuracy when compared to (e).

parametric histogram model (Sec. 4.4) with 5 bins, which exhibit the greatest robustness to outliers. Specifically, they clearly highlight the true local maximum positions with a high probability (bright yellow) and smooth out noisy peaks with a moderate probability (indicated by the arrows in Fig. 7e-f). The closed-form nonparametric models, however, require more computations (see Sec. 4.4.1). The computational time of the closed-form nonparametric solution is mitigated with the semianalytical solution (see Sec. 4.4.2) while maintaining comparable accuracy, as seen from Fig. 7e and Fig. 7f. All timing results reported in Fig. 7 are again obtained with a serial Python implementation on a quad-core Intel I7 processor. To be able to accommodate the high-quality nonparametric solutions in the visualization systems, we accelerate our algorithms using a C++ parallel implementation with VTK-m [42] (Sec. 4.5), as presented for the real dataset results in the next section.

5.3 Real Datasets

Climate Data: Figure 8 visualizes the mean sea-level pressure variable simulated from an earth and climate simulation model – Energy Exascale Earth System Model (E3SM) [22] – with a grid resolution of $0.25°$. The result is visualized for a tropical region sampled on a regular grid of size 240×960, in which the local minima in relatively low pressure regions (i.e., blue) indicate the potential of existence of tropical cyclones. The dataset was compressed using an error-controlled lossy compressor – MGARD [23] – under a relative L2 error bound (*eb*) of 1×10^{-3}, which provided a compression ratio of 16.68. We study the uncertainty of critical points in data decompressed using MGARD.

Figure 8a visualizes the original mean sea-level pressure data colormapped with magnitude. The critical points (i.e., local minima l_{min}) are shown as the blue spheres extracted using TTK [59]. Figure 8b visualizes the decompressed data and its critical points. As observed, the number of critical points increases in certain regions due to the compression errors, as illustrated by the inset view. There is, however, no indication of how much uncertainty or probability of critical points is in a decompressed field, which can lead to less reliable TDA. Figure 8c visualizes critical points in a decompressed field along with uncertainty. We utilize the value of error bound *eb* to derive critical point probabilities. Specifically, at each grid position, the uncertain data range is $[d' - \frac{eb}{2}, d' + \frac{eb}{2}]$, where d' denotes the decompressed value. Since we do not have prior knowledge of the distribution over the uncertain range, we model data uncertainty with the uniform distribution and apply our proposed algorithms (Sec. 4.2).

In Fig. 8c, we visualize the derived local minimum probability using a heightfield, an idea similar to the previous work by Petz et al. [49]. In a heightmap, each grid point of interest is elevated and colormapped proportional to the critical point probability. In Fig. 8c, the heightfield tailored to critical points clearly indicates points with relatively high probability. For the inset view in Fig. 8c, the high probability critical

points (enclosed by the green dotted box) also appear in the original data in Fig. 8a. In contrast, newly created critical points due to compression errors have a low probability. Figure 8d visualizes the heightfield for every grid pixel, in which the magenta regions (with $Pr[p = l_{min}] > 0.2$) reflect a pattern of critical point positions that is similar to the one in the original data (Fig. 8a). We additionally present a quantitative evaluation for the dataset in the supplementary material.

We measure the performance and accuracy of the climate dataset results through our VTK-m implementation. Since the VTK-m implementation is platform portable, we run it on a serial processor and AMD GPUs on the Oak Ridge National Laboratory's Frontier supercomputer [3] and NVIDIA GPUs on National Energy Research Scientific Computing Center's Perlmutter supercomputer [37]. Using a conventional MC solution with 1000 samples per grid point takes 4.94 seconds on a serial backend. The proposed algorithms for closed-form computation (Sec. 4.2) compute the true solution in 0.012 seconds on a serial backend, thereby providing a 411× speed-up. Running the VTK-m code on AMD GPU yields a closed-form result in 0.003 seconds, which corresponds to a 1646× speed-up compared to the MC solution and a 4× speed-up compared to the closed-form solution on a serial backend. The NVIDIA GPU yields a closed-form result in 0.004 seconds, a performance close to the AMD GPU. We present additional accuracy and performance results of our VTK-m code for the AMD GPU in the supplementary material.

Oceanology Data: In our next experiment, we evaluate the quality and performance of parametric and nonparametric noise models for the Red Sea ensemble simulations [56]. The results are visualized in Fig. 1. The dataset is downloaded from the 2020 IEEE SciVis contest website. The ensemble comprises 20 members each with grid resolution 500×500. Understanding eddy positions is crucial for oceanologists to gain insight into energy and particle transport patterns in oceans. Therefore, we investigate the local minima positions that potentially correspond to the eddy features. Since the original simulation data are too noisy, we applied topological simplification [19] using TTK [59] to each ensemble member as a denoising (preprocessing) step until each ensemble member has approximately the same number of critical points. This strategy corresponds to the *persistence graph* idea from the prior work by Athawale et al. [7], which plots the number of local minima against the persistence simplification level to decide the amount of simplification. Having simplified the topology, we efficiently compute and visualize critical point uncertainty in ParaView with our VTK-m code as a backend, as documented in the supplement.

Fig. 1a-b visualize the results for the parametric noise models. We apply spherical glyphs in ParaView to help quickly identify the positions with high probability. The points with a larger local minimum probability ($Pr[p = l_{min}]$), therefore, have a bigger radius and a red/orange color in the sphere glyphs. The result of the multivariate Gaussian
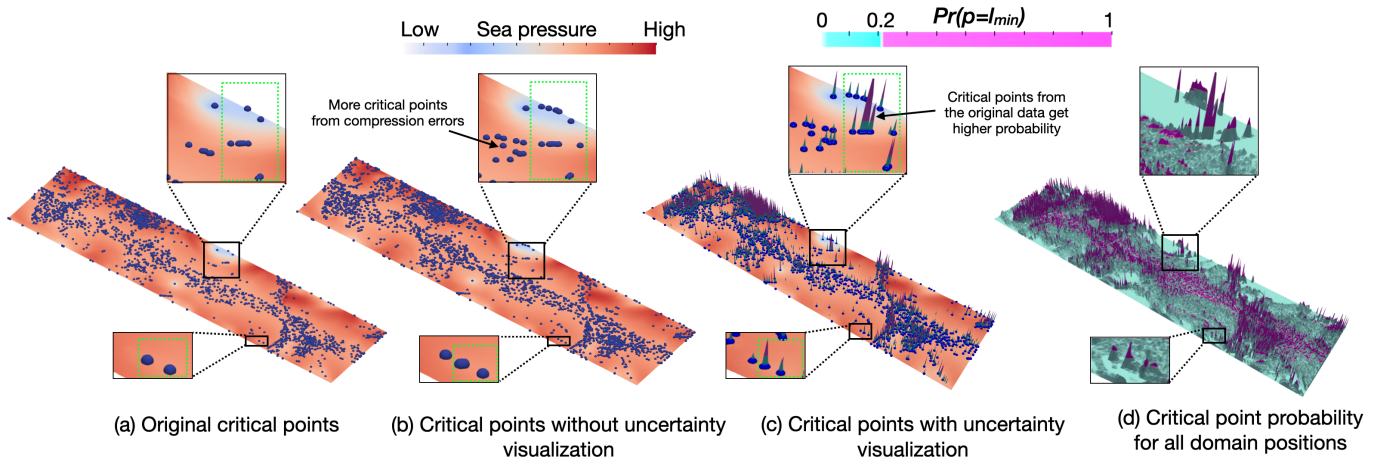
Fig. 8: Critical point visualization for the climate dataset. (a) The colormapped original data with local minima shown as the blue spheres. (b) Compression errors result in increased numbers of critical points for which no uncertainty is visualized. (c) Computation of critical point probability with the uniform noise model and visualization of uncertainty through elevation proportional to probability. The critical points in the original data have higher probabilities (tall mountain with magenta color enclosed by the green box). (d) The heightmap of probabilities for every domain position. High probabilities (i.e., tall mountains with magenta peaks) are observed in similar regions as the critical points in the original data.

model from previous work by Petz et al. [49] and Liebmann et al. [38] with 2000 MC samples is visualized in Fig. 1c. Capturing the correlation using the multivariate Gaussian model significantly reduces the probability of local minima in certain regions (blue regions) and emphasizes fewer critical points. Fig. 1d visualizes the results for our proposed nonparametric histogram model with four bins (Sec. 4.4.1). The uniform, Epanechnikov, multivariate Gaussian, and histogram models took 0.094, 0.102, 0.167 and 0.145 seconds, respectively, on the Frontier supercomputer's AMD GPU. Fig. 1e-f visualize the critical points (yellow) extracted from two random members of the ensemble.

Since we do not know the true critical points for the ensemble data, we make a few interesting observations by comparing results of different noise models (shown with the boxes in Fig. 1). The white boxes indicate positions where the two local minimum positions are consistently observed with high probability across all models, and therefore, can be trusted. The green boxes show critical points that are captured as high probability by the proposed nonparametric histogram models, but not by other models. This result is interesting because the histogram showed the highest resilience to outliers, and therefore, more trustworthy results, in our synthetic experiments in Fig. 7. As observed in individual ensemble members, critical points are also seen in the areas marked by the green boxes. Similarly, the cyan boxes mark critical points that are captured by the multivariate Gaussian model, but not by any other model. This result necessitates further investigation because the multivariate Gaussian model was less robust to outliers in our synthetic experiment in Fig. 7. Lastly, the pink boxes show the positions where the multivariate Gaussian and nonparametric models agree, whereas the others do not agree.

6 CONCLUSION AND FUTURE WORK

In this paper, we study the propagation of uncertainty in critical points, the fundamental topological descriptors of scalar field data. Our main contribution of this paper is the creation of a novel efficient algorithms that compute critical point probability in closed form for parametric and nonparametric uncertainty with finite support. We demonstrate the effectiveness of our algorithms through enhanced accuracy and performance over classical MC sampling. We integrate our algorithms with the VTK-m library [42] to further accelerate performance using serial, AMD, and NVIDIA backends. We show seamless integration of our VTK-m algorithms with ParaView [2] (see the supplement), which is a key to making our algorithms accessible to a wide audience. Our synthetic experiments show the greater resilience of our proposed nonparametric models to outliers compared to parametric models, similar to the prior studies [8, 53]. We present the practical utility of our techniques through application to climate and oceanology datasets.

A few limitations of this work are important and need to be ad-

dressed in the future. Currently, we assume that the data at each grid point have uncertainty only over finite bounds. Although the finite bounds assumption is generally true in practice, finding exact bounds that are needed for our proposed algorithms can be nontrivial. In the demonstrated results, we derived the upper and lower bounds for the climate and Red Sea ensemble datasets at each grid point based on the data captured by simulations. Thus, the quality of our results strongly depends on how well the application can provide upper and lower bounds for the values at a grid point. That said, one of the benefits of the proposed nonparametric models is that they can mitigate the effects of overestimation (or coarseness) of bounds resulting from the outliers by assigning a lower weight or probability to outliers (see Sec. 4.4 and the synthetic experiments in Sec. 5.2). In the frequent case of ensemble simulations, however, a priori knowledge or experiments demonstrating the robustness of lower and upper bounds provided by the ensembles can help to further improve the trustworthiness of the results.

Another limitation of our work is that we assume neighboring data points to be uncorrelated. This independent noise assumption can be true in the case of measurement data but is not often true for the real datasets or ensemble simulations encountered in practice [4, 9, 13, 51]. Spatially close grid points are often strongly correlated in real datasets because to reliably represent a smooth function, any reasonable grid must have a spatial resolution finer than frequencies of relevance. Our independent noise assumption, therefore, can lead to overestimation of probabilities from ignoring the local correlation (as shown too in the previous studies [9, 51]). Considering spatial correlation, however, has a caveat of higher sensitivity of results to outliers, as demonstrated for the multivariate Gaussian model in our synthetic experiments in Sec. 5.2. Even though the core integration formulae in Eq. (3) and Eq. (4) are valid for the correlated data, their computation becomes more challenging because of the complexity of accommodating linear or nonlinear correlations. Thus, further research is needed to derive closed-form solutions for critical point probability computation that can accommodate spatial correlation and are robust to outliers.

Our methods are currently limited to critical points of uncertain 2D fields based on four neighbors per grid point. In the future, we would like to extend our work to more neighbors (e.g., six- or eight-pixel neighborhood based on the triangulation) and 3D datasets, which can be complex because of higher order of integration templates. We utilize the sphere glyphs and heightmaps [49] for the visualization of critical point probabilities. However, both methods can lead to occlusion and clutter. Thus, further study is needed to evaluate the perceptual quality of sphere glyph and heightmap methods and, possibly, derive new rendering techniques for enhanced perception of uncertainties. Finally, we will also investigate uncertainty in other topological visualizations based on critical points, including contour trees and persistence diagrams.

REFERENCES

[1] D. H. Ackley. *A connectionist machine for genetic hillclimbing*. Kluwer Academic Publishers Norwell, MA, USA, 1987. doi: 10.1007/978-1-4613-1997-9 6

[2] J. Ahrens, B. Geveci, and C. Law. *ParaView: An End-User Tool for Large Data Visualization*, chap. 36, pp. 717–731. Elsevier, 2005. doi: 10.1016/B978-012387582-2/50038-1 1, 2, 9

[3] S. Atchley et al. Frontier: Exploring exascale. In *SC: High Performance Computing, Networking, Storage and Analysis*, November 2023. doi: 10.1145/3581784.3607089 8

[4] T. M. Athawale and A. Entezari. Uncertainty quantification in linear interpolation for isosurface extraction. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2723–2732, 2013. doi: 10.1109/TVCG.2013.208 2, 9

[5] 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*, 29(1):613–623, 2023. doi: 10.1109/TVCG.2022.3209424 2

[6] T. M. Athawale, B. Ma, E. Sakhaee, C. R. Johnson, and A. Entezari. Direct volume rendering with nonparametric models of uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1797–1807, Feb. 2021. doi: 10.1109/TVCG.2020.3030394 2, 6

[7] T. M. Athawale, D. Maljovec, L. Yan, C. R. Johnson, V. Pascucci, and B. Wang. Uncertainty visualization of 2D Morse complex ensembles using statistical summary maps. *IEEE Transactions on Visualization and Computer Graphics*, 28(4):1955–1966, Apr. 2022. doi: 10.1109/TVCG.2020.3022359 2, 8

[8] T. M. Athawale, E. Sakhaee, and A. Entezari. Isosurface visualization of data with nonparametric models for uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):777–786, 2016. doi: 10.1109/TVCG.2015.2467958 1, 2, 6, 9

[9] T. M. Athawale, S. Sane, and C. R. Johnson. Uncertainty visualization of the marching squares and marching cubes topology cases. In *2021 IEEE Visualization Conference (VIS)*, pp. 106–110, 2021. doi: 10.1109/VIS49827.2021.9623267 9

[10] T. M. Athawale, B. Triana, T. Kotha, D. Pugmire, and P. Rosen. A comparative study of the perceptual sensitivity of topological visualizations to feature variations. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):1074–1084, 2024. doi: 10.1109/TVCG.2023.3326592 7

[11] T. M. Athawale, Z. Wang, C. R. Johnson, and D. Pugmire. Data-Driven Computation of Probabilistic Marching Cubes for Efficient Visualization of Level-Set Uncertainty. In C. Tominski, M. Waldner, and B. Wang, eds., *EuroVis 2024 - Short Papers*. The Eurographics Association, 2024. doi: 10.2312/evs.20241071 2

[12] P.-T. Bremer, G. Weber, V. Pascucci, M. Day, and J. Bell. Analyzing and tracking burning structures in lean premixed hydrogen flames. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):248–260, 2010. doi: 10.1109/TVCG.2009.69 1

[13] K. Brodlie, R. A. Osorio, and A. Lopes. A review of uncertainty in data visualization. In J. Dill, R. Earnshaw, D. Kasik, J. Vince, and P. C. Wong, eds., *Expanding the Frontiers of Visual Analytics and Visualization*, pp. 81–109. Springer Verlag London, 2012. doi: 10.1007/978-1-4471-2804-5_6 1, 2, 3, 9

[14] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *ACM-SIAM Symposium on Discrete Algorithms (SODA 2000)*, pp. 918–926, January 2000. doi: 10.1016/S0925-7721(02)00093-7 1

[15] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, K. Bonnell, M. Miller, G. Weber, C. Harrison, D. Pugmire, T. Fogal, C. Garth, A. Sanderson, E. W. Bethel, M. Durant, D. Camp, J. Favre, O. Rübel, P. Navratil, and F. Vivodtzev. VisIt: An end-user tool for visualizing and analyzing very large data. *SciDAC*, pp. 1–16, 01 2011. 2

[16] S. Djurcilov, K. Kim, P. Lermusiaux, and A. Pang. Visualizing scalar volumetric data with uncertainty. *Computers & Graphics*, 26(2):239–248, 2002. doi: 10.1016/S0097-8493(02)00055-9 2

[17] H. Edelsbrunner and J. Harer. *Computational Topology - an Introduction*. American Mathematical Society, 2010. 1

[18] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Morse-Smale complexes for piecewise linear 3-manifolds. In *Symposium on Computational Geometry (SoCG)*, SCG '03, p. 361–370. Association for Computing Machinery, New York, NY, USA, 2003. doi: 10.1145/777792.777846 1

[19] H. Edelsbrunner, D. Letscher, and A. J. Zomorodian. Topological persistence and simplification. *Discrete and Computational Geometry*, 28:511–533, 2002. doi: 10.1109/SFCS.2000.892133 8

[20] 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, Jan. 2016. doi: 10.1109/TVCG.2015.2467204 2

[21] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann. Visual analysis of spatial variability and global correlations in ensembles of iso-contours. *Computer Graphics Forum*, 35(3):221–230, 2016. doi: 10.1111/cgf.12898 2

[22] J.-C. Golaz, L. P. Van Roekel, X. Zheng, A. F. Roberts, J. D. Wolfe, W. Lin, A. M. Bradley, Q. Tang, M. E. Maltrud, R. M. Forsyth, et al. The DOE E3SM model version 2: Overview of the physical model and initial model evaluation. *Journal of Advances in Modeling Earth Systems*, 14(12):e2022MS003156, 2022. doi: 10.1029/2022MS003156 8

[23] Q. Gong, J. Chen, B. Whitney, X. Liang, V. Reshniak, T. Banerjee, J. Lee, A. Rangarajan, L. Wan, N. Vidal, et al. MGARD: A multigrid framework for high-performance, error-controlled data compression and refactoring. *SoftwareX*, 24:101590, 2023. doi: 10.1016/j.softx.2023.101590 8

[24] G. Grigoryan and P. Rheingans. Probabilistic surfaces: point based primitives to show surface uncertainty. In *IEEE Visualization Conference (VIS)*, pp. 147–153, 2002. doi: 10.1109/VISUAL.2002.1183769 2

[25] D. Günther, J. Salmon, and J. Tierny. Mandatory critical points of 2D uncertain scalar fields. *Computer Graphics Forum*, 33(3):31–40, 2014. doi: 10.1111/cgf.12359 2

[26] H. Guo, W. He, T. Peterka, H.-W. Shen, S. M. Collis, and J. J. Helmus. Finite-time Lyapunov exponents and Lagrangian coherent structures in uncertain unsteady flows. *IEEE Transactions on Visualization and Computer Graphics*, 22(6):1672–1682, 2016. doi: 10.1109/TVCG.2016.2534560 2

[27] A. Gyulassy and V. Natarajan. Topology-based simplification for feature extraction from 3D scalar fields. In *IEEE Visualization Conference (VIS)*, pp. 535–542, 2005. doi: 10.1109/VISUAL.2005.1532839 1

[28] M. Han, T. M. Athawale, D. Pugmire, and C. R. Johnson. Accelerated probabilistic marching cubes by deep learning for time-varying scalar ensembles. In *IEEE Visualization Conference (VIS)*, pp. 155–159, 2022. doi: 10.1109/VIS54862.2022.00040 2

[29] S. Hazarika, A. Biswas, and H.-W. Shen. Uncertainty visualization using copula-based analysis in mixed distribution models. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):934–943, 2018. doi: 10.1109/TVCG.2017.2744099 2

[30] W. He, C.-M. Chen, X. Liu, and H.-W. Shen. A Bayesian approach for probabilistic streamline computation in uncertain flows. In *IEEE Pacific Visualization Symposium (PacificVis)*, pp. 214–218, 2016. doi: 10.1109/PACIFICVIS.2016.7465273 2

[31] F. Jiao, J. M. Phillips, Y. Gur, and C. R. Johnson. Uncertainty visualization in HARDI based on ensembles of ODFs. In *IEEE Pacific Visualization Symposium*, pp. 193–200, 2012. doi: 10.1109/PacificVis.2012.6183591 2

[32] C. R. Johnson. Top scientific visualization research problems. *IEEE Computer Graphics and Applications*, 24(4):13–17, 2004. doi: 10.1109/MCG.2004.20 2

[33] C. R. Johnson and A. R. Sanderson. A next step: Visualizing errors and uncertainty. *IEEE Computer Graphics and Applications*, 23(5):6–10, Sept.-Oct. 2003. doi: 10.1109/MCG.2003.1231171 2

[34] D. K. Jones. Determining and visualizing uncertainty in estimates of fiber orientation from diffusion tensor MRI. *Magnetic Resonance in Medicine*, 49(1):7–12, Dec. 2002. doi: 10.1002/mrm.10331 2

[35] 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*, 24(5):861–890, May 2021. doi: 10.1007/s12650-021-00755-1 2

[36] H. Li, I. J. Michaud, A. Biswas, and H. Shen. Efficient level-crossing probability calculation for Gaussian process modeled data. In *2024 IEEE 17th Pacific Visualization Conference (PacificVis)*, pp. 252–261. IEEE Computer Society, Los Alamitos, CA, USA, apr 2024. doi: 10.1109/PacificVis60374.2024.00035 2

[37] J. Li, G. Michelogiannakis, B. Cook, D. Cooray, and Y. Chen. Analyzing resource utilization in an HPC system: A case study of NERSC's Perlmutter. In *High Performance Computing*, May 2023. doi: 10.1007/978-3-031-32041-5_16 8

[38] T. Liebmann and G. Scheuermann. Critical points of Gaussian-distributed scalar fields on simplicial grids. *Computer Graphics Forum*, 35(3):361–370, 2016. doi: 10.1111/cgf.12912 2, 7, 8, 9

[39] S. Liu, J. A. Levine, P.-T. Bremer, and V. Pascucci. Gaussian mixture model based volume visualization. In *IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 73–77, Oct. 2012. doi: 10.1109/LDAV.2012.6378978 2

[40] C. Lundström, P. Ljung, A. Persson, and A. Ynnerman. Uncertainty visualization in medical volume rendering using probabilistic animation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1648–1655, 2007. doi: 10.1109/TVCG.2007.70518 2

[41] M. Mihai and R. Westermann. Visualizing the stability of critical points in uncertain scalar fields. *Computers & Graphics*, 41:13–25, 2014. doi: 10.1016/j.cag.2014.01.007 2

[42] K. Moreland, C. Sewell, W. Usher, L.-T. Lo, J. Meredith, D. Pugmire, J. Kress, H. Schroots, K.-L. Ma, H. Childs, M. Larsen, C.-M. Chen, R. Maynard, and B. Geveci. VTK-m: Accelerating the visualization toolkit for massively threaded architectures. *IEEE Computer Graphics and Applications*, 36(3):48–58, 2016. doi: 10.1109/MCG.2016.48 1, 2, 6, 8, 9

[43] V. Natarajan, P. Koehl, Y. Wang, and B. Hamann. Visual analysis of biomolecular surfaces. In L. Linsen, H. Hagen, and B. Hamann, eds., *Visualization in Medicine and Life Sciences*, pp. 237–255. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. doi: 10.1007/978-3-540-72630-2_14 1

[44] F. Nauleau, F. Vivodtzev, T. Bridel-Bertomeu, H. Beaugendre, and J. Tierny. Topological analysis of ensembles of hydrodynamic turbulent flows an experimental study. In *2022 IEEE 12th Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 1–11, 2022. doi: 10.1109/LDAV57265.2022.9966403 1

[45] 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

[46] M. Otto, T. Germer, and H. Theisel. Uncertain topology of 3D vector fields. In *IEEE Pacific Visualization Symposium (PacificVis)*, pp. 67–74, Mar. 2011. doi: 10.1109/PACIFICVIS.2011.5742374 2

[47] A. T. Pang, C. M. Wittenbrink, and S. K. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 13:370–390, 1997. doi: 10.1007/s003710050111 2

[48] E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, Sept. 1962. doi: 10.1214/aoms/1177704472 2, 6

[49] C. Petz, K. Pöthkow, and H.-C. Hege. Probabilistic local features in uncertain vector fields with spatial correlation. *Computer Graphics Forum*, 31(3pt2):1045–1054, 2012. doi: 10.1111/j.1467-8659.2012.03097.x 2, 7, 8, 9

[50] K. Pöthkow and H.-C. Hege. Positional uncertainty of isocontours: Condition analysis and probabilistic measures. *IEEE Transactions on Visualization and Computer Graphics*, 17(10):1393–1406, 2011. doi: 10.1109/TVCG.2010.247 2

[51] K. Pöthkow, B. Weber, and H.-C. Hege. Probabilistic marching cubes. *Computer Graphics Forum*, 30(3):931–940, June 2011. doi: 10.1111/j.1467-8659.2011.01942.x 2, 9

[52] K. Potter, P. Rosen, and C. R. Johnson. From quantification to visualization: A taxonomy of uncertainty visualization approaches. In A. M. Dienstfrey and R. F. Boisvert, eds., *Uncertainty Quantification in Scientific Computing*, pp. 226–249. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. doi: 10.1007/978-3-642-32677-6_15 2

[53] K. Pöthkow and H.-C. Hege. Nonparametric models for uncertainty visualization. *Computer Graphics Forum*, 32(3pt2):131–140, July 2013. doi: 10.1111/cgf.12100 1, 2, 6, 9

[54] P. J. Rhodes, R. S. Laramee, R. D. Bergeron, and T. M. Sparr. Uncertainty Visualization Methods in Isosurface Rendering. In *Eurographics (Short Presentations)*. Eurographics Association, 2003. doi: 10.2312/egs.20031054 2

[55] S. Sane, T. M. Athawale, and C. R. Johnson. Visualization of uncertain multivariate data via feature confidence level-sets. In M. Agus, C. Garth, and A. Kerren, eds., *EuroVis 2021 (Short Papers)*. The Eurographics Association, 2021. doi: 10.2312/evs.20211053 2

[56] S. Sanikommu, H. Toye, P. Zhan, G. Langodan, O. Knio, and I. Hoteit. Impact of atmospheric and model physics perturbations on a high-resolution ensemble data assimilation system of the Red Sea. *Journal of Geophysical Research: Oceans*, 125(8):e2019JC015611, July 2020. doi: 10.1029/2019JC015611 1, 8

[57] S. Schlegel, N. Korn, and G. Scheuermann. On the interpolation of data with normally distributed uncertainty for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2305–2314, 2012. doi: 10.1109/TVCG.2012.249 2

[58] F. Siddiqui, T. Höllt, and A. Vilanova. A progressive approach for uncertainty visualization in diffusion tensor imaging. *Computer Graphics Forum*, 40(3):411–422, 2021. doi: 10.1111/cgf.14317 2

[59] J. Tierny, G. Favelier, J. A. Levine, C. Gueunet, and M. Michaux. The Topology Toolkit. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):832 – 842, Jan 2018. doi: 10.1109/TVCG.2017.2743938 2, 8

[60] J. Vidal, J. Budin, and J. Tierny. Progressive Wasserstein barycenters of persistence diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):151–161, 2020. doi: 10.1109/TVCG.2019.2934256 2, 7

[61] D. Vietinghoff, M. Böttinger, G. Scheuermann, and C. Heine. Detecting critical points in 2D scalar field ensembles using Bayesian inference. In *IEEE Pacific Visualization Symposium (PacificVis)*, pp. 1–10, 2022. doi: 10.1109/PacificVis53943.2022.00009 2

[62] D. Vietinghoff, M. Böttinger, G. Scheuermann, and C. Heine. Visualizing confidence intervals for critical point probabilities in 2D scalar field ensembles. In *IEEE Visualization Conference (VIS)*, pp. 145–149. IEEE Computer Society, Los Alamitos, CA, USA, oct 2022. doi: 10.1109/VIS54862.2022.00038 2

[63] D. Vietinghoff, M. Böttinger, G. Scheuermann, and C. Heine. A mathematical foundation for the spatial uncertainty of critical points in probabilistic scalar fields. In *Topological Data Analysis and Visualization (TopoInVis)*, pp. 30–40. IEEE Computer Society, Los Alamitos, CA, USA, oct 2023. doi: 10.1109/TopoInVis60193.2023.00010 2

[64] J. Wang, S. Hazarika, C. Li, and H.-W. Shen. Visualization and visual analysis of ensemble data: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 25(9):2853–2872, 2019. doi: 10.1109/TVCG.2018.2853721 2

[65] Z. Wang, T. M. Athawale, K. Moreland, J. Chen, C. R. Johnson, and D. Pugmire. *FunMC*^{2}: A Filter for Uncertainty Visualization of Marching Cubes on Multi-Core Devices. In R. Bujack, D. Pugmire, and G. Reina, eds., *Eurographics Symposium on Parallel Graphics and Visualization*. The Eurographics Association, 2023. doi: 10.2312/pgv.20231081 2, 6

[66] C. Wittenbrink, A. Pang, and S. Lodha. Glyphs for visualizing uncertainty in vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):266–279, 1996. doi: 10.1109/2945.537309 2

[67] Wolfram|Alpha. Wolfram Alpha LLC. http://www.wolframalpha.com. 6

[68] K. Wu and S. Zhang. A contour tree based visualization for exploring data with uncertainty. *International Journal for Uncertainty Quantification*, 3:203–223, 2012. doi: 10.1615/Int.J.UncertaintyQuantification.2012003956 2

[69] L. Yan, T. B. Masood, R. Sridharamurthy, F. Rasheed, V. Natarajan, I. Hotz, and B. Wang. Scalar field comparison with topological descriptors: Properties and applications for scientific visualization. *Computer Graphics Forum*, 40(3):599–633, 2021. doi: 10.1111/cgf.14331 2

[70] L. Yan, Y. Wang, E. Munch, E. Gasparovic, and B. Wang. A structural average of labeled merge trees for uncertainty visualization. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):832–842, 2020. doi: 10.1109/TVCG.2019.2934242 2, 7