Statistical and Machine Learning Approaches For Visualizing and Analyzing Large-Scale Simulation Data

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Subhashis Hazarika, B.Tech., M.S.

Graduate Program in Department of Computer Science and Engineering

The Ohio State University

2019

Dissertation Committee:

Dr. Han-Wei Shen, Advisor

Dr. Rephael Wenger

Dr. Yusu Wang

Abstract

Recent advancements in the field of computational sciences and high-performance computing have enabled scientists to design high-resolution computational models to simulate various real-world physical phenomenon. In order to gain key scientific insights about the underlying phenomena it is important to analyze and visualize the output data produced by such simulations. However, large-scale scientific simulations often produce output data whose size can range from a few hundred gigabytes to the scale of terabytes or even petabytes. Analyzing and visualizing such large-scale simulation data is not trivial. Moreover, scientific datasets are often multifaceted (multivariate, multi-run, multi-resolution, etc.), which can introduce additional complexities to the analyses and visualization activities.

This dissertation addresses three broad categories of data analysis and visualization challenges: *(i) multivariate distribution-based data summarization*, *(ii) uncertain analysis in ensemble simulation data*, and *(iii) simulation parameter analysis and exploration*. We proposed statistical and machine learning-based approaches to overcome these challenges.

A common strategy to deal with large-scale simulation data is to partition the simulation domain and create data summaries in the form of statistical probability distributions. Instead of storing high-resolution raw data, storing the compact statistical data summaries results in reduced storage overhead and alleviated I/O bottleneck issues. However, for multivariate simulation data using standard multivariate distributions for creating data summaries is not feasible. Therefore, we proposed a flexible copula-based multivariate distribution modeling

strategy to create multivariate data summaries during simulation execution time (i.e, *in situ* data modeling). The resulting data summaries can be subsequently used to perform scalable post-hoc analysis and visualization.

In many cases, scientists execute their simulations multiple times with different initial conditions and/or input parameters in order to model the underlying uncertainty of the physical phenomena. Analyzing this collection of simulation outputs, generally referred to as *ensemble simulation data*, can be overwhelming for the scientists. To this end, we proposed a copula-based approach to model uncertainty in ensemble simulations using mixed statistical distribution models and preserving the spatial correlation among local neighbors. We utilize this statistical model to extract and visualize the uncertainty of features like isocontours and vortices in ensemble simulation data. Moreover, to guide the users in identifying interesting features for further uncertainty analysis, we proposed a two-stage information-theoretic framework for the exploration of scalar value ranges as well as the corresponding ensemble isocontours of selected values.

Finally, for many newly designed simulation models, it is important to properly calibrate the simulation input parameters before applying them in real scientific studies. For computationally expensive simulations, performing such exploratory parameter analyses can become computationally prohibitive operations. Therefore, we proposed a neural network assisted visual analysis framework to enable interactive simulation parameter analysis. A trained neural network acts as a surrogate model, replacing the expensive simulation, to facilitate interactive exploratory analysis. We collaborated with computational biologists to assist them in analyzing an expensive yeast cell polarization simulation model.

Dedicated to *deuta*, *maa*, and *pinky ba*.

Acknowledgments

My doctoral journey at The Ohio State University has been truly a life-changing experience. It would have not been possible without the support and guidance from so many people in my life. I would like to take this opportunity to express my heartfelt appreciation to these individuals at the beginning of my dissertation.

Firstly, I would like to express my sincere gratitude to my advisor Prof. Han-Wei Shen for his continuous support throughout my graduate studies and for helping me shape my research career. His knowledge, motivation, and patience have been a constant source of inspiration for me. In particular, I would like to sincerely thank my dissertation committee members, Prof. Rephael Wenger and Prof. Yusu Wang for providing valuable insights and suggestions to help improve my dissertation.

I was fortunate to get the opportunity to work as a graduate research intern for two summers at Los Alamos National Laboratory (LANL). I would like to sincerely thank Christine Sweeney and Dr. Ayan Biswas for mentoring me during my internships and helping me advance my doctoral research. I would also like to thank my collaborators Prof. Jen-Ping Chen, from the Department of Mechanical and Aerospace Engineering and Prof. Ching-Shan Chou, from the Department of Mathematics for their invaluable domain feedbacks, which helped in increasing the real-world impact of my research. I would also like to take this opportunity to thank all my previous mentors and teachers from my undergraduate studies and high-schools in India. I would like to thank my undergraduate

mentor Prof. Sanjib Sadhu for inspiring me to pursue research in the field of computer science. Special gratitude goes to my high-school math tutor Mariakutty Jacob, who played a significant role in transforming me as a student and developing a profound love for mathematics.

Conducting my research as a member of the GRAVITY research group, led by Prof. Shen, was a privilege. The interactions and intense research discussions that I had with my fellow lab-mates have helped me immensely in developing my research ideas. I would like to thank all my seniors and current lab-mates in the group: Soumya Dutta, Ayan Biswas, Ko-Chih Wang, Junpeng Wang, Cheng Li, Wenbin He, Chun-Ming Chen, Tzu-Hsuan Wei, Kewei Lu, Xiaotong Liu, Xin Tong, Jiayi Xu, Haoyu Li, Jingyi Shen, Yamei Tu, and Neng Shi. I have been fortunate to meet a lot of wonderful people and make lasting friendships during the course of my graduate studies. The time I spent with my friends outside of research has helped me cope with the stress and rigor of graduate life in a foreign country. I would like to thank my friends Arjun Bakshi, Sourav Chakraborty, Deblin Bagchi, Rajaditya Mukherjee, Siddhartha Bora, Dhruba Jyoti Deka, and Angshuman Kapil. Special thanks go to my friend Ayana Ghosh for taking out the time to help me in reviewing and proof-reading my dissertation draft.

Most importantly, this dissertation and my doctoral studies would not have been possible without the unconditional love and support of my father Kanak Chandra Hazarika, who passed away in the summer of 2016, my mother Chikuni Hazarika, and my sister Latasri Hazarika. I will always be indebted to them for the scarifies they have made to fulfill my dreams. My deepest gratitude goes to my brother-in-law Nalinakshya Kalita and his family for the love and support that they have bestowed upon me over the years. I would like to thank my three-years old niece, Jisa, who can liven up my mood any time of the day with

her sweet little smile. Last but not the least, I would also like to thank all my close relatives back home, who have been supporting me and my family through thick and thin. I consider myself truly lucky and blessed to have these wonderful people in my life supporting my academic journey.

Vita

June 7, 1988	Born - Nagaon, Assam, India
2007 - 2011	B.Tech., Computer Science and Engineering, National Institute of Technology, Durgapur, WB, India
May - August, 2010	Summer Intern, CERN, Geneva, Switzerland
2011 - 2013	Software Engineer, Novell Inc., Bangalore, India
2014 - 2016	Graduate Teaching Assistant, The Ohio State University
May - July, 2017	Graduate Research Intern, Los Alamos National Laboratory
May - August, 2019	Graduate Research Intern, Los Alamos National Laboratory
May 2019	M.S., Computer Science and Engineering, The Ohio State University
2016 - present	Graduate Research Assistant, The Ohio State University

Publications

Research Publications

Subhashis Hazarika, Haoyu Li, Ko-Chih Wang, Han-Wei Shen, Ching-Shan Chou "NNVA: Neural Network Assisted Visual Analysis of Yeast Cell Polarization Simulation". *IEEE Transactions on Visualization and Computer Graphics*, 2019 (Early Access). **[IEEE VAST 2019 Best Paper Honorable Mention]**

Subhashis Hazarika, Soumya Dutta, Han-Wei Shen, Jen-Ping Chen "CoDDA: A Flexible Copula-based Distribution Driven Analysis Framework for Large-Scale Multivariate Datasets". *IEEE Transactions on Visualization and Computer Graphics*, 25(1): 1214-1224 (2019)

Junpeng Wang, Subhashis Hazarika, Cheng Li, Han-Wei Shen "Visualization and Visual Analysis of Ensemble Data: A Survey". *IEEE Transactions on Visualization and Computer Graphics*, 25(9): 2853-2872 (2019)

Subhashis Hazarika, Ayan Biswas, Han-Wei Shen "Uncertainty Visualization Using Copula-Based Analysis in Mixed Distribution Models". *IEEE Transactions on Visualization and Computer Graphics*, 24(1): 934-943 (2018)

Subhashis Hazarika, Ayan Biswas, Soumya Dutta, Han-Wei Shen "Information Guided Exploration of Scalar Values and Isocontours in Ensemble Datasets". *Entropy 2018, (Special Issue Information Theory Application in Visualization)*, 20(7), 540.

Subhashis Hazarika, Soumya Dutta, Han-Wei Shen "Visualizing the Variations of Ensemble of Isosurfaces". *IEEE Pacific Visualization Symposium (PacificVis)*, 209-213, (2016).

Fields of Study

Major Field: Department of Computer Science and Engineering

Studies in:

Computer Graphics & Visualization	Prof. Han-Wei Shen
Artificial Intelligence	Prof. Eric Fosler-Lussier
High Performance Computing	Prof. P. Sadayappan

Table of Contents

List of Tables

List of Figures

Figure **Page**

Chapter 1: Introduction

1.1 Background and Motivation

Over the past few decades, *computational science* has emerged as a significant field of research for conducting scientific studies alongside more traditional and well-established areas like *theoretical* and *experimental sciences*. It involves application of different computational and numerical techniques to solve large-scale scientific problems utilizing state-of-the-art computing resources. Computational scientists often design complex mathematical/numerical models to simulate various real-world physical phenomenon that they want to study in detail.

Recent advances in the field of computational sciences (numerical techniques, optimization algorithms, etc.) and high-performance computing (parallel architectures, distributed algorithms, etc.) have enabled researchers to execute their simulation models at very high spatial and temporal resolutions. Such high-resolution large-scale simulations often generate data in the scale of terabytes($\sim 10^{12}$), petabytes($\sim 10^{15}$), or even exabytes($\sim 10^{18}$) in near future [1, 164]. Unfortunately, current storage and input/output (I/O) technologies are not at par with ever-increasing computing speed and massive data sizes. Owing to constraints like disk-storage restrictions and I/O bandwidth limitations in supercomputing environments,

analyzing and visualizing the large-scale data generated from these simulations can become computationally prohibitive and non-trivial activities.

These simulation models have become an intricate part of the overall process of scientific studies in a variety of fields ranging from astrophysics to molecular biology, and from geology to environment science, just to name a few. Analyses and visualization of the results of simulations are important to gain key scientific insights while facilitating novel discoveries. Therefore, to utilize the wealth of information captured by high-resolution datasets and harness the power of such large-scale simulation models, it is important to develop novel as well as intelligent data analysis/visualization strategies.

One of the popular approaches to address these issues is to adopt *in situ*-based strategies, where important data analysis and/or visualization tasks are performed during the simulation execution time (i.e, when the generated data still resides in the memory). This can significantly bring down the analyses overhead by writing out only analysis/visualization results to storage-disk instead of the high-resolution raw datasets generated from large-scale simulations. Another common strategy to facilitate analysis of computationally expensive simulation models is to design a lightweight *surrogate model*, mimicking the original expensive simulation. The surrogate model can then be used to perform interactive exploratory analysis and visualization activities which requires frequent execution of the simulation to study its properties.

Besides the challenges arising from the scale and time complexity of these simulation models, there can be additional complexities associated with the *type of data* being simulated. These can make analysis/visualization tasks very cumbersome and overwhelming for scientists. Multiple facets/attributes of data may be involved in the analysis process. For example, while modeling many real world phenomena, scientific simulations generally measure more

than one physical variables (pressure, temperature, precipitation, etc). Such large-scale *multivariate dataset* has the additional challenge of considering the variable relationships and performing different multivariate analyses in a scalable manner. Yet another set of complexities can arise when scientists try to model the underlying uncertainty in the experiment by running the same simulation multiple times with different input parameter settings and/or initial conditions. The resulting simulation data is referred to as an *ensemble dataset*. In many newly designed simulation models, scientists are also interested in analyzing the simulation input parameters along with the corresponding simulated results. Such parameter space analyses can pose different set of challenges in computationally expensive simulation models.

In this dissertation, we have proposed different statistical and machine learning based approaches to address the challenges and concerns of performing data analysis and visualization activities for large-scale scientific simulation data.

1.2 Problem Statements

We have primarily identified three specific problems concerning data analysis and visualization of large-scale simulation data which are addressed and elaborated in this section.

1.2.1 Multivariate Distribution-based Data Summarization

A popular and effective approach for dealing with large-scale scientific data is to use *statistical probability distributions* in the analysis and visualization pipeline. Distributions like Histogram, Gaussian Distributions, Gaussian Mixture Models (GMM), Kernel Density Estimates (KDE) have been widely used to model large-scale data in the field of scientific data analysis and visualization. Distribution-based data representation offer two significant

benefits. *First*, storing probability distributions instead of original raw data help reduce the overall storage footprint for large-scale datasets. *Second*, many feature-based and query-driven analysis and visualization tasks rely on computing local data statistics, which can be easily evaluated from local distributions without accessing the original raw data.

However, while dealing with multivariate data the above benefits are not always applicable for standard multivariate distributions. Unlike their univariate counterparts, it becomes increasingly difficult to work with the corresponding standard multivariate distribution representations when the dimensionality increases. Some potential disadvantages of using standard multivariate distributions for modeling multivariate data in large-scale scientific simulations can be categorized as follows:

1. **Storage Footprint:** The storage footprint of a multivariate histogram can increase exponentially with number of variables, making them ineffective for data summarization. Although a sparse representation of multivariate histogram can reduce the exponential storage size, still, compared to the size of the raw data, it is still not useful for the purpose of data reduction. Moreover, the size of such sparse representations is sensitive to how the data is distributed and the number of histogram bins used.

2. **Estimation Time:** GMM is another popular data summarization alternative because of its compact representation and good modeling accuracy. However, the estimation of multivariate GMM using expectation-maximization is computationally very expensive compared to its univariate counterpart. The computation time increases rapidly with the number of variables. Therefore, despite the storage advantages, the high estimation times of multivariate GMMs will overshadow any I/O bottleneck alleviation, making them infeasible for multivariate data summarization in *in situ* applications.

3. **Flexibility:** Standard multivariate distributions are very rigid with respect to the assumptions in regard to their corresponding univariate distributions. For example, in a multivariate histogram, the individual variables are also histograms (i.e, marginal histograms) and a multivariate GMM with 3 modes always assume that the individual variables are modeled by univariate GMM with 3 modes. However, if a certain variable can be modeled by a simple Gaussian distribution with sufficient confidence, then, by using a Gaussian distribution (which requires storing just two parameters) instead of a distribution with more parameters to store, we can achieve higher levels of data reduction without compromising on quality. Such flexibility is not offered implicitly by the standard multivariate distributions.

Standard multivariate distribution models are not flexible enough to meet all of the above requirements at the same time. Therefore, to reap the benefits of distribution-based analysis and visualization solutions for large-scale multivariate simulation data, there is a need to adopt a different multivariate distribution modeling strategy.

1.2.2 Uncertain Analysis in Ensemble Simulation Data

The lack of knowledge about the ground truth of the simulated physical phenomenon often forces the scientists to execute the same simulation model multiple times using different initial conditions and/or different simulation input parameter configurations to get an estimate of the possible real outcomes [23, 108, 134]. The corresponding collection of simulation outputs is generally referred to as *ensemble* simulation data. Ensemble datasets are one of the primary sources of uncertain datasets in scientific studies. Each independent output in the collection is called an ensemble *member* or a *realization*, and depending on the study being conducted, ensemble members may have varying degrees of correlation among

themselves. For large-scale scientific ensemble simulations, with individual high-resolution ensemble members, performing uncertainty analysis and visualization tasks can become particularly challenging and cumbersome for the scientists.

Ensemble data analysis and visualization techniques can be broadly classified into two categories [129], i.e, *location-based* and *feature-based*. Location-based techniques are directed towards modeling and analyzing uncertainty of different spatial locations in simulation domain. Whereas, feature-based techniques analyze uncertainty associated with a specific feature of interest. In this work, we address challenges concerning both the categories of ensemble data analysis and visualization techniques.

1. **Challenges in Location-based Techniques:** The multiple realizations/values at each spatial location in ensemble datasets represent uncertainty in that location and are often modeled as stochastic random variables. Statistical distribution models can then be created to extract and visualize uncertain features from ensemble datasets. To preserve the correlation among the different spatial locations in the dataset, various standard multivariate distribution models have been proposed in visualization literature. Standard multivariate distributions (both parametric and nonparametric) assume that all of its univariate marginals are of the same type/family of distribution. But in reality, different spatial locations show different statistical behavior which may not be modeled best by the same type of distribution. Moreover, as discussed in Section 1.2.1 above, it can be very challenging to work with standard multivariate distribution model in large-scale simulations. Therefore, there is a need to adopt a different multivariate strategy to model uncertainty in scientific datasets, which is flexible enough to model the individual random variables at different spatial locations with different types of

distributions as well as be able to model the multivariate dependency among the random variables.

2. **Challenges in Feature-based Techniques:** Analyzing and visualizing uncertain isocontours/isosurfaces have become popular to effectively explore scalar ensemble datasets. Various techniques such as contour-boxplot [183], circular glyphs [158], contour variability-plot [56], probabilistic marching-cubes [144] have been proposed over the years to visualize ensemble isocontours. All of these techniques assume that a scalar value of interest is already known to the user. However, not all scalar values have the same degree of isocontour uncertainty and therefore, scientists as well as visualization practitioners may not have a clear idea of which scalar values to select for uncertainty analysis. Not much work has been done in guiding the users to select the interesting scalar values for such uncertainty analysis. Moreover, for a selected scalar value, individual members do not contribute equally to the overall uncertainty of the ensemble isocontour structure. The existing ensemble isocontour analysis techniques do not offer insights into the contribution of individual members towards the uncertainty. In short, a single coherent analysis framework that analyzes the uncertainty of both scalar value range as well as ensemble isocontours of an individual scalar value is mostly missing.

1.2.3 Simulation Parameter Analysis and Exploration

Despite modeling real-world physical phenomena with high degree of accuracy, large-scale scientific simulations often tend to be computationally very expensive. They also involve a large number of simulation input parameters which require detail investigation and analysis. Specially for newly designed simulation models, the simulation input parameters

need to be thoroughly analyzed and properly calibrated before the models can be applied for real scientific studies. Scientists need to get a clear picture of how different input parameters are influencing simulation outcomes under various scenarios. This requires performing exploratory analysis tasks, which involve repeated execution of the expensive simulations on new and unseen parameter configurations to study simulation characteristics. For compute-intensive simulation models with high-dimensional input and output spaces, this can become a computationally prohibitive and non-trivial analysis task. Sufficient computational resources need to be allocated to execute these simulation models. Each execution of the simulation may take hours to converge, thus, making regular exploratory analyses all the more difficult and overwhelming for the scientists.

A popular and effective strategy in the scientific community to address such issues has been to construct a statistical/mathematical *surrogate model*. A surrogate model, built using a finite set of simulation results tries to model the output of the complex simulation from the possible input parameter space. Various machine learning models are commonly used for constructing such surrogate models. The trained surrogate model can then replace the original expensive simulation during the analysis process for rapid parameter space exploration. Guided by the research methodology of *analysis-by-synthesis* [18], surrogate models can assist in many data analysis and visualizations tasks. However, the choice of the surrogate model is important to facilitate effective simulation parameter analysis. Sedlmair et al. [163] proposed a conceptual framework to categorize various analysis tasks and navigation strategies that are desired in visual parameter space analysis system. They described a set of six analysis tasks: *optimization*, *uncertainty*, *sensitivity*, *partitioning*, *fitting*, and *outliers*. They also identified four navigation strategies: *informed trial and error*, *local-to-global*, *global-to-local*, and *steering*. Therefore, it is important to choose a

surrogate model which can support most, if not all, of the above identified analysis tasks and navigation strategies for effective simulation parameter analysis.

1.3 Proposed Solutions

In this dissertation, we have proposed different statistical and machine learning based solutions to address the problem statements elaborated in Section 1.2 above. In this section, we briefly explain the different solutions proposed in our work and the corresponding problem statements that they help in addressing.

1.3.1 Copula-based Multivariate Distribution Modeling

As explained in Section 1.2.1 for multivariate data summarization and Section 1.2.2 for location-based uncertainty modeling, working with multivariate distributions can be challenging for large-scale simulation data. Therefore, we proposed a copula-based approach to model multivariate distributions rather than using standard multivariate distribution models. Copula functions offer a statistically robust mechanism to decouple the process of multivariate distribution estimation into two independent tasks: *univariate distribution estimation* and *dependency modeling*. As a result, the exponential cost of storage and/or distribution estimation time can be reduced significantly because we can independently model the individual variables using arbitrary univariate distribution types, while the copula function captures the dependency among them separately. In Chapter 3, we explain in detail the concept of copula function and how it can be used to model multivariate distributions. Using our proposed copula-based approach, we performed *in situ* multivariate data summarization. The summaries are subsequently used to carry out scalable multivariate post-hoc analysis and visualization tasks. Details about this application is provided in Chapter 4. We

also used our approach to model uncertainty in ensemble datasets and visualize uncertain features like isosurfaces and vortices, which is covered in Chapter 5 of this dissertation.

1.3.2 Information-theoretic Uncertainty Analysis Framework

As discussed in Section 1.2.2, with feature-based uncertainty analysis and visualization techniques for ensemble data, there is a need to guide the scientists as well as visualization practitioners towards identifying interesting features for further analysis. To address this requirement, we have proposed a two-stage information-theoretic framework for the exploration of scalar values as well as their corresponding ensemble isocontours. Using *specific information* measures like *predictability* and *surprise* of specific scalar values, we evaluate the ensemble isocontour uncertainty of all the scalar values in an efficient way. *Predictability* of a scalar value conveys the relative similarity of the corresponding ensemble isocontours, while, *surprise* conveys the relative importance of the scalar values in the field. Moreover, for a single scalar value, we proposed a *conditional entropy* based approach to identify the contribution of individual members towards the overall uncertainty of the ensemble isocontours. By accounting for the information overlap among their corresponding member isocontours, conditional entropy helps us to effectively measure the relative importance of the individual ensemble members. Detail discussion about this approach and its application is provided in Chapter 6 of this dissertation.

1.3.3 Neural Network Assisted Visual Analysis

For computational expensive simulation models, as discussed in Section 1.2.3, it is important to choose a proper surrogate model which can assist in performing simulation parameter analysis and exploration. To address this, we proposed the use of neural network-based surrogate models to facilitate the design of interactive visual-analytic frameworks for

parameter analysis. Besides accurately predicting the output of high-dimensional non-linear functions, a trained neural network can also be utilized to extract and analyze interesting properties about the original simulation. We collaborated with computational biologist to design an interactive visual analysis framework, backed by a neural network-based surrogate model, which can assist them in analyzing and visualizing a complex yeast cell polarization simulation. During the exploration stage, using the trained surrogate model, scientists can quickly preview the predicted simulation results for new parameter configurations instead of running the expensive simulation model every time. We incorporated *parameter sensitivity*, *parameter optimization* and *uncertainty quantification* techniques to visually guide the scientists towards discovering new parameter configurations of interest. In Chapter 7, we discuss in detail about our neural network assisted visual analysis approach for yeast simulation model.

Chapter 2: Related Works

In this chapter, we provide a literature survey of the various related works in the field of scientific visualization and data analysis which are relevant to our proposed work. We categorize them into the following research sub-fields.

2.1 Distribution-Driven Analysis and Visualization

Statistical probability distributions have been widely used in the field of scientific data analysis and visualization. Liu et al. [107] exploited GMMs for stochastic sampling-based volume rendering on the GPU. Lundstrom et al. [111] studied the design of transfer functions in direct volume rendering based on local histograms. Statistical distribution fields have been visualized by displaying distribution properties like mean, standard deviation and skewness using color channels, height maps and glyphs [90, 112, 145, 146]. Distributions have also been widely used to model uncertainty in scientific datasets. Jarema et al. [83] used directional distributions to perform comparative visual analysis of vector field ensembles. With respect to distribution-based data summarization for large-scale data, Thompson et al. [172] proposed Hixels, which stores histogram per data block to preserve the statistical properties of data. Dutta et al. [45, 49] stored GMMs per data block to track time-varying uncertain features. They also proposed a homogeneity preserving data partitioning scheme [47], where the local data was modeled using a hybrid mixture of Gaussian distributions and GMMs.

Wang et al. [181] stored spatial GMMs per bin of the local data histogram to achieve good reconstruction results. Almost all of these distribution-based data summarization works are targeted for univariate dataset. In this dissertation, we proposed a framework to facilitate distribution-based data summarization for large-scale multivariate data.

2.2 *In situ* Data Modeling and Analysis

With increasing sizes of scientific simulation data, *in situ* data processing is becoming increasingly popular for scalable analysis and visualization tasks. Bauer et al. [14] performed a comprehensive survey of the *in situ* visualization techniques. Direct visualization of the simulation data can be performed with LibSim using VisIt [184] and CATALYST using Paraview [55]. Vishwanath et al. [173] in their work, GLEAN, improved the process of *in situ* analysis. Yu et al. [195] performed *in situ* visualization of combustion data. Woodring et al. [189] proposed an *in situ* eddy census for ocean simulation models. However, exploratory data analysis tasks, which require back-and-forth interaction with the raw data are not feasible with pure *in situ* techniques [46]. To address such limitations, recently, a new *in situ* practice has been gaining popularity, where, large-scale data is statistically summarized and later used for post-hoc analysis using the data summaries rather than the raw data [36,99]. An *in situ* image-based approach was used by Ahrens et al. [8] for post-hoc feature exploration. Woodring et al. [188] adopted a sampling-based method to visualize Cosmology data. To facilitate interactive post-hoc visualization of particle data, Ye et al. [194] computed probability distribution functions *in situ*. Dutta et al. [45, 47] performed *in situ* estimation of combinations of GMMs to create data summaries, which are later used for post-hoc

feature exploration. To the best of our knowledge, similar approaches to facilitate post-hoc multivariate analysis on large-scale multivariate data does not exist. Our proposed multivariate data modeling strategy is targeted to address this issue.

2.3 Multivariate Analysis and Visualization

Multivariate analysis and visualization is a well-researched topic in the field of scientific visualization. Wong et al. [186] and Fuchs et al. [59] provided an extensive review of the multivariate data analysis and visualization techniques in the field. Sauber et al. [159] studied the local correlation coefficients among the variables to analyze and visualize multivariate data. Bethel et al. [17] computed correlation fields to perform query-driven analysis with multivariate data. Gosnik et al [67] used local statistical distributions to improve query-driven analysis for multivariate data. Jänicke et al. [82] adapted local statistical complexity to identify informative regions in multivariate data. Creating efficient multivariate distributions have always been a challenging task. Various compact representations of the multivariate joint histogram have been proposed to tackle the curse of dimensionality [28, 109].

2.4 Uncertainty Analysis and Visualization

Uncertainty analysis and visualization of scientific datasets is considered as one of the top few challenges in our field [85, 86, 185]. Over the past few years, there have been significant research contributions towards visualizing and modeling uncertain data [25, 134, 147]. Here, we specifically discuss only the works that use statistical distributions to model uncertainty and are related to our proposed strategy. Techniques were proposed to visualize datasets where each grid locations have data distributions rather than single data point [108, 112, 131]. The use of distributions to model the uncertainty in data and its subsequent analysis to extract

probabilistic features have gained popularity in the recent past. Pöthkow et al. [141] proposed the concept of level-crossing probability (LCP) to compute probabilistic isocontours in uncertain data. LCP computes the probability of an isocontour passing through a cell of the data. It assumed that the data at each grid location follows a Gaussian distribution and there is no correlation among the grid location. This approach was later extended to introduce the local spatial correlation [144]. Methods have also been proposed to extract and analyze features like vortices and critical points using statistical distributions as uncertainty modeling tools [103, 118, 132, 133, 136]. All these works assumed that the data at each grid location follow a Gaussian distribution. Pöthkow et al. [142] later extended uncertainty analysis to include nonparametric models. Athawale et al. [11, 12] proposed closed-form analytic solution to compute uncertain isocontours in nonparametric distribution models. Pfaffelmoser et al. [138] performed detailed study on the properties of global and local correlation in uncertain data. Schlegel et al. [160] proposed Gaussian process regression based interpolation scheme and investigated the influence of correlation functions on the level-crossing probabilities in Gaussian random field. Dutta et al. [49], on the other hand, used GMMs to perform feature tracking in time varying data. Despite its flexibility, it is computationally very expensive to estimate GMMs for multivariate models where correlation has to be accounted for. In general, depending on the modeling scenario, all distribution types have their own advantages and disadvantages. To the best of our knowledge, none of the current distribution-driven feature analysis works try to utilize the benefits of using different distribution types to model the uncertainty at different locations. Our proposed copula-based multivariate modeling strategy facilitates such flexibility in use of distribution models while preserving the local spatial dependency at the same time.

2.5 Ensemble Data Visualization

Visualization of ensemble data falls in the general category of uncertainty visualization. The field of uncertainty visualization has seen a lot of innovations over the past two decades [43, 86, 108, 134]. Potter et al. [147] provided an extensive survey of the sources of uncertainty in data as well as possible visualization based answers for analyzing them in different dimensions. Ensemble dataset is a special category of uncertain datasets where data is generated from multiple simulations or runs with varying parameter settings. This type of dataset is very popular in the field of weather forecasting and simulation sciences [120, 121]. Obermair et al. [129] categorized the different ensemble visualization techniques based on their approaches and discussed the possible future challenges in ensemble visualization. Wang et al. [177], in a recent survey of visualization and visual analysis techniques for ensemble data, provide a structured view of the general approaches of dealing with ensemble data analysis. Potter et al. [148] built a comprehensive framework called Ensemble-Vis to visualize 2D weather forecasting and climate modeling ensembles using multiple statistical visualization techniques. Demir et al. [41] tried addressing the challenges of 3D ensemble data by using multi-chart visualizations. For ensemble datasets, effective isocontour visualization is a very challenging task. In meteorology, spaghetti plots are commonly used to display simultaneously all the ensemble isocontours. Sanyal et al. [158] introduced a tool called Noodles which enhanced spaghetti plots by using circular glyphs and confidence ribbons to highlight the spread of isocontour lines. Alabi et al. [9] proposed Ensemble Surface Slicing (ESS) to show the variation of an ensemble of isosurfaces. Hazarika et al. [73] visualized the order-statistics of ensemble isosurfaces of multiple isovalues using parallel-coordinate systems. Attempts have been made to create uncertain isocontours using the probabilities of level-set crossing [71, 141, 143]. This led to the concept of probabilistic

marching cubes by Pothkow et al. [144], used to extract uncertain isosurfaces from a distribution field. To address the quantitative aspect of ensemble isocontour visualizations, Whitaker et al. [183] proposed contour boxplot to visualize the statistical properties, outliers and other variabilities of contours. Recently, Ferstl et al. [56] proposed a contour variability plot by clustering groups of ensemble isocontours. In all these uncertainty analysis techniques, it is assumed that a certain scalar value of interest is already known to the users. To the best of our knowledge, not much work has been done to help the users in selecting such scalar values for uncertain isocontour analysis.

2.6 Copula-based Statistical Modeling

The relationship between a generic multivariate function and a copula function was first formalized by Sklar in 1959 [166]. Since then it has been widely used as a robust statistical tool for multivariate data modeling. In the article titled, *Coping with Copula* [161], Schmidt provides a detailed explanation of the workings of copula functions and their potential application in various fields. Copula functions have been widely used in the field of financial modeling and risk analysis [35, 53, 116, 126]. Over the past few years, copula functions, especially, Gaussian copula, have been gaining popularity in the field of machine learning as well, for the purpose of modeling high-dimensional distributions [50, 153]. Machine learning approaches like dimensionality reduction [69, 70], mixture modeling [60, 171], component analysis [93, 113] and clustering [154] have benefited from the flexibility offered by copula functions. Besides, copula related analysis have been used in the field of Uncertainty Quantification as well [13]. The multivariate distribution modeling flexibility offered by Copula functions have not been fully utilized in the field of scientific visualization and analysis. Our proposed work demonstrates the usefulness of a copula-based approach for

in situ data modeling of large-scale multivariate simulation [72] as well as uncertainty modeling of ensemble simulation data [71].

2.7 Information theory in Visualization

A large number of visualization and computer graphics problems have been solved using information theory [39]. The recently published book, titled, *Information Theory in Visualization* [34], covers in great details how information theory helped solve many challenging problems in visualization. For time-varying datasets, Wang et al. [175] performed an information based block-wise analysis to identify important time varying features. Chen and Janicke [33] provided evidences to show that information theory can be used to analyze many visualization problems. For flow visualization, an information theoretic framework was provide by Xu et al. [192] to evaluate the effectiveness of visualizations in communicating the original data information to the users. One popular information-theoretic measure is mutual information. Mutual information has been used in the medical image registration and multi-modal data analysis for a long time now [80, 140]. Bruckner et al. [27] used mutual information to measure the similarity of isosurfaces in scientific datasets and proposed the isosurfaces similarity map to identify salient isosurfaces. Wei et al. [182] used similar mutual information based method to evaluate isosurfaces for surface morphing. Specific information measures are essentially decompositions of mutual information. I_1 and I_2 specific information measures were first introduced in the works of DeWeese and Meister [42]. Bramon et al. [24] used them to perform fusion of multi-modal images. Dutta et al. [48] used mutual information and its various decompositions to select important isosurfaces in multivariate time-varying datasets. Biswas et al. [21] proposed an information-theoretic framework for multivariate data analysis. They used mutual information, specific

information and conditional entropy to identify salient isocontours in multivariate datasets. In the field of pattern recognition and feature selection, various mutual information based tools and techniques have been proposed to optimally select features based on criterion like minimizing redundancy and maximizing relevance [26, 128]. However, techniques for feature analysis and selections which meet the needs of ensemble scientific data is mostly missing. In our work, we propose an information theoretic approach towards understanding the effect of uncertainty in scalar values and their features (isocontours) in ensemble datasets.

2.8 Visual Exploration of Parameter Space

Over the years, multiple visual analysis systems have been proposed to facilitate interactive visual exploration of the input parameter space for simulation models. Each of them are application specific, and caters to the requirements of their domain experts. Orban et al. [130] projected input parameters and output data in material science to 2D spaces, and allowed users to manipulate in the input space and observe the change in the output space. Wang et al. [180] developed a nested parallel coordinate plot for parameter analysis of multi-resolution climate ensemble datasets. Biswas et al. [20] used a Gaussian Process-based surrogate model to perform interactive exploration in a shock physic application. Coffey et al. [37] designed an interface which uses a mapping between model features and simulation inputs to enable direct simulation input parameter manipulations. Piringer et al. [139] proposed an approach called HyperMoVal, which can evaluate the bad fit of surrogate models and provide visual validation for their physical plausibility. Berger et al. [16] proposed an uncertainty-aware statistical approach to predict results of given parameters for real-time analysis.

Sedlmair et al. [163] provided an extensive survey and proposed a conceptual framework to categorize the various analysis tasks and navigation strategies used in such visual analysis systems. Our proposed visual analysis system, discussed in Chapter 7, encompasses three of the six analysis tasks formalized by Sedlmair et al., namely, *optimization*, *uncertainty*, and *sensitivity*. Among the four navigation strategies that they identified, our system covers two of them, namely *informed trial and error* and *local-to-global*. Simulation parameter analysis is also a popular topic in scientific visualization community. Parameter sensitivity analysis techniques [52, 94, 124] have been widely used to perform various uncertainty-aware scientific analysis and visualization [22, 68, 193]. The recent survey on visualization techniques for ensemble simulation data by Wang et al. [177] also covers the sub-category of simulation parameter analysis in the visualization community.

2.9 Machine Learning and Visual Analysis Symbiosis

There has always been a symbiotic relationship between the research areas of Machine Learning and Visualization. In this section, we cover the related previous works in the area which emphasis this mutual relationship.

2.9.1 Machine Learning Models for Visual Analysis

Visualization community often uses machine learning techniques to enhance their visual analytic tools [54]. Machine learning models act as a medium to extract interesting insights about the data, which is then presented to the end users through interactive visual analytic systems. Besides enhancing the data-analysis experience, this acts as a platform for users without much machine learning background to reap the benefits of sophisticated machine learning models. Among the recent neural network-based models, CNN (Convolutional Neural Network) [191] and Word2Vec [110, 197] models have been used to create interactive

visual analysis systems for different application domains. Moreover, traditional models like SVM (Support Vector Machine) [190], LDA (Latent Dirichlet Allocation) [98, 106], KNN (K Nearest Neighbor) [115], Bayes' rule [65], learning-from-crowds model [105], and online metric learning [102] have also been extensively utilized by visualization researchers to enhance the data-analysis experience in their systems. Along similar lines, our proposed system utilizes a trained multilayer perceptron model to design an interactive visual analysis framework for a scientific application.

2.9.2 Visual Analysis for Machine Learning Models

Since the past few years, the visualization community has played a significant role in explaining the inner workings of complicated machine learning models. Multiple visual analytic tools have been developed to visualize different machine learning algorithms, such as Adaboost, SVM, decision tree, and random forest [38,77,81,157,174]. Recently, Hohman et al. [79] published a comprehensive survey on the various visual analytic approaches to explain deep neural network models, which are gaining significant popularity in the machine learning community. At a high-level, we can divide these approaches into three categories. The goal of one category of visualization tools is to open the black box by interpreting the trained model [89, 104, 119, 169, 179, 187]. While, another category of visualization tools not only interpret, but also diagnose the trained model [88, 168, 176, 196]. Recently, a third category of visualization tools, focusing more on assisting the machine learning experts to improve their models is gaining popularity [19, 178].

Besides the visualization community, the machine learning community is also working in parallel to create various post-hoc analysis techniques to interpret and explain complicated models. Unlike most of the visualization tools, these post-hoc analysis functions are intended

to be more generic and applicable for different network architectures. Montavon et al. [122] covers in great details the various post-hoc analysis techniques that can be performed on trained neural networks to make them more interpretable and explainable. These analysis techniques for trained neural networks have seen wide-spread application in scientific domains, ranging from cancer diagnosis to quantum physics [10, 63, 64, 91, 162, 170]. In our proposed visual analysis system, we use different post-hoc analysis functions on the trained neural network-based surrogate model to study and analyze the original yeast simulation. We also visualize the network structure (weight matrices) to extract and validate the knowledge learned by the surrogate model during the training process.

Chapter 3: Copula-based Multivariate Distribution Modeling

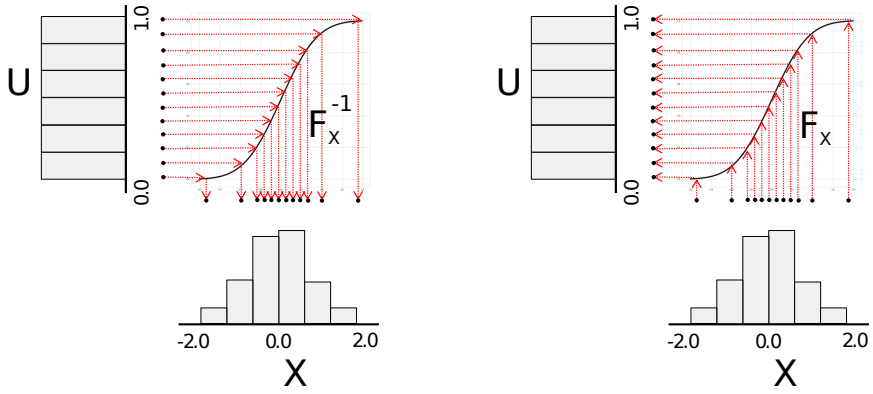
The term *copula* was first used in the work of Sklar [166] and is derived from the latin word *copulare*, to connect or to join. Copula functions are used as tools for modeling dependence/interrelationship of several random variables. The idea of copula is closely tied with the definition of multivariate distribution functions. In the subsequent sections of this chapter, we first revisit some of the important definitions and properties of multivariate distributions which are relevant to understand the concept of copula and then formally introduce the copula functions along with an example.

3.1 Multivariate Distribution

Consider a set of d real valued random variables, $X_1, X_2, ..., X_d$. The joint multivariate cumulative density function (CDF) is defined as the probability of the random variable X_i taking values less than or equal to x_i i.e;

$$F(x_1, x_2, ...x_d) \stackrel{\text{def}}{=} P(X_1 \leq x_1, X_2 \leq x_2, ...X_d \leq x_d) \tag{3.1}$$

where, x_i is a realization of the random variable X_i and $P(.)$ is the probability function. The joint CDF, $F : \mathbb{R}^d \rightarrow [0,1]$, maps the multivariate random variable to a scalar value in between 0 and 1. Similarly, a univariate CDF of the random variable X_i can be denoted as, $F_i(x_i) \stackrel{\text{def}}{=} P(X_i \leq x_i)$.



(a) $F_X^{-1}(U) \sim X$ (b) $F_X(X) \sim U$

Figure 3.1: Property of CDF: (a) If we know the inverse CDF F_X^{-1} of a distribution of variable X, we can always transform uniform samples to follow distribution of X (b) The output of a continuous CDF F_X, is always a uniform distribution $U[0,1]$

3.2 Distribution Transformation Properties

A univariate CDF has two interesting distribution transformation properties [116], which are frequently used in copula-based techniques to model multivariate distributions with arbitrary types of univariate distributions.

- ***Property 3.2.1*:** If U is a uniform random variable (i.e, $U \sim U[0,1]$) and F_X is a univariate CDF, then its inverse function, $F_X^{-1}(U)$, corresponds to the random variable X, (i.e, $F_X^{-1}(U) \sim X$)

$$P(F_X^{-1}(U) \leq x) = F_X(x) \tag{3.2}$$

- ***Property 3.2.2*:** If a real valued random variable X has a continuous cumulative distribution function F_X, then

$$F_X(X) \sim U[0,1] \tag{3.3}$$

Of the two stated properties, the former is extensively used in many statistical libraries to generate random numbers following a particular distribution. However, the second property, which states that we can transform any continuous CDF to a uniform distribution is not frequently used. Proofs of these two properties are provided in the appendix A. Figure 3.1 (a) and (b) illustrate the two distribution transformation properties respectively. The main take away from these properties is that, we can always transform uniformly distributed samples to a target distribution, say X, and vice versa, as long as we have a well-defined CDF and inverse CDF of the distribution of X.

3.3 Copula function

By definition, a *copula function* or a *copula* in general, is a multivariate cumulative density function (CDF) whose univariate marginals are uniform distributions. Mathematically, $C : [0,1]^d \rightarrow [0,1]$ represents a d-dimensional copula (i.e., d-dimensional multivariate CDF) with uniform marginals. For d-uniform random variables $u_1, ...u_d$, it can be also be denoted as $C(u_1, ...u_d)$. *Sklar's theorem* [166] formally established that every joint CDF in $\mathbb{R}^d$ implicitly consists of a d-dimensional copula function. If F is the joint CDF as shown in Equation 3.1 and $F_1, F_2, ...F_d$ are the marginal CDF's for a set of d real valued random variables, $X_1, X_2, ...X_d$ respectively, then Sklar's theorem can be formally represented as;

$$\begin{aligned} F(x_1, x_2 ... x_d) &= C(F_1(x_1), F_2(x_2), ...F_d(x_d)) \\ &= C(u_1, u_2, ...u_d) \quad \text{(using } \textit{Property 3.2.2}\text{)} \end{aligned} \tag{3.4}$$

Using Property 3.2.2 (Figure 3.1b), Equation 3.4 is equated to the standard copula notation, where, u_i represents the realizations of a uniform distribution $U[0,1]$. The complete Sklar's theorem is provided in appendix. If f is the multivariate probability density function (PDF) of the CDF F and f_i, the corresponding univariate PDFs of the CDFs F_i, then in terms

of probability density functions Equation 3.4 can be written as follows:

$$f(x_1,x_2...x_d) = c(F_1(x_1),F_2(x_2),...F_d(x_d))\prod_{i=1}^{d} f_i(x_i) \tag{3.5}$$

where,

$$c(u_1,...u_d) = \frac{\partial C(u_1,...u_d)}{\partial u_1...\partial u_d} \tag{3.6}$$

Therefore, Equation 3.4 and 3.5 imply that, given the univariate marginal CDFs and a copula function, we can derive the original multivariate CDF. This is the major idea behind many copula-based multivariate distribution modeling techniques. It is important to choose the right copula function C when modeling a multivariate distribution. Using the property $F_i \cdot F_i^{-1}(x) \geq x$, we can rewrite Equation 3.4 to compute C as follows;

$$C(u_1,...,u_d) = F(F_1^{-1}(u_1),...,F_d^{-1}(u_d)) \tag{3.7}$$

However, when the true multivariate distribution function is not known or well-defined, a popular technique is to estimate C using a standard multivariate distribution. Such copulas are called *implicit copulas*. The most popular implicit copula is the *Gaussian* copula, which is computed from multivariate Gaussian distribution. Gaussian copula is well-suited for the purpose of data reduction and uncertainty modeling in large-scale scientific datasets because it requires storing only the correlation matrix of the data, which can be efficiently computed in an *in situ* environment. In recent years, Gaussian copula have found wide-spread usage in the field of machine learning to perform dependency based clustering and classification in multivariate models [153]. We discuss more in details about the properties of Gaussian copula in the next section along with an example to show its practical usage in dependency modeling. However, it is important to know that there is a good number of other predefined copulas as well, referred to as *explicit copulas*. Explicit copulas (like *Gumbel*, *Clayton* and

Frank copula) are designed keeping in mind special statistical tasks at hand [53, 161]. They have gained sufficient popularity in the field of financial modeling and risk analysis.

3.4 Gaussian Copula

In Equation 3.4, if F is a standard normal distribution of d-dimensions, then the corresponding $C(.)$ is a Gaussian copula. For a d-dimensional standard normal distribution $\mathcal{N}_d(\mathbf{0}, \rho)$, with zero mean vector $\mathbf{0}$ and correlation matrix ρ, the corresponding Gaussian copula function C_ρ^G with the parameter ρ can be denoted as;

$$C_\rho^G(u_1, ...u_d) = \Phi_\rho(\Phi^{-1}(u_1), ...\Phi^{-1}(u_d)) \tag{3.8}$$

where, Φ^{-1} represents the inverse CDF of a standard normal distribution and Φ_ρ represents the CDF of a multivariate standard normal distribution with correlation matrix ρ. Standard normal distributions have well-known closed-forms for the CDF functions, therefore, we can easily compute the Gaussian copula function using equation 3.8, provided we know the correlation matrix ρ. Since the marginals of a copula function are uniform distributions, we can easily construct multivariate distributions with arbitrary marginal distribution types by transforming the uniform distributions to the target univariate distributions using the Property 3.2.1 illustrated in Figure 3.1(a). The final multivariate distribution, thus obtained, is often termed as *meta-Gaussian* [155] distribution since the dependency structure is Gaussian but the marginals can be arbitrary distributions.

3.5 Gaussian Copula-based Multivariate Sampling

In this section, we show how to use a copula-based approach to model and sample multivariate distributions using a simple bivariate example. Consider a multivariate sample of two random variables X and Y, with a strong negative correlation ($\rho = -0.9$). The

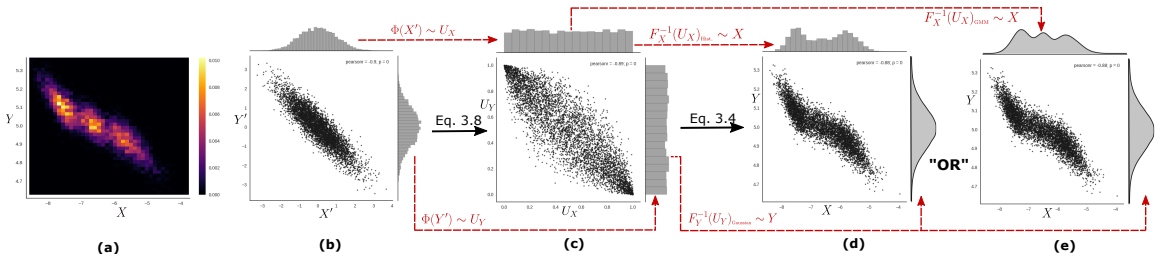


Figure 3.2: Copula-based sampling example: (a) Joint distribution of the original bivariate samples with correlation coefficient -0.9. (b) Step 1: Generate new bivariate samples from a bivariate standard normal distribution. (c) Step 2: Construct the Gaussian copula with uniform marginals. (d,e) Step 3: Final bivariate samples with arbitrary univariate distribution types. A histogram representation for Y in (d) and a GMM representation for Y in (e), while, X is being modeled by a Gaussian distribution in both the scenario.

original joint distribution of the two variables is shown in Figure 3.2(a). Let F_X and F_Y be the CDFs of the desired univariate distribution respectively. Since a copula-based approach is independent of the type of univariate distributions used, F_X and F_Y can be of any arbitrary type (Histogram, GMM or Gaussian). Given F_X, F_Y and ρ(=-0.9), the three steps involved in our sampling method are as follows:

- **Step 1:** Generate new multivariate samples from a *standard bivariate normal distribution* with the correlation matrix ρ. Figure 3.2(b) shows the scatter plot view of the generated samples. In this step, the samples only preserve their correlation, while the univariate marginals are standard normal distributions with mean value 0 and standard deviation of 1.

- **Step 2:** The output of a CDF always follow a uniform distribution as illustrated in Figure 3.1(b). Using this property, we transform the bivariate samples generated in Step 1 to a bivariate uniform distribution as shown in Figure 3.2(c). By equation 3.8, these transformed samples, generated from a bivariate standard normal distribution

represent the corresponding bivariate Gaussian copula. The dependency structure between the variables is still preserved but the marginals are uniform distributions.

- **Step 3:** Finally, we transform the uniform distributions of the two variables to the desired distribution types using the inverse functions of the precomputed CDFs F_X and F_Y. If we know the inverse CDF of a distribution, we can always transform uniform samples to the corresponding distribution, a fact, illustrated by Figure 3.1(a). As shown in Figure 3.2(d,e), the final bivariate samples (with sample $\rho = -0.88$) closely represent the initial bivariate samples. Since the transformation in this step takes place from uniform marginals, we can use arbitrary target univariate distribution for transformation. For example, F_X can be a Histogram (Figure 3.2(d)) or a GMM (Figure 3.2(e)), while F_Y is a Gaussian in the two alternatives.

For a d-dimensional multivariate system, we start Step 1 above with a d-dimensional standard normal distribution. Using this 3-step sampling strategy, we are able to generate multivariate samples from our proposed multivariate data summaries that preserve the correlation among the variables, an important property desired in any multivariate analysis task.

3.6 Storage Requirements

To summarize, our proposed copula-based multivariate distribution modeling approach involves storing the desired *univariate distributions for the individual variables* and their *Gaussian copula parameters (i.e, correlation matrix ρ).* Since, our objective is to reduce storage footprint, instead of storing the complete correlation matrix, ρ, which is a symmetric matrix, we store only the pairwise correlation coefficient of all the variables, which constitutes the lower and the upper triangles in the matrix. Therefore, for multivariate data with n

variables, the overall storage requirement of our proposed technique can be formalized as;

$$S = \sum_{i=1}^{n} m_i + \binom{n}{2} \tag{3.9}$$

where, m_i is the storage footprint of the univariate distribution chosen for the i-th variable, while $\binom{n}{2}$ is the cost of storing the Gaussian copula parameter. We can optimally choose univariate distribution models for individual variables depending on factors like individual storage footprint (m_i), modeling accuracy and computation times and estimate them in parallel.

Chapter 4: *In Situ* Copula-based Multivariate Data Summaries for Large-Scale Multivariate Datasets

4.1 Introduction

Scientists often measure multiple physical attributes/variables at the same time in their computational models. These variables are used to perform various multivariate analyses to gain in-depth insights into the underlying physical phenomenon. Recent advances in the field of high-performance computing have enabled scientists to simulate their computational models at very high resolutions, thus, generating data in the scale of terabytes or even petabytes. The multivariate nature of the simulation adds to the complexity of such large-scale scientific datasets, thereby, possessing significant challenges with respect to performing multivariate analysis and visualization tasks.

A popular and effective strategy for analyzing and visualizing large-scale scientific datasets is to first partition the simulation domain and then store statistical data summaries for each partition [36, 45, 47, 99]. This strategy is particularly useful in many *in situ* applications to alleviate issues like storage overhead and I/O bottleneck for large-scale data. Such applications create the data summaries *in situ* (i.e, while the simulation is still running) and write-out the compact statistical representation instead of the raw data. These summarized data representations are later used to perform post-hoc analysis and

visualization in a much scalable manner (even on commodity hardware). Such summaries, often represented in the form of various statistical probability distributions (Histogram, Gaussian Mixture Models, etc.) offer two significant benefits. *First*,storing probability distributions for local neighborhood helps reduce the overall storage footprint for large-scale datasets. *Second*, many feature-based and query-driven analysis and visualization tasks rely on computing local data statistics, which makes such statistical summaries a prudent choice for compact data representation [49, 67, 111, 145, 146]. However, for multivariate data, where it is important to preserve the multivariate relationship among variables, using standard multivariate probability distribution models for data summarization does not always yield similar benefits. They are either not space efficient for the purpose of data reduction (e.g multivariate histograms) or are computationally very expensive to estimate when the number of variables increases, thus, overburdening the actual simulation execution (e.g multivariate Gaussian Mixture Models). Therefore, there is a need to rethink how to model large-scale multivariate data, such that we still have similar benefits as univariate data summaries. Moreover, performing multivariate analysis tasks *in situ* may not always be helpful, especially, for exploratory analysis tasks [36], where, in the initial stages scientists usually do not have a clear understanding of the important variables to analyze and/or the precise value ranges to query for [46]. Such exploratory analysis involves back-and-forth interaction with the data, trying various choices before developing a clear idea. However, it is often computationally prohibitive to run large simulations in supercomputing environments multiple times for such exploratory analysis. Therefore, there is a real necessity to have a good multivariate data summarization solution for large-scale multivariate simulations, that can preserve the various multivariate relationships as well as be computationally efficient both with respect to storage footprint and estimation time.

In this chapter, we propose a flexible distribution-driven analysis framework (CoDDA: **Co**pula-based **D**istribution **D**riven **A**nalysis) for large-scale multivariate data that addresses the aforementioned concerns. In the first stage of our framework, to achieve a compact data representation, we partition the simulation domain and store the corresponding univariate distributions of the variables for each partition. The dependency among the variables for each partition is separately estimated using copula functions. To preserve the spatial information in our model, we also consider the spatial variables as extra dimensions along with the physical variables and store the corresponding spatial distributions in an efficient representation. In the second stage of our framework, to demonstrate the efficacy of our proposed multivariate data representation, we perform two broad categories of post-hoc multivariate analysis tasks using a copula-based sampling strategy. (a) For effective post-hoc visualization, we propose a multivariate sampling-based technique to create sample scalar fields of arbitrary user-specified grid resolutions. (b) For multivariate query-driven analysis tasks, we propose the computation of probabilistic multivariate queries from our data summaries. Besides evaluating our proposed data modeling strategy on two large-scale multivariate datasets, we also test our method in a real-world *in situ* scenario, by running it directly with a large-scale CFD simulation. We conduct both quantitative and qualitative assessment of our generated results and offer insights into various choices that we make.

4.2 System Overview

Figure 4.1 provides a schematic overview of the different stages of our proposed framework. The two main stages are: (a) data modeling/summarization, which can be performed *in situ* alongside the simulation and (b) subsequent post-hoc multivariate analysis using the constructed data summaries. The data modeling stage consists of first partitioning the

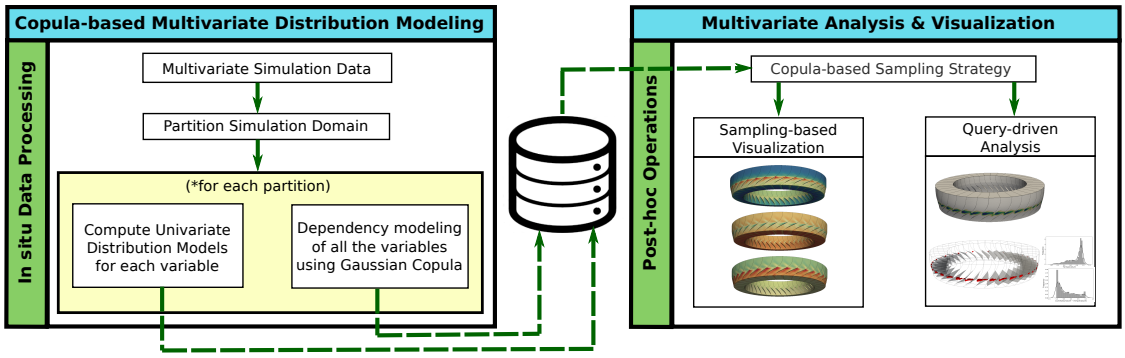
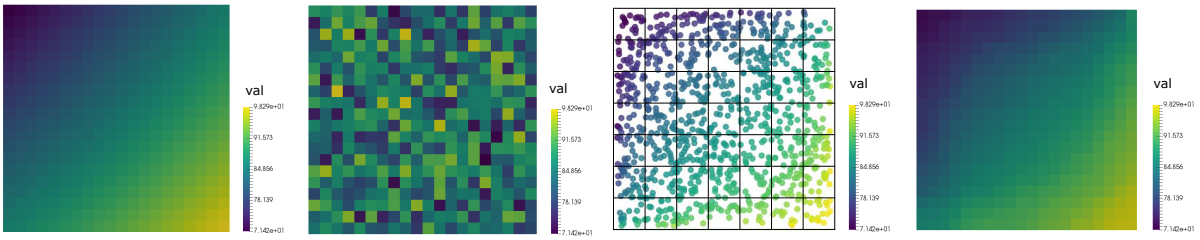


Figure 4.1: A schematic overview of the stages of our proposed method.

simulation domain and then modeling the individual variables in each partition using suitable univariate distribution models (mainly Histogram, Gaussian or GMMs are used in our work). The dependency among the variables is modeled separately using copula functions (Gaussian copula). The dependency parameters and the respective univariate distributions, computed *in situ*, together comprises our proposed multivariate data summary, which gets written-out to the secondary storage instead of the raw simulation data. In the latter stage, copula-based sampling strategies are used to facilitate various post-hoc multivariate analysis and visualization tasks using the stored data summaries.

4.3 Method

In this section, we elaborate in detail the main activities of our proposed framework. We first explain the significance of storing the spatial information in our multivariate data summaries and then discuss the various post-hoc analysis procedures that can be performed using the proposed data representation.



(a) (b) (c) (d)

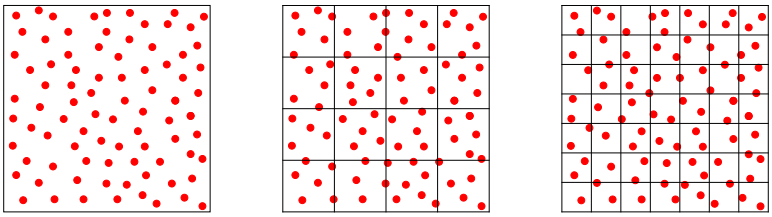
Figure 4.2: Advantage of spatial distributions: (a) Original scalar field of resolution 20×20. (b) Scalar field resampled from a histogram, without any spatial information. (c) Samples generated by the copula-based strategy with spatial distributions. (d) Density field constructed from the copula-based samples in (c).

4.3.1 Advantage of Spatial Distributions

By storing only the value distributions of the physical variables in the simulation, we cannot retain the spatial context in the data. Spatial information is a vital property of scientific datasets and many analysis and visualization tasks require spatial queries and context of the data. Therefore, in our work, besides considering the physical variables, we also consider the spatial variables (i.e., x, y and z - dimensions) as part of our multivariate system. In other words, the effective number of variables in our system is $n = n_p + n_s$, where n_p is the number of physical variables computed in the simulation and n_s is the number of spatial variables (3 for a three-dimensional spatial model). We store the spatial variables in the form of *spatial distributions*. A benefit of using our copula-based flexible framework for storing the spatial distributions is that, for a regular partitioning, which is a popular partitioning scheme, we can use uniform distributions to model the spatial variables. Since copula functions have uniform marginals implicitly, we do not have to effectively store any

extra information for the spatial distributions apart from their correlation coefficients with all the other variables.

Therefore, the multivariate samples generated from our proposed data summaries using the sampling startegy discussed in Chapter 3, can be denoted as $(v_1, ..., v_{n_p}, x, y, z)$, where, v_i's are the sample values for the n_p physical variables and (x, y, z), the corresponding sample location in the spatial domain. The spatial information associated with every sample not only facilitates post-hoc analysis but also strengthens dependency modeling accuracy of the copula functions. Figure 4.2 shows the results of a simple experiment to highlight the advantage of storing spatial distributions along with the value distributions of the physical variables. Consider, a small two-dimensional scalar field of resolution 20×20, with values linearly increasing along the diagonal from the top-left to the bottom-right corner of the field, as shown in Figure 4.2(a). Let, H_V be the histogram of the scalar value (say variable V). By sampling H_V, we get possible values of V, but without any spatial context. Therefore, if we visualize the generated random samples we get a noisy scalar field with similar value distribution, but inaccurate spatial information as shown in Figure 4.2(b). On the other hand, if we consider this as a three-dimensional multivariate system with variables V, X and Y, where X and Y are the spatial variables in the field, we are able to retain the spatial information in our generated samples (Figure 4.2c). Figure 4.2(d) shows the density field for the generated particle samples, where we are able to generate more accurate statistical realizations of the initial field. Moreover, since it is a regular Cartesian grid we can use uniform distribution to model X and Y.



(a) samples (b) 5×5 (c) 8×8

Figure 4.3: Arbitrary grid resolutions for the sample scalar fields.

4.3.2 Post-hoc Multivariate Analysis and Visualization

Using our proposed multivariate data summaries, we facilitate two broad categories of analysis tasks, i.e, *multivariate sampling-based visualization* and *multivariate query-driven analysis*. Both these tasks rely on the copula-based sampling strategy discussed in Chapter 3 (Section 3.5).

4.3.3 Multivariate Sampling-based Visualization

Visualizing the scalar fields of the individual variables in the form of volumes or surfaces is a common practice among scientists while dealing with multivariate data. In order to facilitate such visualizations using our proposed multivariate data summaries, we generate statistical realizations/samples from our data representation to create multivariate scalar fields that can be visualized as a replacement of the raw data. We generate multivariate samples for each partition in the spatial domain using our copula-based sampling strategy. Since the generated multivariate samples contain spatial locations, we can create the sample scalar fields by performing particle density estimation at the grid points [135]. For each multivariate sample, we assigned the distance-weighted average of the physical variables to the nearest grid point. The generated sample scalar fields can be in any arbitrary user-specified

grid resolutions as illustrated in Figure 4.3. As a result, depending on the computational resources available on the analysis machine, users can specify a high or a low-resolution sample grid to visualize.

Algorithm 1 Generating a sample scalar field

1: $\mathscr{D} \leftarrow [D_1, ...D_p]$ ▷ list of distributions for p partitions
2: $S_j[T_x, T_y, T_z] \leftarrow \mathbf{0}$ ▷ sample scalar field of size (T_x, T_y, Tz)
3: $sumOfWeights[T_x, T_y, T_z] \leftarrow \mathbf{0}$
4: **for all** D_i in $\mathscr{D}$ **do**
5: $\quad \mathscr{S}[N] \leftarrow generateMVsamples(D_i, N)$ ▷ sample size N
6: $\quad$ **for all s** in $\mathscr{S}[.]$ **do** ▷ $\mathbf{s} \sim (s_1, .., s_n, s_x, s_y, s_z)$
7: $\quad\quad (g_x, g_y, g_z) \leftarrow nearestGridLocation(s_x, s_y, s_z)$
8: $\quad\quad dis \leftarrow distance(\{g_x, g_y, g_z\}, \{s_x, s_y, s_z\})$
9: $\quad\quad weight \leftarrow 1/dis$
10: $\quad\quad S_j[g_x, g_y, g_z] \mathrel{+}= (s_j * weight)$
11: $\quad\quad sumOfWeights[g_x, g_y, g_z] \mathrel{+}= weight$
12: $S_j[.] \mathrel{/}= sumOfWeights[.]$ ▷ the final sample scalar field

The pseudo-code in Algorithm 1 shows the steps involved in generating a sample scalar field. We create a sample scalar field S_j of user-specified target resolution (T_x, T_y, T_z) for the j^{th} variable in a system with n variables. For each multivariate data summary D_i (corresponding to each partition), we generate N multivariate samples using our copula-based sampling strategy as explain above, via the function $generateMVsamples(.)$ in line 5 of Algorithm 1. We then compute the distance-weighted average of the sample values of the physical variables (here s_j) to eventually create the final statistical realization of the scalar field, i.e, S_j. The number of samples generated, N, depends on the size of each partition and is generally kept higher than the number of grid points in the partition to get reliable results.

Using a simple two-dimensional real-world multivariate data, we demonstrate the effectiveness of our proposed method. We consider 2D slices (resolution 250×250) of Pressure

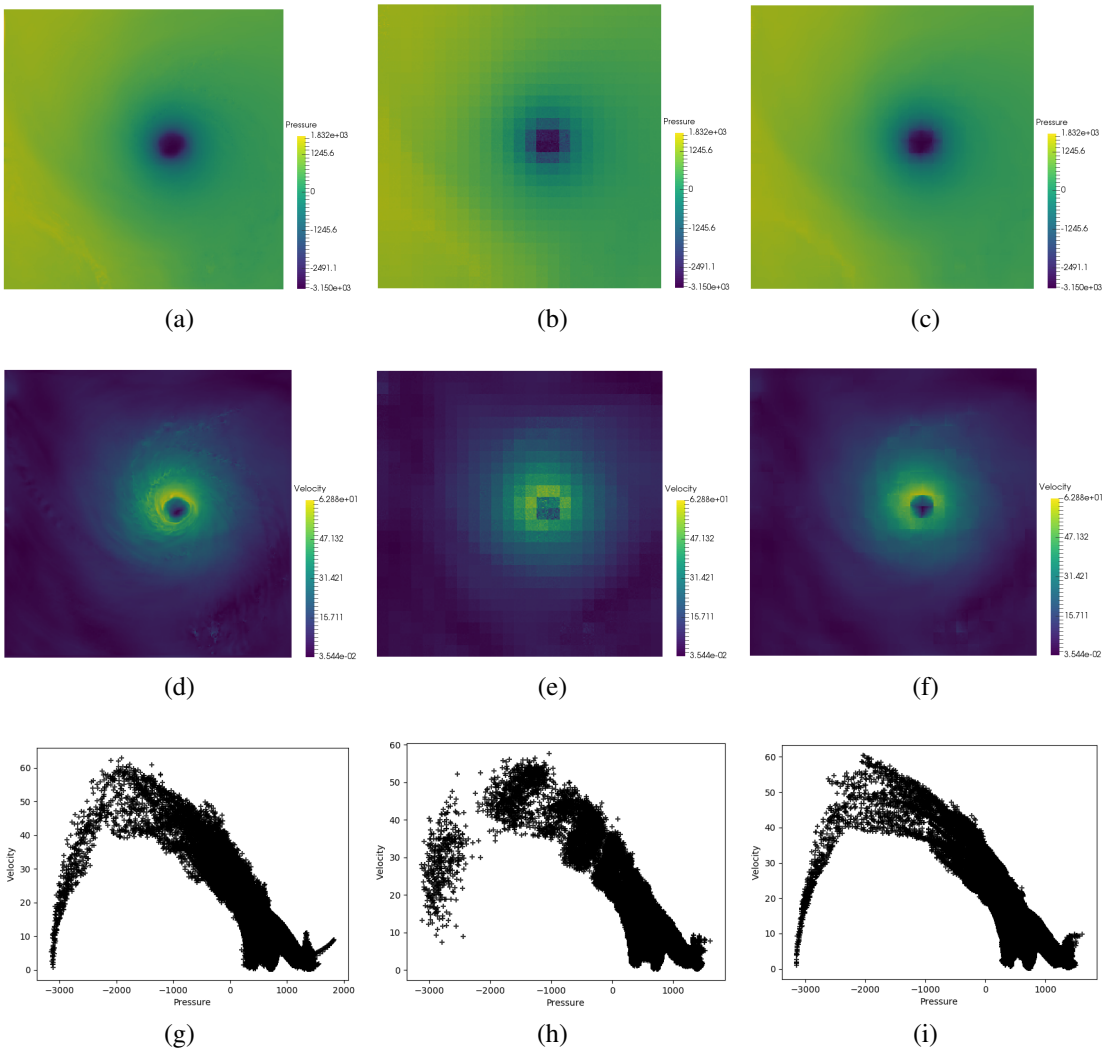
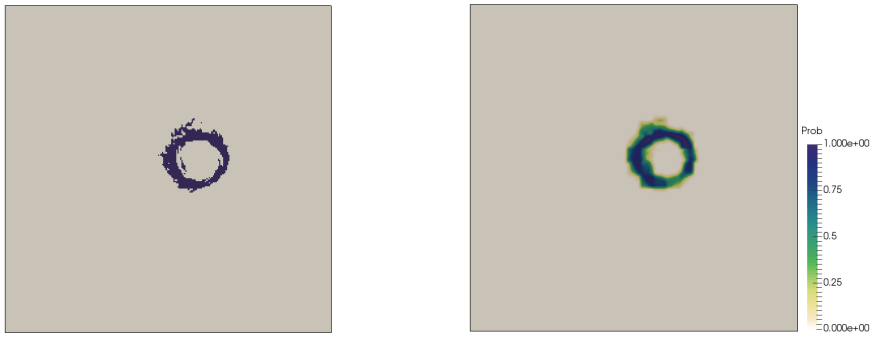


Figure 4.4: Two dimensional slices (250×250) of Isabel dataset, partitioned into 10×10 blocks: (a) Original Pressure field. (b) Pressure field sampled from multivariate histogram. (c) Pressure field sampled using copula-based strategy. (d) Original Velocity field. (e) Velocity field sampled from multivariate histogram. (f) Velocity field sampled using copula-based strategy. (g) Scatter-plot view of original field. (h) Scatter-plot view of the fields sampled from multivariate histogram. (i) Scatter-plot view of the field sampled using copula-based strategy.

and Velocity variables from the Hurricane Isabel dataset. The full volumetric datasets with 11 physical variables will be used later for extensive evaluation in the subsequent

Sections. The original Pressure and Velocity scalar fields are shown in Figure 4.4(a) and (d) respectively, while Figure 4.4(g) shows the scatter-plot view of how the two variables are related. As can be seen, there is a non-linear relationship between the two variables. However, partitioning the spatial domain into smaller blocks help break down the complex global multivariate relationship into relatively simpler local relationships [114, 159], which can be accurately modeled by the Gaussian copula. In this example, we partition the spatial domain into regular blocks of size 10×10. To compare our copula-based strategy with a standard multivariate distribution based strategy, we compute multivariate histograms for the two variables Pressure and Velocity across all the partitions. Using our proposed framework, we only compute the univariate distributions of Pressure, Velocity and the two spatial dimensions X and Y. We use univariate histograms for Pressure and Velocity (with similar bin counts as the multivariate histogram, i.e., 64), while uniform distributions for X and Y. Also, we store the 6, i.e., $\binom{4}{2}$ correlation coefficients to capture the correlation matrix (parameter for Gaussian copula function). Figure 4.4(b) and (e) show the results of the sample scalar fields generated with the multivariate histograms, while Figure 4.4(c) and (f) show the results from our copula-based sampling. The sample scalar fields are in the same resolution as the initial raw slices (250×250). Figure 4.4(h) and (i) show the corresponding scatter-plot views for the two cases. As can be seen, the copula-based sample scalar fields are able to closely resemble the complex multivariate relationship between Pressure and Velocity compared to just using a standard multivariate histogram. Therefore, the flexibility of adding the spatial information as extra variables in our multivariate model helps us to not only create a more accurate scalar field for the individual variables but also reliably capture their multivariate relationships.



(a) (b)

Figure 4.5: Multivariate Query-Driven Analysis: (a) The deterministic results of the query $-2000 < Pressure < 500$ and $40 < Velocity < 50$ in the original raw data. (b) Probabilistic result generated by our methods, i.e., $P(-2000 < Pressure < 500$ AND $40 < Velocity < 50)$.

4.3.4 Multivariate Query-Driven Analysis

Query-driven analysis methods are a class of highly effective discovery visualization strategies [156]. They reduce the computational workload and the cognitive stress in large-scale scientific data by selecting regions of interest and filtering out the other non-pertinent regions. By focusing analysis and visualization efforts only on the regions of interest, such query-driven techniques make the work-flow of scientists more manageable and effective. For example, if scientists are interested in only a certain value range for two variables, a query-driven method helps them to focus only on the parts of the data that specifically meet their multivariate query, instead of looking at the entire simulation domain. They can further drill down into analyzing how the other variables behave in the region of interest to gain more insights. Many query-driven strategies rely on computing local data statistics to perform efficient query search operations [29, 67]. Therefore, the use of statistical data summaries is a wise choice for data reduction in large-scale simulations because it can easily facilitate

such query-driven strategies. In this section, we explain in detail the process of performing multivariate query-driven analysis using our proposed multivariate data summaries.

To illustrate our copula-based multivariate query-driven analysis, consider the same 2D slices of the Isabel data. Consider performing a query on the Pressure range of $[-2000Pa - 500Pa]$ and Velocity range of $[40ms^{-1} - 50ms^{-1}]$. To compute the probability of seeing a multivariate value in this queried range, we selectively sample the stored multivariate distributions using our copula-based sampling method. To expedite the process, for each partition, we first check whether the corresponding univariate distributions of the queried variables satisfy the individual query ranges or not. We generate multivariate samples using our copula-based strategy only for the partitions which satisfy this initial check. As mentioned in the previous section, the multivariate samples generated in our method retains the spatial context in the form of spatial locations for each sample. By creating a spatial density field of the generated samples satisfying the query, we can produce the *probabilistic multivariate query field*, which highlights the probability of the specified multivariate query (i.e., $P(-2000 < Pressure < 500 \text{ AND } 40 < Velocity < 50)$). Figure 4.5(a) shows the region which satisfies the query in the original raw data. Figure 4.5(b) shows the corresponding probability density field for the query with probability values ranging from 0 to 1. A high value indicates a high possibility of seeing co-occurring Pressure and Velocity values in the specified ranges.

4.4 Quantitative and Visual Evaluation

To demonstrate the effectiveness of our proposed multivariate data summarization strategy, we first evaluated it on two off-line multivariate data before applying it on a full-scale *in situ* simulation. We used the following off-line datasets: (a) Hurricane Isabel WRF

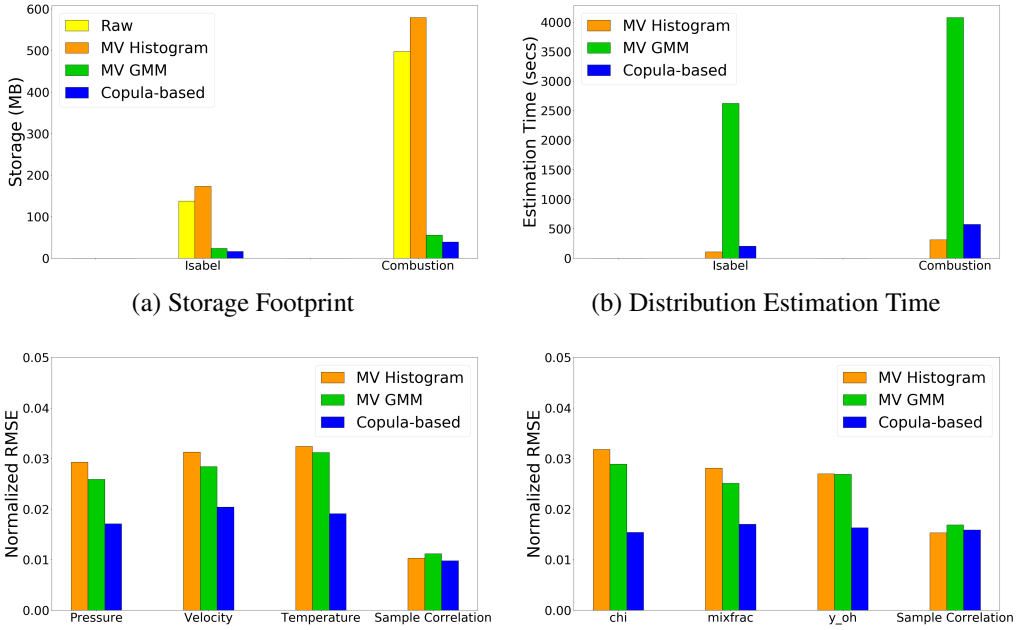
Table 4.1: Distribution Storage and Estimation Time

Dataset (Resolution)	#variables	Raw Size (MB)	block size	MV Histogram		MV GMM		Hybrid + Copula	
				Size (MB)	Est. Time (s)	Size (MB)	Est. Time (s)	Size (MB)	Est. Time (s)
Isabel (250x250x50)	11	137.5	5x5x5	173.1	106.1	23.7	2623.6	16.2	203.9
			7x7x7	152.5	111.5	8.13	4671.6	5.8	205.4
			10x10x10	113.7	98.2	2.95	5006.2	2.2	230.2
Combustion (480x720x120)	3	497.7	5x5x5	579.4	311.7	55.7	4077.7	39.2	573.3
			7x7x7	509.1	322.4	39.7	5150.4	14.3	561.7
			10x10x10	434.2	305.7	27.8	9708.5	5.1	583.6

model data of resolution $250 \times 250 \times 50$, with 11 physical variables, which models the development of a strong hurricane in the West Atlantic region, and (b) Combustion data of resolution $480 \times 720 \times 120$, with 3 physical variables, modeling a turbulent combustion process. For the purpose of our evaluation, we considered a single time step of the above datasets (time step 20 for Isabel and time step 30 for Combustion). All evaluations were performed on a standard workstation PC (Intel i7 at 3.40GHz and 16GB RAM).

4.4.1 Experiment Setup

In our experiment, we used non-overlapping regular partitioning scheme of equal block sizes to partition the simulation domain. Multivariate data summaries were then created for individual partitions. We tested our proposed summarization model against standard multivariate distribution models like multivariate histogram (sparse representation with 32 bins of equal width for all the dimensions) and multivariate GMM of 3 modes (with full covariance matrix). In our proposed flexible framework, to model the individual variables, we used a hybrid combination of univariate distributions involving *GMMs*, *Gaussian distributions* and *uniform distributions*, while, *Gaussian copula* was used to model the dependency among these hybrid distributions. For each partition, we performed a normality test (D'Agostino's



(a) Storage Footprint

(b) Distribution Estimation Time

(c) RMSE for Isabel

(d) RMSE for Combustion

Figure 4.6: Quantitative evaluation results for block size of 5^3.

K-squared test [40]) on the individual variables. For variables with a high certainty of following a normal distribution, we used a Gaussian distribution, else GMM of 3 modes was used, whereas, uniform distributions were used to model the spatial variables (i.e., x, y and z dimensions) for each partition. Therefore, the effective number of variables in our method for Isabel dataset is *14 (11 physical + 3 spatial)* and for the Combustion dataset is *6 (3 physical + 3 spatial)*.

4.4.2 Storage Footprint

The storage size of our proposed multivariate data summaries was significantly less as compared to the standard multivariate distributions, even when including the 3 spatial variables and extra indexing information for recording the hybrid univariate distribution

types at each partition. Figure 4.6(a) compares the storage sizes for the three different models in the Isabel and Combustion datasets for block sizes of 5^3. Clearly, multivariate histogram is not a good alternative for the purpose of data-reduction. Also, the fact that in our hybrid model, we selectively used GMMs of 3 modes and single Gaussian distributions, helps us achieve better storage size than the standard multivariate GMM (of 3 modes).

4.4.3 Estimation Time

We compared the estimation times of the three data summarization models for the two datasets. As shown in Figure 4.6(b), the distribution estimation time for multivariate GMM is significantly high compared to the other models. As a result, despite having good storage advantages, multivariate GMMs will greatly increase the simulation time when used in *in situ* applications. On the other hand, estimating multiple univariate distributions is comparatively less expensive, because of which our proposed multivariate data modeling strategy performed significantly better. The estimation time of our model included the time for normality test, the individual univariate distribution estimation and the Gaussian copula parameter computation time. Table 4.1 reports the storage sizes and estimation times for different block sizes.

4.4.4 Accuracy

Using the three data summarization models, we created sample scalar fields of resolutions similar to the original raw data. In the case of multivariate histogram and multivariate GMMs, for each grid location in the reconstructed field, we draw random samples from the distribution corresponding to the block (partition) that the grid location belongs to. The value of this sample is assigned to the specific grid location. This approach is similar to the

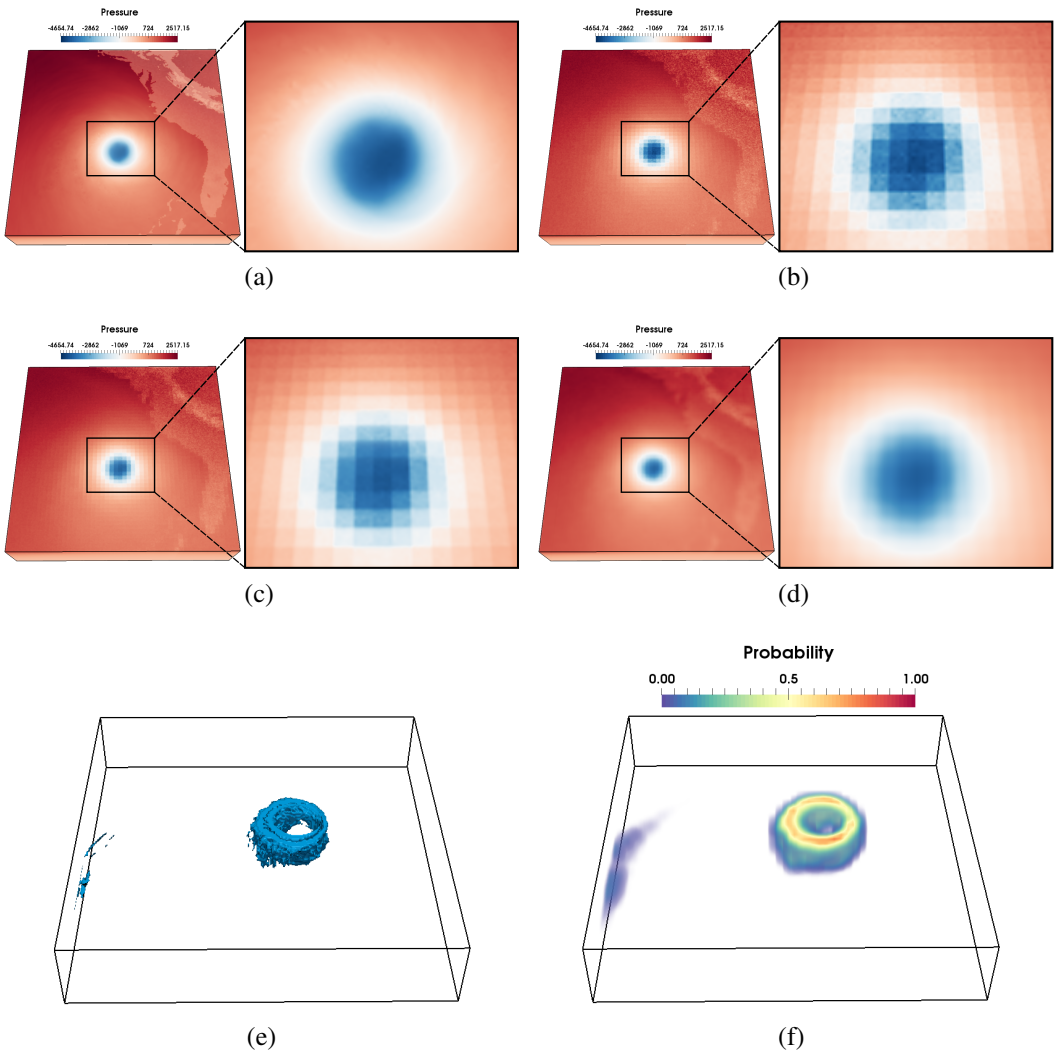
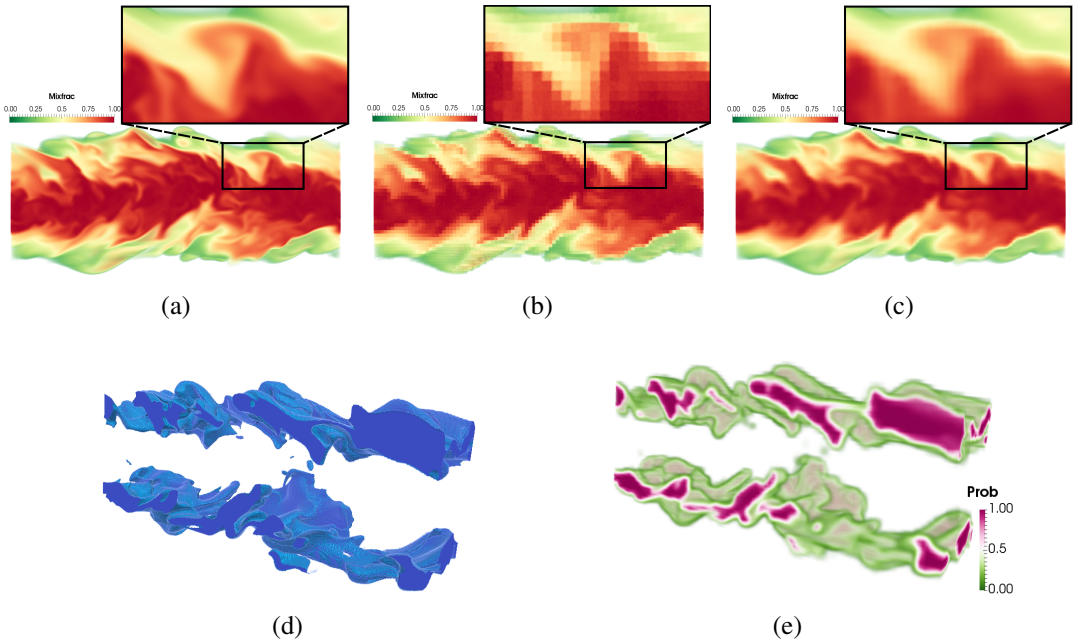


Figure 4.7: Results from Isabel dataset for block size 5^3: (a) Original Pressure scalar field. (b) Pressure field constructed from multivariate histograms representation. (c) Pressure field constructed from multivariate GMM of 3 modes. (d) Pressure field created by our copula-based model, which retains the spatial context in the multivariate samples. (e) Region in the original raw data corresponding to the multivariate query of $-2000 < Pressure < 500$ and $40 < Velocity < 50$. (f) The probability field generated by our copula-based strategy for the similar query, i.e., $P(-2000 < Pressure < 500 \text{ AND } 40 < Velocity < 50)$.

reconstruction strategies employed in other univariate distribution-based data summarizations works [47, 181]. On the other hand, we employed our copula-based strategy to generate

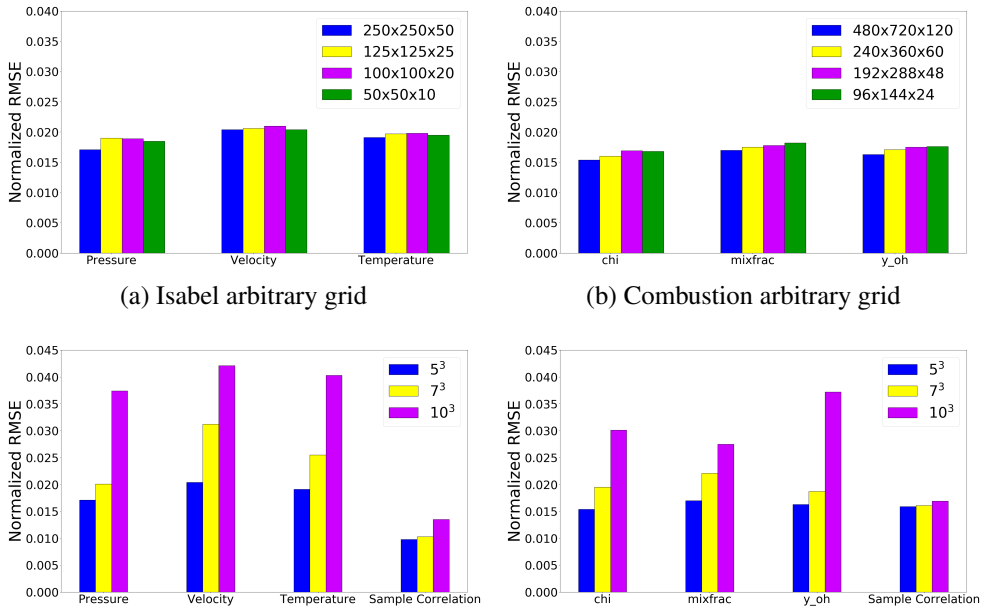


(a) (b) (c)

(d) (e)

Figure 4.8: Results from Combustion dataset for block size 5^3: (a) Original mixfrac scalar field. (b) Mixfrac field constructed from multivariate GMM of 3 modes. (c) Mixfrac field created by our copula-based model. (e) Region in the original raw data corresponding to the multivariate query of $0.3 < Mixfrac < 0.7$ and $y\_oh > 0.0006$. (f) The probability field generated by our copula-based strategy for the similar query, i.e., $P(0.3 < Mixfrac < 0.7 \text{ AND } y\_oh > 0.0006)$.

the sample scalar fields using our proposed data summaries. To compare the accuracies of the sample scalar fields, we computed their normalized root mean squared error (RMSE) with the corresponding original raw fields. Figure 4.6(c) and (d) show the RMSE results for three variables in both the datasets. The results of all the 11 variables for Isabel is provided in the supplementary material. To evaluate the multivariate relationship preserved by the models, we computed the RMSE values of the sample correlation coefficients of all the pairs of variables with the original correlation coefficients across all the partitions. As shown in the last stack of bar-charts in Figure 4.6(c) and (d), the sample correlation errors from the three different models are mostly similar, this is because, the use of copula



(a) Isabel arbitrary grid

(b) Combustion arbitrary grid

(c) Isabel block sizes

(d) Combustion block sizes

Figure 4.9: (a) and (b) show the consistent RMSE values for different grid resolutions of the sample scalar field, when block size is 5^3. (c) and (d) show the trend of increasing RMSE values with increasing block-sizes.

is just another way of modeling multivariate distributions. Figure 4.7(a-d) show the visual comparison of the sample scalar field generated for the Pressure variable in Isabel dataset, while Figure 4.8(a-c) show the results for the Mixfrac variable in Combustion dataset (more results are provided in the supplementary material). The accuracy of scalar fields generated by our copula-based sampling strategy is better than the standard models because we were able to retain the spatial information in the form of spatial distributions. Therefore, based on the above three criteria, i.e., *storage footprint*, *estimation time* and *accuracy*, we can say that our proposed flexible multivariate data summary framework is better suited for the analysis of large-scale multivariate data than the corresponding standard multivariate distributions.

4.4.5 Multivariate Query

To facilitate query-driven analysis tasks, we computed the probability field for a given multivariate query using our hybrid model. Figure 4.7(e) shows the deterministic query result on the original raw data for the multivariate query $-2000 < Pressure < 500$ and $40 < Velocity < 50$ for the Isabel dataset. Figure 4.7(f) shows the corresponding probability field generated for the same query using our multivariate data summaries (i.e., $P(-2000 < Pressure < 500 \text{ AND } 40 < Velocity < 50)$). As a result of the spatial information preserved in our model, we were able to successfully identify the region of interest for the specific query along with uncertainty information, provided in the form of the probability values. The regions with high probability value have higher chances of satisfying the given query. Based on the query results, scientists can further analyze the properties of other variables in this spatial range (more results are provided in the supplementary material). Similarly, Figure 4.8(d) shows the deterministic query results on the original Combustion raw data for the query $0.3 < mixfrac < 0.7$ and $y\_oh > 0.0006$, while, Figure 4.8(e) shows the corresponding probabilistic query ($P(0.3 < mixfrac < 0.7 \text{ AND } y\_oh > 0.0006)$).

4.4.6 Arbitrary Grid Resolution

The sample scalar fields generated by our method can be created in arbitrary user-specified grid resolutions because of the spatial information retained in the multivariate samples. As a result, users have the flexibility to create a high or a low-resolution sample field directly from the summaries depending on the computational resources available at their disposal for analysis. To test the results of the arbitrary grid resolutions, we computed the RMSE scores of the generated sample scalar fields with that of the corresponding scalar fields sub-sampled from the original raw field. Figure 4.9(a) shows the normalized RMSE

scores for three variables in the Isabel dataset. For a single variable, each bar corresponds to the RMSE score of the corresponding grid resolution. The sub-sampled scalar field generated from the original raw data is considered as the baseline for each resolution size. Similarly, Figure 4.9(b) shows the results for Combustion dataset. The RMSE scores remain consistent across different grid resolutions for the individual variables.

4.4.7 Effect of block sizes

We also studied the effect of partition block sizes (i.e, granularity of domain partitioning) on the overall storage size, estimation time and RMSE values. With larger block sizes, the overall storage footprint decreases but the overall estimation time increases. This increase of estimation time is more significant with multivariate GMMs. Table 4.1 shows the storage and estimation times for different block sizes for the two test datasets. Also, with larger block sizes the overall RMSE values for the analysis results increases. Figure 4.9(c) and (d) show the increasing trend of RMSE values for some of the individual variables and the sample correlation coefficients in Isabel and Combustion datasets respectively. The number of multivariate samples generated from each multivariate data summary also depends on the partition block size. To get statistically reliable results the number of samples is generally larger than the number of grid points in each partition. We tested with different sample sizes and observed that with increasing sample sizes the overall accuracy does not differ significantly after a certain size. For our case, we used sample sizes of 500, 1000 and 1500 for block sizes of 5^3, 7^3 and 10^3 respectively.

4.5 *In Situ* Application

Based on the positive evaluation results in off-line multivariate data, next, we applied our proposed flexible multivariate data summarization framework on a real-world *in situ*

environment. Using our proposed model, we want to facilitate flexible and scalable multivariate analysis of data generated in a large-scale computational fluid dynamics (CFD) simulation code, TURBO [31, 32]. TURBO, developed at NASA, is a Navier-Stokes based, time-accurate CFD simulation code to study transonic jet engine compressors at high resolutions. Domain experts compute various physical variables to study and analyze the inception of flow instability across the compressor blades. Flow instability can lead to potential stalls in the engine, which can damage the blades. Therefore, it is important to understand and analyze what roles the different variables play in the creation of such unstable flow structures. However, the computational cost and the amount of data produced from a single simulation is quite significant, which makes such multivariate analysis very unwieldy and overwhelming for the scientists.

For this case, scientists were interested in analyzing the multivariate relationship among the variables Entropy, Uvelocity and Temperature. We computed our proposed multivariate data summaries for partitions of size 5^3 across the simulation domain. Based on the results of normality test, we used either a Gaussian distribution or a GMM (with 3 modes) to model the univariate distribution of individual variables. The spatial variables were modeled using uniform distributions, while Gaussian copula captured the dependency structure among all these variables (i.e., 6, 3 physical + 3 spatial). The *in situ* simulation was performed in a cluster (Oakley [6], at the Ohio Supercomputer Center) containing 694 nodes with Intel Xeon x5650 CPUs (12 cores per node), and 48 GB of memory per node. The simulation was run on 328 cores in total. We executed 2 full revolutions of the jet turbine, resulting in 7200 time steps. *In situ* multivariate data summarization was performed every 10^{th} time step, thereby storing 720 time steps. We created our hybrid multivariate data summaries by accessing the simulation memory directly without additional data copies. The domain of the

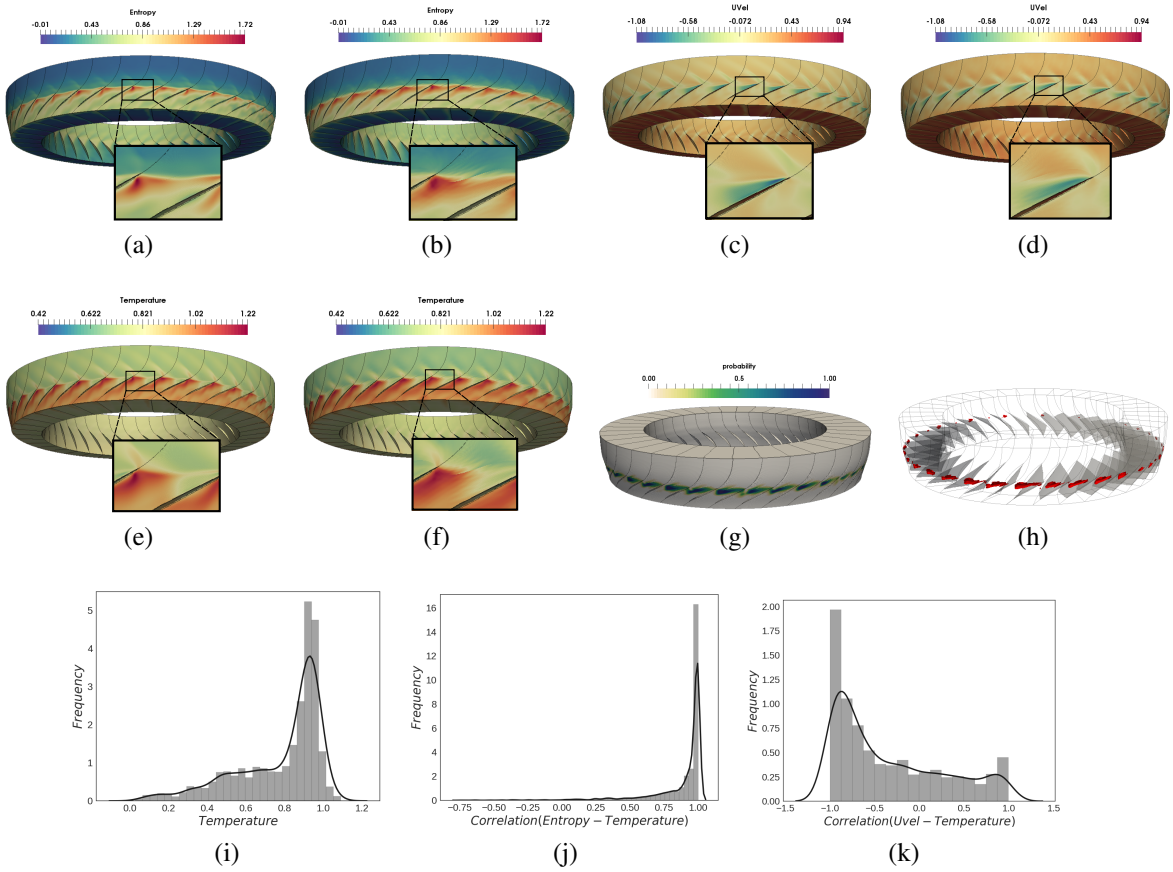


Figure 4.10: Post-hoc analysis of the jet turbine dataset. (a) Original Entropy field. (b) Sample scalar field of Entropy. (c) Original Uvelocity field. (d) Sample scalar field of Uvelocity. (e) Original Temperature field. (f) Sample scalar field of Temperature. (g) Probabilistic multivariate query result i.e., $P(Entropy > 0.8 \text{ AND } Uvel < -0.05)$ (h) Isosurface for probability value 0.5. (i) Distribution of Temperature values in the queried region i.e., $P(Temp|Entropy > 0.8 \text{ AND } Uvel < -0.05)$. (j) Distribution of correlation coefficients between Entropy and Temperature for the queried region. (k) Distribution of correlation coefficients between Uvelocity and Temperature for the queried region.

compressor consists of 36 blade passages, each with a spatial resolution of $151 \times 71 \times 56$. The simulation outputs raw data in multi-block PLOT3d format of size 690 MB per time step, which accounts for **496.8** GB for just two 2 revolutions. On the other hand, our proposed multivariate data summaries result in only **19.6** GB of total storage footprint.

Table 4.2: *In situ* Performance

Simulation Time (hrs)	Raw I/O Time (hrs)	In situ Data Summarization (hrs)	Data Summaries I/O Time (hrs)
13.5	1.76	2.09	0.0063

Table 4.3: Post-hoc Analysis Performance

MV Query per time step(secs)	Sample Scalar Field per time step (secs)	Normalized RMSE		
		Entropy	U-Vel	Temp
64.6	178.3	0.0211	0.0174	0.0184

Table 4.2 shows the overall simulation times for our *in situ* application. Our multivariate data summary creation process requires about 15.4% of the original simulation time but offers the flexibility of scalable post-hoc analysis as compared to storing the raw data (the raw data I/O time itself takes 13% of the simulation time).

Multivariate data summaries were later used to generate sample scalar fields for the variables of interest, as well as perform multivariate query-driven analysis. Figure 4.10(a,c,e) show the original scalar fields for Entropy, Uvelocity and Temperature respectively, whereas Figure 4.10(b,d,f) shows the corresponding sample scalar fields for the respective variables generated by our copula-based sampling strategy. Scientists were interested to see how the selected variables affect flow instability in the turbine. Prior studies on univariate data [30, 45, 47] highlights that Entropy values great than 0.8 and negative Uvelocities correspond to potentially unstable flow structures. Therefore, we computed the multivariate query, $Entropy > 0.8$ and $Uvel < -0.05$ from our stored data summaries. The corresponding probability field is shown in Figure 4.10(g), whereas, Figure 4.10(h) shows the isosurfaces of probability value 0.5 across the blade structures. Figure 4.10(i) shows the distribution of Temperature values in this queried region (i.e., $P(Temp|Entropy > 0.8$ AND $Uvel <$

-0.05)). The peak in the distribution suggests that Temperature values around 0.9 can be related to potential flow instability. Figure 4.10(j) and (k) show how Temperature is correlated with Entropy and Uvelocity respectively, in the selected queried range. There is a strong positive correlation with Entropy and a substantial amount of negative correlation with Uvelocity. Such exploratory analysis activity can help the scientists to gain more insights into the multivariate relationships in their simulation. All post-hoc analysis were performed on a standard workstation PC (Intel i7 at 3.40GHz and 16GB RAM) with 8 CPU cores. Using OpenMP parallelization, we ran the analysis tasks on all the CPU cores. Table 4.3 shows the average post-hoc analysis time and accuracy results for a single time step.

4.5.1 Domain Expert Feedback

We presented the results and explained the idea behind of our proposed framework to the domain scientist. The expert agrees with the fact that having a summarized version of the original multivariate data is useful, as it facilitates effective post-hoc multivariate analysis. Previous analysis works on this simulation were primarily centered around studying the effect of the variables independently [30,45,47], but our expert feels that this framework will be useful to study how the interaction among different variables influence flow instability in the engine. The result of our multivariate query aligns with the expert's knowledge that the potential unstable regions generate near the edges of the blades, as shown in Figure 4.10(g,h). The expert feels that the distribution of Temperature and correlation strengths in this queried region is similar to what is originally expected. Generally, because of the large storage requirements, the raw simulation data was stored only after around 25-30 time steps. But, with our proposed data summaries, we can now store at finer temporal resolutions (every

10^{th} time step in this case). Expert feels that this will help analyze the finer temporal events in the simulation. Overall, the expert acknowledges that our proposed framework is an effective strategy to understand the multivariate relationships in his simulation without having to store the large-scale simulation data off-line.

Chapter 5: Uncertainty Visualization Using Copula-based Mixed Distribution Models in Ensemble Datasets

5.1 Introduction

Most of the numerical simulations which are used to model complex real world physical phenomenon generate uncertain data. The lack of a proper ground truth and/or simulation parameter knowledge are some of the common causes of uncertainty. In order to avoid making erroneous decisions using such data, it is important to incorporate the uncertainty into the analysis process itself. For example, tasks like feature extraction and visualization should reflect the effect of uncertainty in the data. With recent advances in computing power and resources, scientists are able to model the uncertainty by running ensemble of simulations with varying experiment parameters, thus, generating multiple realizations of the same physical phenomenon. These multiple realizations/values at each of the spatially sampled points (grid locations) represent the uncertainty in that location and are often modeled as stochastic random variables. Various approaches have been proposed [12, 133, 141] to extract probabilistic/uncertain features from such a field of random variables using standard statistical tools.

An important property to be taken into account while modeling uncertainty in spatially sampled scientific datasets is the correlation among the grid locations due to the inherent local data continuity [137, 138, 142, 144]. Therefore, various multivariate distribution models have been proposed to model the uncertainty in the data which can preserve the dependency/correlation among the random variables at each grid locations. The choice of the statistical model plays an important role in any distribution driven uncertainty analysis. Among the parametric models, the multivariate Gaussian distribution is the most popular choice [143, 144], while, most common nonparametric models are histograms, empirical distributions and kernel density estimates (KDE) [12, 142]. Multivariate Gaussian distributions are useful to model the multivariate dependency but it has the basic assumption that the univariate marginal distributions (at the grid locations) are all Gaussians. This can lead to misleading results if the underlying distribution at a location does not follow a normal distribution. Pöthkow et al. [142] highlighted this problem in parametric models and extended their work to consider nonparametric multivariate models which fits the data better. However, there are two possible challenges with such nonparametric models. First, the estimation and subsequent analysis of multivariate nonparametric distribution models is computationally intensive (both in terms of time and memory footprint). Second, they are susceptible to generate biased results if an over-fitted nonparametric model is chosen for a sample which shows high confidence of following a particular parametric model. A general problem with all standard multivariate models (both parametric and nonparametric) is that they consider all the univariate marginals to follow the same family/class of distribution. But in reality, not all the locations in the data show uncertainty trends which can be best modeled by the same type of distribution. For example, a simple statistical normality test can reveal the fact that not all the grid locations show equal confidence of following a

normal distribution. Some locations show high certainty, whereas, others show very low certainty. Recently, Bensema et al. [15] in their modality driven analysis, have shown that the ensemble distributions at different locations can vary significantly. Therefore, there is a need to adopt a different multivariate strategy to model uncertainty in scientific datasets, which is flexible enough to model the univariate marginals by different types of distributions as well as be able to model the multivariate dependency among the random variables.

In this chapter, we propose a new copula-based uncertainty modeling and analysis technique for scientific datasets which can separate the estimation of multivariate dependency structures from the process of estimating univariate marginal distributions at each grid location. The proposed technique makes it possible to choose the best possible univariate distribution to model the uncertainty at each grid location. The resulting distribution field, which can have different types of distribution (Gaussian, Histogram, KDE, GMM etc.) at different grid locations is henceforth referred to as a *mixed distribution field* in our work. In fact, copula-based techniques can accommodate any univariate distribution type, as long as it is a continuous distribution with a valid cumulative density function (CDF). A major advantage, for example, of using such a flexible strategy is that we can choose the aforementioned computationally intensive nonparametric models only for those grid locations where parametric models fail to model with sufficient confidence. This can significantly reduce the computational cost without compromising the quality of the analysis. To demonstrate the effectiveness of our flexible scheme, we propose copula-based techniques to visualize uncertain/probabilistic features in the resulting mixed distribution fields. We introduce methods to compute the level-crossing probability values to extract probabilistic isocontours in mixed scalar distribution fields as well as vortex-core probabilities to determine uncertain vortex features in mixed vector distribution fields. We compare the results of our probabilistic

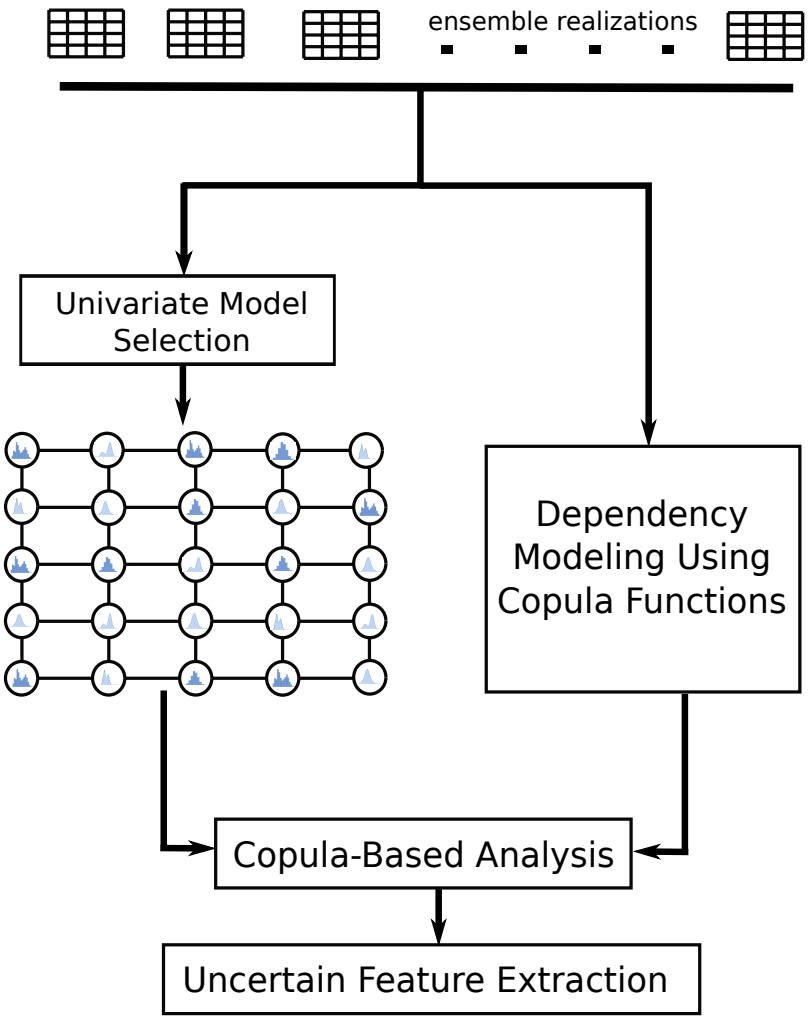


Figure 5.1: A high-level schematic overview of our proposed method.

features against the results generated by existing methods which use standard multivariate models and evaluate them based on correctness and computational complexity. The selection of the optimal univariate model at each grid location is an important task. Since there are multiple statistical tests available at our disposal, we guide the users by identifying the ones that are useful for scientific data modeling and highlight their advantages.

5.2 Motivation and Overview

5.2.1 Motivation

Our proposed technique is specifically tailored to meet the needs of uncertainty modeling in scientific datasets. Some of the key aspects of distribution-based uncertainty modeling in scientific datasets that need to be taken into account are as follows:

1. **Univariate Model Selection:** A single class/type of distribution may not be the best way to model the underlying uncertainty at each spatially sampled grid location because different locations show different statistical properties. Therefore, to identify the optimal distribution model, some form of statistical test must be performed at each grid location.

2. **Dependency Modeling:** Since scientific datasets have the inherent property of local data continuity, there exists a spatial correlation among the values at the grid locations. Therefore, any uncertainty modeling technique must consider the spatial correlation/dependency among the univariate distributions at each grid location.

3. **Computational Complexity:** The size of the distribution field (indicated by the number of univariate distribution models) is dictated by the resolution of the dataset. Performing tasks like feature extraction and/or query on large distribution fields can become computationally expensive (in terms of time and memory footprint) if complex models are used. There needs to be a balance between the degree of correctness and the overall computational expense of generating the results.

Standard multivariate distribution models are not flexible enough to meet all of these requirements at the same time because the multivariate dependency structure is strongly

coupled with the type of univariate marginal distribution. Our proposed copula based technique provides a framework to separate the two tasks i.e, univariate model selection and dependency modeling. As a result, users have the flexibility to choose an optimal distribution at each location to model uncertainty and still preserve the correlation among the locations. This independent execution of the two tasks and the fact that we can now use the computationally expensive models only when it is absolutely necessary, in turn, helps us to achieve faster analysis time while generating similar, if not more statistically reliable results.

5.2.2 System Overview

A high-level overview of our proposed idea is shown in Figure 5.1. Ensemble realizations are used to select an optimal distribution model at each grid location by performing standard statistical tests. The resulting *mixed distribution field* can have different types of distribution models at each grid location as illustrated in Figure 5.1. On the other hand, the spatial correlation among the neighboring grid locations is modeled from the ensemble realizations separately using Copula functions (Gaussian copula). Finally, using a sequence of quantile transformations we model the desired multivariate distribution, comprising of mixed marginal distributions and the preserved dependency structure. We use this technique to extract uncertain features like isocontours and vortices in various mixed distribution fields. Our proposed method uses a Monte-Carlo based integration technique to model the multivariate distribution. We compare our results with other Monte-Carlo based techniques that use multivariate Gaussian models [144] and other nonparametric models (histograms, KDE) [142].

5.3 Univariate Model Selection

The task of selecting a proper model to represent the uncertainty at each grid location is vital for subsequent uncertainty analysis. The choice of model often varies on the types of data and the eventual goal that needs to be achieved via modeling. Statistical models are broadly classified into two categories, *parametric* and *nonparametric*. Parametric models are based on assumptions about the distribution of the underlying population from which the sample was taken. Nonparametric models do not rely on any assumptions about the shape or parameters of the underlying population distribution. Depending on factors like initial sample size, type of post-hoc analysis to be performed and computational complexity both forms of models have distinct advantages and disadvantages. Therefore, it is highly advisable to judge the pros and cons before deciding to select specific models.

A popular statistical maxim is that nonparametric models generally have less *statistical power*[1] for the same sample size than the corresponding parametric model which shows high certainty [51]. Therefore, any statistical test showing high certainty for a parametric model should be preferred over nonparametric models. However, if an appropriate parametric model cannot be ascertained with sufficient confidence, often a nonparametric form of model is recommended. For small sample sizes, statistical tests for parametric models often fail, in which case, nonparametric models are the only option. But one must be aware of the potential side-effects of an over-fitted nonparametric model. When the models are used for some Monte-Carlo sampling based post-hoc analysis, the risk of over-fitted nonparametric models generating biased results increases. Cost of working with the eventually selected model is also another important factor. While parametric forms have a fixed (usually low) number of parameters, the nonparamteric models, despite the name, have to store parameters

[1] *Statistical power* of any test is defined as the probability that it will reject a false null hypothesis.

in proportion with sample data size. As a result, working with the nonparametric models may require more computational time compared to the parametric models. For large sample sizes, working with nonparametric models become computationally very expensive (both in terms of time and memory).

Many standard statistical tests currently exist to decide which model can best represent a given sample. However, often a single test cannot inspect all the aforementioned concerns. Therefore, users have to carefully design their tests based on their requirements. Depending on the application and scale of operation, model selection tasks can be as simple as graphical validation of the shapes of distributions to as complex as solving an optimization function with desired requirements as the function variables. In this section, we put forward some of the most commonly used model testing practices prevalent in the field and the ones that are specifically useful for scientific datasets.

5.3.1 Normality Test

Checking for a Gaussian behavior is the most common and useful test that can be made before testing any other distribution because most data simulating a natural phenomenon is Gaussian in nature [62]. However, if the underlying distribution is not normal, making a normality assumption to represent the data can lead to erroneous results. In statistics, normality tests are used to compute how likely it is for a random variable of a dataset to be normally distributed. Studies have shown that for the same sample size Shapiro-Wilk test [165] is the most powerful (i.e, *statistical power*) normality test [150]. The Shapiro-Wilk test returns a likelihood value, commonly referred to as pValue, which lies between 0 and 1. Small pValues lead to the rejection of normality whereas a value of 1 ascertains normality with high confidence. A pValue in the range of $[0.05, 0.1]$ is often considered as a good

threshold value to make a call to decide normality. Data showing pValues less than the threshold normality value can be checked for further parametric distributional properties or select a suitable nonparametric model.

5.3.2 Generalized Goodness-of-fit Test

While the normality test tells us whether a dataset follows a normal distribution or not it does not offer a means to check the sample for multiple distribution types. Kolmogorov-Smirnov goodness-of-fit test (KS test) [167] is a more generic platform for such comparative validation. It compares the CDF of the distribution we want to test for against the empirical CDF (ECDF). Goodness-of-fit is decided by how close the CDF of a distribution is to the ECDF. If $F(x)$ represents the CDF of the hypothesized distribution and $F_e(x)$ represents the ECDF, then the KS test measure is given as,

$$K = sup_x |F(x) - F_e(x)| \tag{5.1}$$

where *sup* stands for supremum, which means the greatest. This is a more generalized statistical test which lets us test for any continuous distribution as along as it has a valid CDF (i.e $F(x)$). Several goodness-of-fit test are in fact refinement of the KS test. One big advantage of the KS test is the ability to compare a parametric model versus a nonparametric model which is not provided by many other complex statistical test.

5.3.3 Bayesian Information Criterion

Bayesian Information Criterion (BIC) [57] is a popular model selection tool for selecting among a finite set of parametric models. It is based on the log likelihood of a given model on the sample data. It is defined as,

$$BIC = -2L_p + p\log(n) \tag{5.2}$$

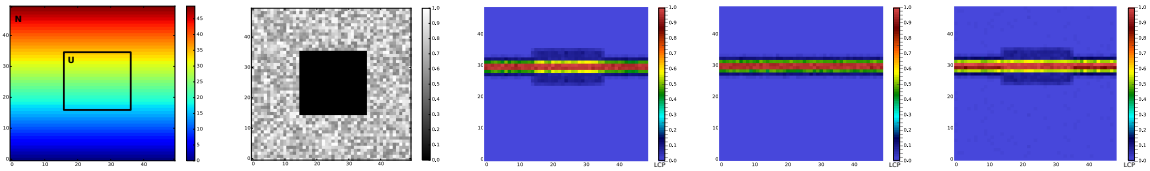
where n is the sample size, L_p is the maximized log-likelihood of the chosen model and p is the number of parameters in the model. A low BIC value indicates a better model. BIC attempts to address the risk of over-fitting by introducing a penalty term $p\log(n)$, which grows with the number of parameters. This eliminates overly complicated models with large number of parameters. BIC severs as a good tool for our model selection task when the desired distributions are all parametric.

5.4 Uncertain Feature Extraction

In this section, we put together the modeling techniques to extract and visualize uncertain features. Our copula-based method allows us to separate the process of estimating the univariate model at each grid location from the dependency modeling of the locations. Using the model selection guidelines proposed in Section 5.3, we create a mixed distribution field by independently modeling the univariate distributions at each grid locations with the desired distribution type. On the other hand, dependency is modeled by computing the correlation matrix of a local neighborhood from the initial sample values at the corresponding grid locations. The spatial correlation among the locations diminishes with Euclidean distance, therefore it is sufficient to consider the correlation within localized spatial regions [137, 138, 142, 144, 160]. Using this proposed strategy, we focus on the extraction of two specific types of features, uncertain isocontours in scalar distribution fields and uncertain vortices in vector distribution fields.

5.4.1 Copula-based Uncertain Isocontour Extraction

Level-crossing probability (LCP) is a popular uncertain isocontour detection measure for distribution based data [137, 141–144]. It involves computing the probability of the level-set/isocontour of a given isovalue passing through the cells of the dataset. Similar to



(a) (b) (c) (d) (e)

Figure 5.2: Synthetic Data: (a) Illustrates the synthetic data creation process. Samples are drawn from a uniform distribution for the locations inside the rectangle and from a normal distribution for outside. (b) shows the result of the Shapiro-Wilk normality test on the initial samples at each grid location. We compute the level-crossing probability (LCP) for isovalue 30 (c) using our proposed method on a mixed distribution field, (d) using the multivariate Gaussian distributions [144] and (e) using multivariate histograms.

the method proposed by Pöthkow et al. [142, 144], we adopt a Monte-Carlo based sampling strategy with the difference that we use a copula-based sampling method to handle the mixed distributional representations.

Consider a 2D cell with four neighboring vertices V_1, V_2, V_3, V_4. Let, $F_1(x)$, $F_2(x)$, $F_3(x)$, $F_4(x)$ be the respective CDFs representing the uncertainty at the four vertices (as mentioned before the CDFs can be in the form of any continuous family of distribution). Let $\rho_{4\times4}$ be the correlation matrix which captures the multivariate dependency in the cell. Using the three step sampling method as discussed in the example in Chapter 3, we draw multivariate samples for the considered cell. Using the cases of the traditional marching cube algorithm, each of the final multivariate samples, representing the cell configuration, are checked to see if a level-set of the given isovalues passes through it or not. The number of times we find such a cell (multivariate sample) out of the total number of samples drawn, determine the level-crossing probability of the cell. Computing this for all the cells give us a density field that highlights the regions through which the level-set of an isovalue is most likely to pass through. As can be seen in this approach, it is really important to model the uncertainty at each grid location in an optimal way to get a result which is neither biased or erroneous and

also easy to compute for a dataset of high resolution. The procedure to compute the LCP of a single 2D cell is formalized in the pseudocode in Algorithm 2

Algorithm 2 LCP computation for a single 2D cell

1: $S[numSamples] \leftarrow getSamples(\mathcal{N}(\mathbf{0}, \rho_{4\times4}))$ ▷ $\mathbf{0} =< 0,0,0,0 >$
2: $numCrossing \leftarrow 0$
3: **for all** s in $S[.]$ **do** ▷ $\mathbf{s} =< s_1, s_2, s_3, s_4 >$
4: $\quad \mathbf{u} \leftarrow \Phi(\mathbf{s})$ ▷ uniform samples $\mathbf{u} =< u_1, u_2, u_3, u_4 >$
5: $\quad s_i = F_i^{-1}(u_i) \quad \forall i \in \{1,...4\}$
6: $\quad$ **if** isCrossing($\mathbf{s}, isoValue$) **then**
7: $\quad\quad numCrossing \leftarrow numCrossing + 1$
8: $LCP \leftarrow numCrossing/numSamples$

First step in the algorithm is to generate *numSample* multivariate samples from a standard normal distribution with correlation matrix $\rho_{4\times4}$. These samples currently only preserves the multivariate dependency among the four cell vertices. Second step is to transform the samples to uniform marginals using the Property 3.2.2. This is shown in step 4 of the pseudocode 2. The third and the final step involves transforming the uniform marginals $u =< u_1, u_2, u_3, u_4 >$ to their correct forms using the inverse of the predetermined CDF functions (i.e, $F_1^{-1}, F_2^{-1}, F_3^{-1}, F_4^{-1}$). The corresponding 3D version of this implementation will have 8 neighboring random variables for each voxel.

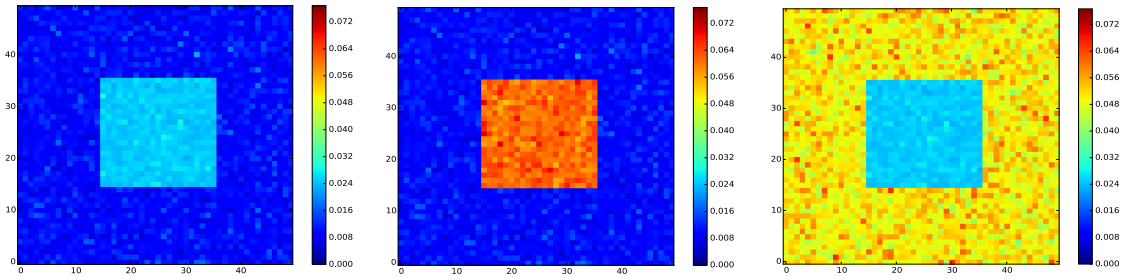
5.4.2 Copula-based Uncertain Vortex Detection

The concept of copula-based analysis in mixed distribution datasets can also be extended to uncertain vector datasets. We use it to extract vortex probabilities along the lines of what Otto et al. [133] proposed. It involves a Monte-Carlo based algorithm of sampling vector fields and using a known vortex detection method (like λ_2-criterion or $Q-$criterion) to compute the probability of observing a vortex core at each grid location.

Unlike the uncertain isocontour extraction approach, vortex detection is a per grid location based computation. Therefore we have to generate samples for each grid location rather than a cell. Sampling at each grid location also involves considering the correlation among the neighboring locations. For a regular 2D dataset, there are at most 4 connected neighbors and each consists of a vector with two components. Therefore, there are 10 (5×2) random variables to be taken into account. Consider a grid location V_0 in a 2D dataset with neighbors V_1, V_2, V_3, V_4. Let, $F_{u_0}(x), F_{u_1}(x), ..., F_{u_4}(x)$ be the marginal CDFs of the respective u-velocities while $F_{v_0}(x), F_{v_1}(x), ..., F_{v_4}(x)$ be the corresponding v-velocity CDFs. Let, $\rho_{10\times10}$ be the correlation matrix for this neighborhood. We then apply a similar copula-based Monte-Carlo sampling to draw sample vectors for location V_0. We used the λ_2 criterion to estimate vortex core probability. Regions with λ_2-criterion below a threshold value (generally 0) is considered highly likely to have a vortex core. Therefore, for all the sampled vector fields we compute the probability of a location having λ_2-criterion less that 0. The resulting density field serves as a visualization of uncertain vortex cores in a mixed vector distribution data. The corresponding procedure to compute the vortex core probability (VCP) for a single grid location is formalized in the pseudocode in Algorithm 3

Algorithm 3 Vortex core probability for a grid location

1: $S[numSamples] \leftarrow getSamples(\mathcal{N}(\mathbf{0}, \rho_{10\times10}))$
2: $numVortex \leftarrow 0$
3: **for all** $\mathbf{s}$ in $S[.]$ **do** ▷ $\mathbf{s} =< s_0, s_1, ..s_9 >$
4: $\quad \mathbf{u} \leftarrow \Phi(\mathbf{s})$ ▷ $\mathbf{u} =< u_0, u_1, ..., u_9 >$
5: $\quad s_i = F_i^{-1}(u_i) \quad \forall i \in \{0, 1, ...9\}$
6: $\quad$ **if** $\lambda_2(\mathbf{s}) < 0$ **then**
7: $\quad\quad numVortex \leftarrow numVortex + 1$
8: $VortexProb. \leftarrow numVortex/numSamples$



(a) (b) (c)

Figure 5.3: Synthetic Data: The KS test for goodness-of-fit, reflects how good are the selected models at each location. The lower the KS test value, the better. The KS test values at each location for (a) mixed distribution field (Gaussian and histogram), (b) only Gaussian at all location and (c) only histogram at all locations is shown in this figure.

5.5 Results and Evaluations

To illustrate the effectiveness of our copula-based analysis on a mixed distribution field, we first show the results on a synthetic dataset. We then apply our method on three different real world ensemble simulation datasets. All computations were performed on a standard workstation PC (Intel i7 at 3.40GHz and 16GB RAM) and implemented using C++.

5.5.1 Synthetic Data

We created a synthetic ensemble dataset of size 50×50 by generating 1000 samples from either a normal distribution or a uniform distribution at each grid location. The mean of the distribution at a location is taken to be the corresponding y-coordinate value at that location and a fixed standard deviation value (1.15) is used across all locations. Figure 5.2(a), colored by the y-coordinate values, illustrates our data creation process. The black rectangle in the image encloses the locations where we used a uniform distribution to generate the samples, while a normal distribution was used for rest of the locations . Also, a high correlation value of $\rho = 0.8$ among the neighboring locations was used to generate the samples. The result

of the Shapiro-Wilk normality test is shown in Figure 5.2(b). A low p-Values in the center block indicates that a Gaussian distribution is not a good choice for modeling the uncertainty at those locations. In this example, we decided to model such locations (i.e, p-Value < 0.1) with histograms. The rest of the locations are modeled using Gaussian distributions.

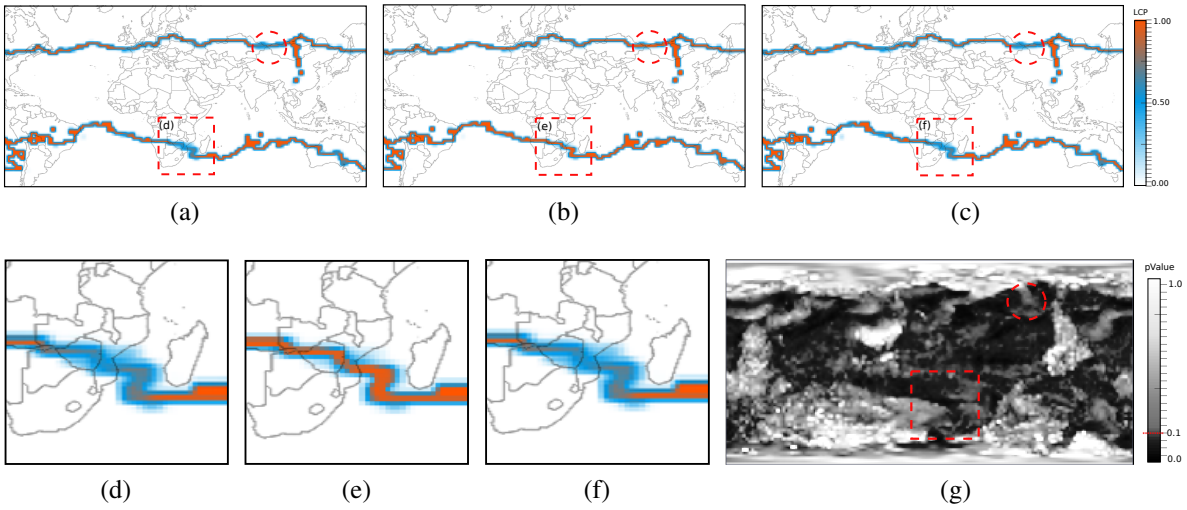
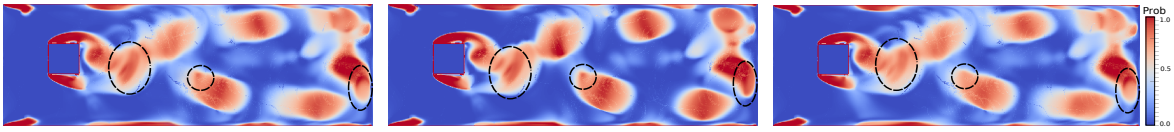


Figure 5.4: Global Ensemble Weather Forecast: The level-crossing probability for isovalues $280K$. (a) The result of using copula-based method on the distribution field with Gaussian and KDE models. (b) The result of using only Gaussian models. (c) The result of using only KDEs. (d) zoomed in view of the selected region in figure (a). (e) zoomed in view of the selected region in figure (b). (f) zoomed in view of the selected region in figure (c). (g) The result of Shapiro-Wilk normality test, a low p-Value indicates the underlying data is less likely to follow normal distribution

We tested our proposed copula-based analysis technique on the resulting mixed distribution field to compute the level-crossing probability (LCP) for isovalue 30 using the technique outlined in Algorithm 2. Figure 5.2(c), (d) and (e) show the LCP fields generated by our method on a mixed distribution field, by using standard multivariate Gaussian distribution [144] and by using multivariate histograms [142] respectively. Since the mean and the

standard deviations for the normal and the uniform distributions are the same, Figure 5.2(d) shows a much smoother result and does not reflect the high uncertainty corresponding to uniform distribution in the middle block. Whereas, when a histogram representation is used for those locations, we are able to see the underlying high uncertainty in those locations (Figure 5.2(c) and (d)). But using the multivariate histogram for all the location is computationally expensive. By selectively choosing histograms only for regions where a Gaussian test fails, we are able to significantly reduce the modeling effort in our proposed method and still generate results similar to the nonparametric version. Our proposed method took 3.4 minutes (including 12 seconds for normality test), whereas, using multivariate histogram took 6.5 minutes. Though the multivariate Gaussian model took 1.10 minutes, it was not able to reflect the true underlying uncertainty in regions where uniform distributions were used to sample. Moreover, the overall memory requirement for the multivariate Gaussian model, proposed copula-based model and multivariate histogram was 0.77MB, 0.89MB and 314.7MB respectively. Besides, the results generated by our method is more reliable compared to the other two approaches because we performed a statistical verification (test) of model at each grid before deciding on a model to pick from. A goodness-of-fit test like the KS test discussed in Section 5.3 can be used to quantify how good are the selected models at each location. Figure 5.3 shows the results of KS-test performed at each grid location for the three cases. Lower the value of the KS test, the better the model represents the data. As can be seen in Figure 5.3(a) the KS test values of our mixed distribution field is lower for all the locations compared to the KS test values for using only Gaussian models (Figure 5.3(b)) and Histograms (Figure 5.3(c)). Besides, the visual validation of the results we quantified the difference in the three LCP results by computing their Root-Mean Square Deviation (RMSD). The LCP field of the multivariate Gaussian has a 12.7% and 13.4% deviation from

the corresponding results of mixed distribution field and multivariate histogram respectively. Whereas, our proposed method produces 1.7% deviation from the multivariate histogram result and still takes 48% less time.



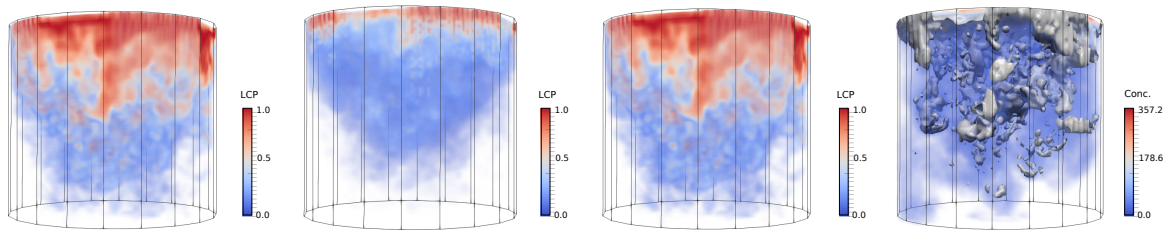
(a) (b) (c)

Figure 5.5: Square Cylinder Vector Ensemble: The results of vortex core probability using (a) copula-based method on mixed distribution field of Gaussian and KDE models, (b) only Gaussian models and (c) only KDE models. The marked regions highlights the difference in vortex structures detected by the three modeling strategies.

5.5.2 Global Ensemble Weather Forecast

We applied our copula-based technique on a 144×73 resolution real world weather forecast ensemble dataset with 21 members, generated by the Global Ensemble Forecast System (GEFS) [3] to compute the probabilistic isocontours of global temperature values. In this example, we used KDE to model the uncertainty at grid locations where normality test fails to show sufficient confidence. Figure 5.4(a) shows the result of our copula-based method on the mixed distribution field (Gaussian and KDE), Figure 5.4(b) shows the result of using multivariate Gaussian models and Figure 5.4(c) shows the result of using nonparametric KDE to model all the grid locations. The result of the Gaussian model is more smoothed out and can be misleading in many regions where a Gaussian distribution is a bad fit as shown in the highlighted regions in the results. The KDE model is able to highlight those uncertain regions which the Gaussian model fails but at the cost of high

computational time and memory usage. Our flexible modeling strategy allows us to use KDE only where Gaussian fails, as a result, we are able to generate similar results as multivariate KDE models but with much less modeling effort. We have marked some of the regions where there are differences in probability values between the three techniques with dotted rectangles and circles in red. The zoomed-in views (Figure 5.4(d),(e),(f)) of the selected rectangular region (southeastern coast of Africa) clearly show the variation of probability values across the three approaches. This is validated by the p-Value results in Figure 5.4(g). The root mean square deviation (RMSD) of the LCP field of the multivariate Gaussian model is 5.9% and 4.1% of the results generated by the KDE model and the mixed model respectively. While the result of the mixed model has only 0.6% deviation from the KDE model. The computation time of the multivariate Gaussian, KDE and mixed models are 3.13, 19.06 and 7.54 minutes respectively. Besides the computational time, the overall memory usage for the multivariate Gaussian, KDE and mixed models are 0.331MB, 1.535MB and 0.414MB respectively.



(a) (b) (c) (d)

Figure 5.6: Salt Concentration Ensemble: Results of uncertain isocontours representing the viscous fingers for salt concentration level of 50 and generated by (a) copula-based technique on a mixed distribution field, (b) assuming Gaussian model at each grid location, (c) assuming trimodal GMM at each grid location. (d) shows the shape of an isosurface from a randomly chosen ensemble member.

5.5.3 Square Cylinder Vortex Ensemble

We tested our copula-based technique on vector distribution fields to detect vortex core probabilities. The dataset represents the flow field behavior around a square shaped cylinder in a 600×200 resolution 2D grid. A set of 10 simulations were generated with slightly different parameters to model the uncertainty in flow structures. Figure 5.5 shows the results of vortex probabilities (λ_2-criterion values) as computed by the method outlined in Section 5.4.2. Figure 5.5(a) shows the result generated by our copula-based method where we used either a Gaussian or a KDE to model uncertainty of the vector components at each grid location. Figure 5.5(b) and (c) show the results generated by assuming only Gaussian and only KDE respectively across all the locations. The probability values are different when using only Gaussian as compared to using only KDE. The region marked by the dotted black circles in Figure 5.5(b) and (c) highlights some of those differences. Result generated by our proposed method as shown by Figure 5.5(a) is a mixed representation of both the type of distributions, therefore, the regions with high certainty of following a Gaussian distribution are able to show the probable vortex structures that Figure 5.5(b) reflects. Whereas, regions where we used KDE to model the data where able to show results similar to Figure 5.5(c). For example, the region marked by the right-most black circle highlights a feature which was missed out by assuming Gaussian distribution, but was captured by both KDE and our mixed representation. The computation time of using multivariate Gaussian distribution was 10.4 minutes , while, the time for using only KDE was 52 minutes. On the other hand, our copula-based method took 28.5 minutes (including the time to perform normality test for both the vector components at each location), but generated results which are less prone to be erroneous compared to the other two approaches because of the prior statistical test for model selection. The RMSD of the result generated by the Gaussian model is 6.5% of the

result generated by the mixed model and 8.2% of the KDE model. On the other hand, result of the mixed model has a deviation of 2% compared to the KDE model results. While, the overall memory usage for the multivariate Gaussian, mixed model and multivariate KDE are 23.23MB, 27.56MB and 453.3MB respectively.

5.5.4 Salt Concentration Ensemble

Our final dataset is a 3D dataset ($64 \times 64 \times 64$) from the field of fluid dynamics [4]. It represents the density field of concentration of salt particles dissolving in a fluid contained in a cylindrical container. Various boundary conditions were used to study an interesting fluid property called viscous fingers, the shape of which is represented by the isosurfaces of salt concentration values. The data comprises of 50 ensemble members and we selected a single time-step to perform our study. We computed the LCP for isosurface of concentration value 50. Figure 5.6 shows the results of the uncertain viscous fingers (isosurfaces). To model uncertainty at each grid location, we decided to use only parametric models because a nonparametric model like KDE will be computationally very expensive both in terms of time and storage because the number of ensemble members are relatively high for this dataset(50). Instead, we decided to use Gaussian Mixture Models (GMM) of different modes to model the uncertainty. Using the Bayesian Information Criterion (BIC) test explained in Section 5.3, we decide on a optimal number of modes for GMMs (out of 1, 2 and 3) at each grid location. Figure 5.6(a) shows the LCP results of using a copula-based technique on a mixed distribution field of unimodal, bimodal and trimodal GMMs. We compared this against distribution fields where we used only Gaussian distribution (equivalent to unimodal GMMs) and only multivariate GMMs with 3 modes, the results of which are shown in Figure 5.6(b) and (c) respectively. One interesting property of the isosurfaces in this dataset

is that there are many disjoint components across the space as shown in Figure 5.6(d) (it shows the isosurface for isovalue 50 in a randomly chosen ensemble member). Because of this structure, Gaussian model assumption produces an overly smoothed-out result as shown in Figure 5.6(b) and no clear finger structures are visible. Whereas, the results produced by using a multivariate GMM with 3 modes is able to reveal those finger-like structures as shown in Figure 5.6(c). However, the overall estimation of multivariate GMM with 3 modes for each voxel in the dataset was very expensive and took 1 hour 10 minutes to generate the LCP field. On the other hand, our proposed copula-based method on a mixed distribution field took only 35.6 minutes (including the 1.5 minutes for the BIC test) and produced similar results as the multivariate GMM approach (RMSD value of 1.5%). Using only Gaussian models it took just 7.2 minutes but the result was not trustworthy (RMSD value was 22.3 % of the GMM result). The overall memory usage for multivariate Gaussian model, proposed mixed GMM model and multivariate GMM with 3 modes are 30.09MB, 34.07MB and 291.05MB respectively.

5.6 Discussion

Table 5.1 summarizes the overall performance (time and memory) and quantitative comparison values (RMSD) for the various example case studies. One important thing to note here is that the performance gain is attributed to the fact that our proposed uncertainty modeling technique allows us to model the univariate distribution at each grid location and the multivariate dependency separately. Because of this flexibility we can use the computational expensive models only when it is really necessary, thus, bringing down the cost without sacrificing on quality. Based on the modeling requirements and the degree of correctness required in the task, the performance can vary and is strictly controlled by

Table 5.1: Performance Summary of the Example Case Studies

	Complexity			RMSD
	Model	Time (mins)	Memory (MB)	
Synthetic Data	Mixed	3.4	0.89	1.7% (Mixed vs Hist)
	Hist	6.5	314.7	12.7% (Mixed vs Gaussian)
	Gaussian	1.1	0.77	13.4% (Gaussian vs Hist)
GEFS	Mixed	7.54	0.41	0.6% (Mixed vs KDE)
	KDE	19.06	1.53	4.1% (Mixed vs Gaussian)
	Gaussian	3.13	0.33	5.9% (Gaussian vs KDE)
Sq. Cylinder	Mixed	28.5	27.56	2% (Mixed vs KDE)
	KDE	52	453.3	6.5% (Mixed vs Gaussian)
	Gaussian	10.4	23.23	8.2% (Gaussian vs KDE)
Salt Conc.	Mixed	35.6	34.07	1.5% (Mixed vs GMM_3)
	GMM_3	70	291.05	21.6 % (Mixed vs Gaussian)
	Gaussian	7.2	30.09	22.3 % (Gaussian vs GMM_3)

the complexity of the univariate models involved. Even the quality of the result is limited only by the quality that the individual models offer. In this work, we compared our results with the other Monte-Carlo based uncertain feature extraction methods [142, 144] because a copula-based method is inherently a Monte-Carlo process. Closed-form solution in copula-based models is not so straight-forward to derive for mixed univariate marginals [153], therefore, we did not compare our approach with the work proposed by Athawale et al. [12].

The effectiveness of the result generated by the proposed method is highly dependent on the effectiveness of the statistical test performed to decide the distribution type. As was mentioned in Section 5.3, one important factor which determines the effectiveness of a statistical test is the sample size. Small sample sizes are susceptible to generate unreliable test results. A general rule of thumb in statistic is to fall back upon a nonparametric model when the parametric tests do not give reliable results for smaller sample sizes. Therefore, one must be careful and critical while performing such statistical test, especially on small sample sizes, which is common for many real world ensemble experiments.

Chapter 6: Information Guided Exploration of Scalar Values and Isocontours in Ensemble Datasets

6.1 Introduction

Ensemble datasets are one of the primary sources of uncertain datasets in scientific studies. For an effective exploration of scalar ensemble datasets, use of uncertain isocontours has been a popular method and has attracted significant attention recently. Besides the popular spaghetti plot, various techniques like contour-boxplot [183], circular glyphs [158], contour variability-plot [56], probabilistic marching-cubes [144] have been proposed to visualize ensemble isocontours. All these existing works conduct uncertainty analysis by assuming that a particular scalar value of interest is already known. Since not all scalar values have the same degree of isocontour uncertainty, scientists trying to study the effects of multiple simulations/runs on a range of scalar values may lack a thorough understanding of the uncertainty associated with all the values. Such an analysis of uncertainty across all the scalar values can help scientists as well as visualization practitioners to identify interesting scalar values for further analysis using some of the aforementioned isocontour analysis techniques. Note that, for a single selected scalar value, not all the individual members contribute equally to the overall uncertainty of the ensemble isocontour structure. The existing ensemble isocontour analysis techniques do not offer insights into the contribution

of individual members towards the uncertainty. Understanding the contribution of individual members is essential not only to comprehend the important simulations/runs from a large collection of members but also to understand the effect of uncertainty on the scalar value in the experiment. To date, a single coherent analysis framework that analyzes the uncertainty of both the *scalar value range* as well as *the ensemble isocontours of an individual scalar value* is mostly missing. Our work is an effort to fill this gap and address the above unaddressed facets of uncertainty analysis in ensemble datasets.

In this chapter, we introduce a two-stage information-theoretic framework for the exploration of the scalar values as well as their corresponding ensemble isocontours for ensemble scalar datasets. In the first stage, to understand the effect of uncertainty on all the scalar values and eventually guide the user towards selecting interesting scalar values for further analysis, we evaluate the ensemble isocontour variations across all the scalar values. Using two *specific information* measures, we compute the *predictability* and *surprise* of specific scalar values. *Predictability* of a scalar value conveys the relative similarity of the corresponding ensemble isocontours, while, *surprise* conveys the relative importance of the scalar values in the field. Surprise along with predictability help us quantify the importance as well as the uncertainty of the scalar values in ensemble datasets. Therefore, the non-trivial problem of evaluating the uncertainty of the ensemble isocontours for all the individual scalar values can be efficiently addressed by our proposed analysis method. In order to facilitate the identification and selection of interesting scalar values for further exploration, we present an interactive scatter plot view which is linked to a violin plot [78] view showing the distribution of individual predictability values of the members. For values with a high variation in their predictability, it is worth investigating the cause of uncertainty and identifying the contribution of individual member isocontours towards the overall uncertainty. This

leads to the next stage of our information guided exploration of the ensemble isocontours of a single scalar value.

We propose a *conditional entropy* based algorithm to identify the contribution of individual members towards the overall uncertainty of the ensemble isocontours for a single scalar value. Since individual members contribute differently toward the total structural uncertainty of an ensemble of isocontours, use of conditional entropy provides an effective measure to identify the relative importance of the members by accounting for the information overlap among their corresponding isocontours. To assist the users with exploration of the relative importance of individual isocontours for a selected scalar value, we provide an interactive *information gain curve*. This curve conveys the overall information gained about the complete set of ensemble isocontours by selecting a specific sequence of members. Besides exploring the relative importance of each isocontour, it also helps us to select representative subsets/samples of isocontours from large number of members that can represent the uncertainty of all the members.

To summarize, the major contributions of our work are twofold:

- Using specific information measures, we evaluate the ensemble isocontour uncertainty of all the scalar values in an efficient and effective way.
- For a single scalar value, we propose a conditional entropy based approach to identify the contributions of individual members towards the uncertainty of the ensemble isocontours.

6.2 System Overview

In this section, we provide a brief overview of the proposed information guided exploration of scalar values and their isocontours in ensemble datasets. Figure 6.1 gives a

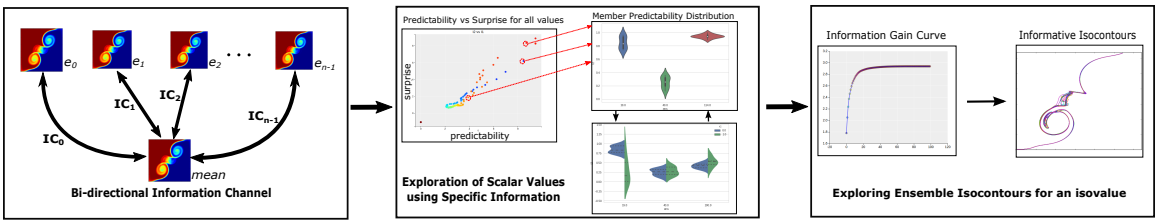


Figure 6.1: A schematic overview showing the main stages of our proposed information-theoretic method.

schematic overview of our method. To explore the uncertainty associated with ensemble isocontours of all the scalar values, we use specific information measures which evaluate the predictability and surprise of specific scalar values. We first establish bi-directional information channels, IC_i, between the scalar fields of individual members i and the mean field as illustrated in Figure 6.1. In the absence of a ground-truth, mean field is a popular and widely accepted single field representation of the members [101, 134, 148]. By computing the specific information measures of each member against the mean reference field, we can compute the total predictability and total surprise of the scalar values. Scalar values with high total predictability indicate low uncertainty of their corresponding ensemble isocontours, whereas, low predictability indicates high variation in their ensemble isocontours, i.e, high uncertainty. On the other hand, surprise is an indicator of the importance of the feature based on the frequency of the corresponding scalar values. We convey this information in the form of an interactive predictability versus surprise scatter plot view. This is complemented with two linked violin plot views which visualize the distributions of individual predictability values for the user selected scalar values in the scatter plot. For scalar values showing high deviation of its individual predictability values, it is important to understand the contribution of individual members to the overall uncertainty of that value. In the second stage of our exploration, we investigate the contribution of each member to the overall uncertainty of

the ensemble isocontours of a single scalar value. We propose a conditional entropy based isocontour selection algorithm to identify the most informative isocontours. To assist the users in selecting the informative isocontours, we create an interactive information gain curve that conveys the information gained by selecting a particular set of members.

6.3 Specific Information Based Scalar Value Exploration

The ensemble isocontours corresponding to different scalar values show different degrees of structural variations. This is because all the scalar values are not equally affected by uncertainty in ensemble simulation experiments. Exhaustively extracting the individual ensemble isocontours for all the scalar values and analyzing their uncertainty is computationally prohibitive for large number of ensemble members. Instead of extracting the isocontours individually, we use information-theoretic measures called *specific information*, which allow us to evaluate the structural variations and hence, the uncertainty of the ensemble isocontours. Specific information measures are essentially decompositions of mutual information, that let us evaluate the information content of specific instances/realizations of a random variable. As a result, by considering the scalar fields as random variables, we can efficiently analyze multiple scalar values using specific information measures. In particular, we use I_1 and I_2 specific information measures [24, 42], which evaluate the *surprise* and *predictability* of specific scalar values respectively. *Predictability* offers a measure of how similar the corresponding ensemble isocontours of a specific scalar value are, while, *surprise* corresponds to the relative importance of the scalar value in the field based on their frequency. Instead of computing these measures for all pairs of ensemble members, which is an exponentially expensive task for large ensemble systems, we evaluate them for each member against the *mean scalar field*. For a system with n ensemble members, $\{e_1, e_2, ...e_n\}$, the value at a grid

location (x,y), in the mean scalar field can be denoted as $\frac{1}{n}\sum_{i=1}^{n} e_i(x,y)$. The mean scalar field acts as a frame of reference for comparing the individual specific information measures of the members. This also allows us to quantify how well the scalar values are represented by the mean scalar fields, which is often utilized by scientists to aggregate or summarize ensemble results into a single field [134, 148]. To compute these values, we first establish bi-directional *information channels* between each of the member scalar fields and the mean field.

6.3.1 Information Channels

Information channel between two scalar fields can be denoted as $X \rightarrow Y$, where X and Y are the input and output random variables representing the two scalar fields. The three basic components of the channel $X \rightarrow Y$ are:

- *Input distribution* $p(X)$, which represents the normalized frequency of each scalar value x in the distribution of X.
- *Conditional probability distribution* $p(Y|X)$, which expresses how the distribution of each of the scalar values (i.e, x) of the input field X match with the distribution of output field Y.
- *Output distribution* $p(Y)$, which represents the normalized frequency of each scalar value y in the distribution of Y.

Throughout this chapter, we use x to refer to a single scalar value in the scalar value distribution of the field corresponding to X (i.e, the bin center in the corresponding histogram).

In our work, we have used bi-directional information channels between the member fields and the mean as illustrated in the first stage in Figure 6.1. The direction "*mean*$\rightarrow$

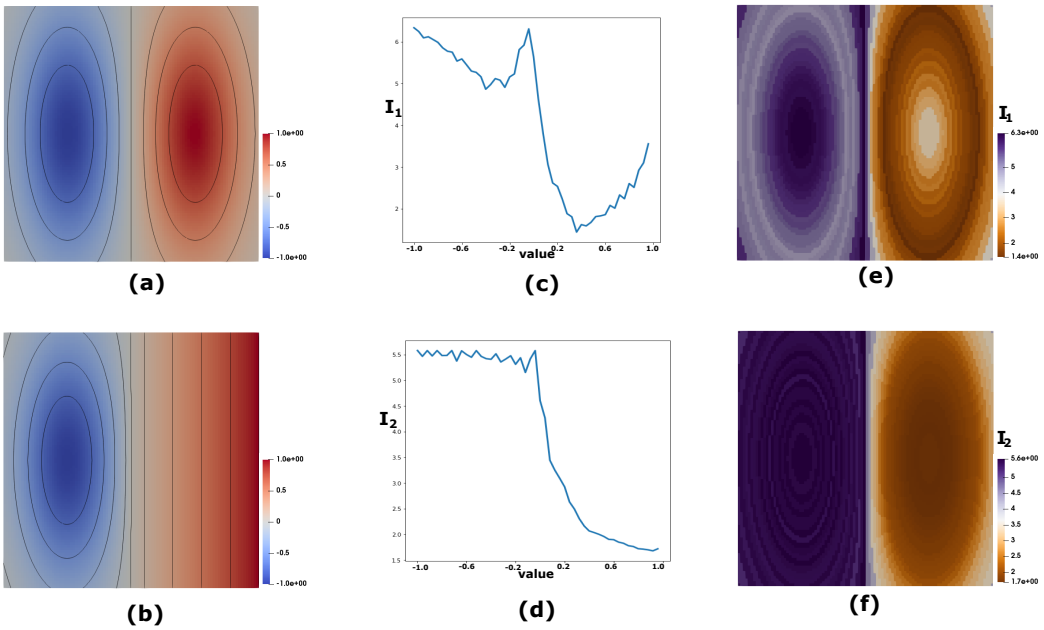


Figure 6.2: Synthetic Data: (a) Scalar field with two Gaussian structures. (b) Scalar field with one Gaussian structure in a linearly increasing field. (c) The I_1 plot for the scalar values of (a) w.r.t the field in (b). (d) The I_2 plot for the scalar values of (a) w.r.t the field in (b). (e) The I_1 values color-mapped to the scalar values of the field (a). (f) The I_2 values color-mapped to the scalar values of the field (a).

member", where input is the mean field and the output is a member field, helps us quantify how much information the member scalar fields retain about the scalar values of the mean field. In other words, it allows us to quantify how well the spatial distribution of the scalar values in the mean field represent the corresponding distribution in the individual member fields. On the other hand, the reverse direction i.e, "*member*$\rightarrow$ *mean*", lets us evaluate how much information the mean field possess about the scalar values in the individual member fields. Before we describe in detail how to compute the specific information measures of a channel, we briefly introduce two closely related information theory concepts i.e, *entropy* and *mutual information*.

6.3.2 Entropy

Entropy provides a measure of the uncertainty associated with a random variable. If $p(x)$ is the probability of event x ($\sim X$), then the uncertainty associated with X can be described by Shannon's entropy as;

$$H(X) = -\sum_{x \in X} p(x) \log p(x) \tag{6.1}$$

Similarly, for two random variables X and Y, the joint entropy can be described as;

$$H(X,Y) = -\sum_{y \in Y} \sum_{x \in X} p(x,y) \log p(x,y) \tag{6.2}$$

6.3.3 Mutual Information

Mutual information is a measure of the information overlap between two random variables. For the two random variables X and Y, mutual information can be denoted as;

$$I(X,Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{6.3}$$

Mutual information between two random variables can also be described as the amount of uncertainty reduced about one variable after observing the other variable. In other words, after observing the variable X, the amount of uncertainty reduced about variable Y can be denoted as;

$$I(X,Y) = H(Y) - H(Y|X) \tag{6.4}$$

6.3.4 Specific Information

In our work, we use two types of specific-information measures, namely, I_1 and I_2 [42], which are essentially two different forms of decomposition of the popular mutual information measure. They were also referred to as *Surprise* and *Predictability* by Bramon et al. [24] based on the type of information that they quantify. We apply these measures to propose

an efficient analysis strategy for evaluating the effect of uncertainty across multiple scalar values in ensemble datasets.

Surprise(I_1): Surprise or I_1 of x (instance of input distribution X) with respect to the output distribution Y is given as:

$$I_1(x;Y) = \sum_{y \in Y} p(y|x) \log \frac{p(y|x)}{p(y)} \tag{6.5}$$

$I_1(x;Y)$ essentially represents the Kullback-Leibler divergence [97] between $p(Y|x)$ and $p(Y)$. A high $I_1(x;Y)$ indicates that given the observed value x in X, certain low frequency occurrences $y \in Y$ have become more probable, which account for an unlikely or surprising behavior. I_1 is an effective tool to understand the importance of a feature corresponding to a scalar value [21] based on the frequency of the value (i.e, the size of the feature). In general, for scientific datasets, low frequency scalar values correspond to interesting foreground features (high surprise), while very high frequency values often correspond to background features (low surprise). Therefore, I_1 or surprise helps to distinguish such important scalar values across the ensemble members in scientific datasets.

Predictability(I_2): Predictability or I_2 is based on the change of entropy of the channel and is given as:

$$\begin{aligned} I_2(x;Y) &= H(Y) - H(Y|x) \\ &= -\sum_{y \in Y} p(y) \log p(y) + \sum_{y \in Y} p(y|x) \log p(y|x) \end{aligned} \tag{6.6}$$

$I_2(x;Y)$ gives the amount of reduction in uncertainty about Y after observing the data value x. Generally, a high $I_2(x;Y)$ indicates that given the observed value x in X, we can predict the corresponding y's in Y with high confidence. Therefore, high predictability corresponds to less uncertainty of the scalar values and vice-versa.

Relationship with Mutual Information: Both the specific information measures are different decompositions of the mutual information measure. Therefore, the average information gained from I_1 and I_2 for all the specific realizations of a variable is equal to the overall mutual information. For I_1, using Equation 6.3, this relationship can be expressed as follows;

$$\begin{aligned}\sum_{x\in X} p(x)I_1(x;Y) &= \sum_{x\in X}\sum_{y\in Y} p(x)p(y|x)\log\frac{p(y|x)}{p(y)} \\ &= \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \\ &= I(X,Y) \end{aligned} \tag{6.7}$$

Further, for I_2, using Equation 6.4, this relationship can be expressed as;

$$\begin{aligned}\sum_{x\in X} p(x)I_2(x;Y) &= \sum_{x\in X} p(x)H(Y) - \sum_{x\in X} p(x)H(Y|x) \\ &= H(Y) - H(Y|X) \\ &= I(X,Y) \end{aligned} \tag{6.8}$$

Synthetic Data: Figure 6.2 highlights the utility of these two measures in evaluating the uncertainty of scalar values using two synthetic datasets. Consider two synthetic datasets as shown in Figure 6.2(a) and (b). Let random variables X and Y represent the two scalar fields shown in Figures 6.2(a) and (b) respectively. X is created by mixing two Gaussian fields, while Y consists of a single Gaussian added to a linearly increasing field. The highlighted isocontours in black, as shown in Figures 6.2(a) and (b) reveal the underlying scalar field structure. Consider an information channel $X \rightarrow Y$, which can be used to calculate the specific information measures I_1 (surprise) and I_2 (predictability) of specific scalar values $x \in X$. The results of these two measures for all x's in X is shown by the plots in Figures 6.2(c) and (d) respectively. Figures 6.2(e) and (f) show the X field color-mapped

to their I_1 and I_2 values of the scalar values respectively, which provide a spatial perspective to the surprise and predictability of the field X. As can be seen in Figures 6.2(d) and (f), the I_2 (predictability) values for the low scalar values corresponding to the left half of X have high predictability about the corresponding scalar values in Y. However this predictability sharply decreases for the higher values which corresponds to the right halves of X and Y. Also, within this high value range, predictability drops gradually for the higher scalar values. This is because as the radius of the elliptic isocontours on the right half of X decreases (i.e, for high scalar values), the predictability of these values with respect to the corresponding vertical contours in the right half of Y decreases at the same time.

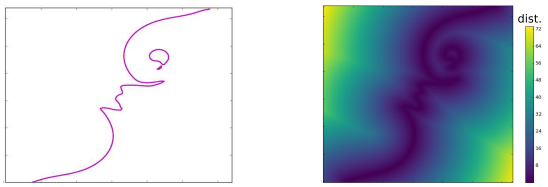
With respect to I_1 or surprise, which is an indicator of the importance of features based on the frequency of scalar values, we see a different trend. As shown in Figures 6.2(c) and (e), the surprise is usually high for the low frequency values and low for the high frequency values. This trend is clearly visible in the left and the right halves of X. The high degree of feature alignment on the left half results in higher overall surprise in the left half than the right half of X. However, within these two halves, we observe the trend of low surprise for high frequency values and vice-versa. These results also corroborate the fact that I_1 is more sensitive to the feature size as compared to I_2. Therefore, I_1 and I_2 together help us in profiling the uncertainty of all the scalar values in a field without having to extract the corresponding isocontours and analyzing them separately. Next, we show the utility of these measures in exploring the uncertainty of scalar values in ensemble scientific datasets.

6.3.5 Exploration of Scalar Values

In our work, we compute the surprise and predictability measures for all the information channels as illustrated in Figure 6.1. The I_1 and I_2 values of individual channels quantify

the information content between the mean reference field and the corresponding member field. For each scalar value, aggregated values of the two measures are obtained by summing the individual I_1 and I_2 values across all the members, thus representing the *total surprise* and *total predictability* respectively. As the first step towards an effective exploration of the scalar value range, we present an interactive scatter plot view of the total predictability versus total surprise of the scalar values. This plot depicts the overall variation or uncertainty of the ensemble isocontours of all the scalar values. The values with high total predictability (I_2) and high total surprise (I_1) refer to low uncertainty of their corresponding isocontours, while, a low predictability score highlights high variation among the ensemble isocontours for that value. On the other hand, surprise determines the importance of the scalar value in the field. Therefore, our scatter plot view reveals the uncertainty as well as the relative importance of the scalar values in the studied ensemble field.

However, the scatter plot only conveys the overall uncertainty of the scalar values, it does not convey how the individual ensemble members are contributing to the uncertainty. To understand how the individual predictability values of the members are distributed we show a secondary visualization in the form of violin plots [78] for the user-selected scalar values in the scatter plot. Violin plots can visualize both the order statistics (like a traditional box-plot) as well as the shape of the distribution of values. Therefore, we use it to reveal the distribution of individual predictability values. The shape of the violin glyph gives us a picture of how agreeing are the individual predictability scores of the members. We offer an additional split-view of the violin plot, where we visualize the distribution of individual predictability measures for both the directions of the bi-directional channels separately. The left side corresponds to the predictability of the values in the mean field (i.e, "*mean* $\rightarrow$



(a) (b)

Figure 6.3: (a) example isocontour (b) corresponding distance field transformation.

member"), while the right side shows the predictability distribution of the member fields (i.e, "*member* $\rightarrow$ *mean*").

For values with high total predictability and low variations, it can be concluded that the mean field is a reliable representation and from an information-theoretic perspective, retains good amount of information about the individual members. However, for values showing low total predictability and high variation of individual predictability it is important to understand how the individual member isocontours contribute to the high uncertainty in detail. This leads us to the second stage of our information guided exploration, where we investigate the contribution of individual isocontours towards the overall uncertainty for a scalar value.

6.4 Conditional Entropy Based Isocontour Exploration

In the second stage of our exploration process, we provide the users with information-based guidance in analyzing the isocontours of a selected isovalue. Analysis and visualization of a large number of isocontours is not a trivial task. The members in an ensemble of isocontours are often of varying shapes. The overall variation of shape of all the members gives an idea of the uncertainty of that isovalue. But not all isocontours contribute equally to the overall variations of the ensemble isocontours. We propose a novel conditional entropy based contour exploration technique which will not only provide a quantitative measure for

the contribution of individual members to the overall uncertainty of the ensemble isocontours but will also help the users in selecting an informative sample/subset of contours from a large number of varying members.

To apply information-theoretic measures on isocontours they need to be first transformed into random variables that capture their respective shape information. This can be achieved by deriving an implicit representation of the isocontours. One popular implicit representation of isocontours is the distance field transformation [80]. The distance field is a spatial representation of a geometric object where at each point in the field we store the distance of that point to the closest point on the object [87]. If C_θ is the isocontour for isovalue θ then the Euclidean distance field transformation at a location p in the field is given as: $D_\theta(p) = \min_{\forall q \in C_\theta} dist(p,q)$. Figure 6.3 shows an isocontour and its corresponding distance field transformation. This implicit field for an isocontour can be considered as a random variable. Henceforth, by entropy of an isocontour we mean the entropy of its corresponding distance field transformation.

6.4.1 Conditional Entropy

Since our goal in this section is to select structurally informative isocontours with maximum contribution to the uncertainty of the system, choosing these contours based solely on their individual uncertainty or entropy does not suffice. Such entropy-based selection does not take into account the information overlap among the other existing contours of the system. For example, if there are two structurally similar contours that have high individual entropies, choosing just one of them will suffice in the current context because of their high information overlap. When one of these two contours is selected, the uncertainty remaining about the other will diminish significantly. In information theory,

conditional entropy provides a method to select informative variables from a system of variables by taking into account the information overlap among all the system variables. For n given variables, $X_1,...,X_n$, if k variables $X_{i1},...,X_{ik}$ are known then the amount of uncertainty left in the system is given by the following conditional entropy formula:

$$H(X_1,...,X_n|X_{i1},...,X_{ik}) = H(X_1,...,X_n) - H(X_{i1},...,X_{ik}) \tag{6.9}$$

Here, $H(X_1,...,X_n)$ represents the joint entropy of the set of n variables $X_1,...,X_n$ and is computed as:

$$H(X_1,...,X_n) = -\sum_{x_1 \in X_1} ... \sum_{x_n \in X_n} p(x_1,...,x_n) log(p(x_1,...,x_n)) \tag{6.10}$$

where, $p(x_1,...,x_n)$ is the joint probability distribution of the variables. Since the joint entropy of a set of variables quantifies the total amount of the uncertainty or the information content of those variables, conditional entropy quantifies the information gained about a system of variables $X_1,...,X_n$ when a subset of k variables $X_{i1},...,X_{ik}$ are known. Thus, the non-trivial problem of identifying the contribution of individual members to the overall structural variation becomes the task of computing the information overlap among their distance fields from an information theory point-of-view.

6.4.2 Informative Isocontour Selection

As shown in Equation 6.9, we can use conditional entropy to identify the contribution of individual isocontours towards the overall uncertainty/entropy of the system of ensemble isocontours. Using a greedy approach, we iteratively select the contour which minimizes the uncertainty (i.e, entropy) left in the system. In each iteration, after a contour X_i has been selected, the uncertainty left in the system is essentially the conditional entropy of the system of ensemble isocontours given that the selected contour is known i.e, $H(X_1,...,X_n|X_i)$. The

corresponding amount of entropy/uncertainty reduced by selecting a contour is referred to as its *information gain* in our work. Information gain of a member isocontour quantifies the informativeness of the member in the ensemble of isocontours. The sequence of isocontours, thus generated by the iterative process can be used to select a subset/sample of contours which can represent the uncertainty of the complete ensemble system.

Algorithm 4 Informative Isocontour Selection Algorithm

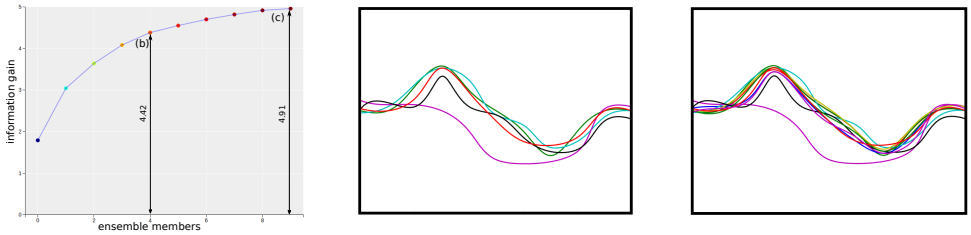
```
1:  allVar := [C^0, C^1, ..., C^{n-1}]
2:  infoVar := empty  stack                          ▷ list of informative contours
3:  infoGain := empty  stack                         ▷ amount of information gain
4:  while allVar ≠ ∅ do                              ▷ check the info gain of all contours
5:      maxGain ← −1.0
6:      importantVar ← ∅
7:      for all c in allVar do
8:          infoVar.push(c)
9:          je ← getJointEntropy(infoVar)            ▷ je: joint entropy
10:         if je > maxGain then
11:             maxGain ← je
12:             importantVar ← c
13:         infoVar.pop()
14:     infoVar.push(importantVar)
15:     infoGain.push(maxGain)
16:     allVar.delete(maxVar)
```

Algorithm 4 explains the steps involved in constructing the sequence of informative isocontours. As shown in the pseudocode, the sequence of informative isocontours are pushed into the stack $infoVar$ and the corresponding cumulative gain of information is pushed into another stack, $infoGain$. The procedure, $getJointEntropy(infoVar)$ returns the joint entropy marginalized on the set of variables present in the parameter $infoVar$. $getJointEntropy()$ queries from the joint histogram of the ensemble isocontours. Efficient construction of joint histograms of large number of variables is not a trivial task. A naive

way to store joint histogram is to store it as a multi-dimensional array. However, as the number of variables increases, the memory cost of using multi-dimensional array increases exponentially, which makes this naive representation computationally prohibitive for a large number of variables. Lu et al. [109] presented a compact representation to store joint histogram by utilizing the sparse property. We used their approach to create a joint histogram for ensemble isocontours. Besides the storage benefits, this representation incorporates many histogram query operations as well. We only need to compute the joint histogram from all isocontours once, and then we can derive other histograms that are needed efficiently based on the histogram marginalization operation.

To facilitate such information guided exploration of the ensemble isocontours, we visualize the information gain values of selecting the ensemble members in a plot called the *information gain curve* (figure 6.4a). The vertical axis of the plot represents the information gain while the horizontal axis represents the sequence of ensemble members. By following along the information gain curve from left to right, users can select the sequence of most informative ensemble members for a particular isovalue. Apart from visualizing the contribution of each members, this interactive information gain curve can also guide the users in selecting subsets of informative isocontours which can represent the total uncertainty of the system (including the anomalous members). In ensemble systems with hundreds of members it is especially important to understand the most contributing members for quick analysis and simpler visualization.

Figure 6.4(a) shows the *information gain curve* for synthetically created 10 member isocontours. Bezier curves were drawn using varying control point locations to create this synthetic data. In this example, the total entropy/information of the system is 4.91. As can be seen, in the corresponding information gain curve, by selecting the first 5 informative



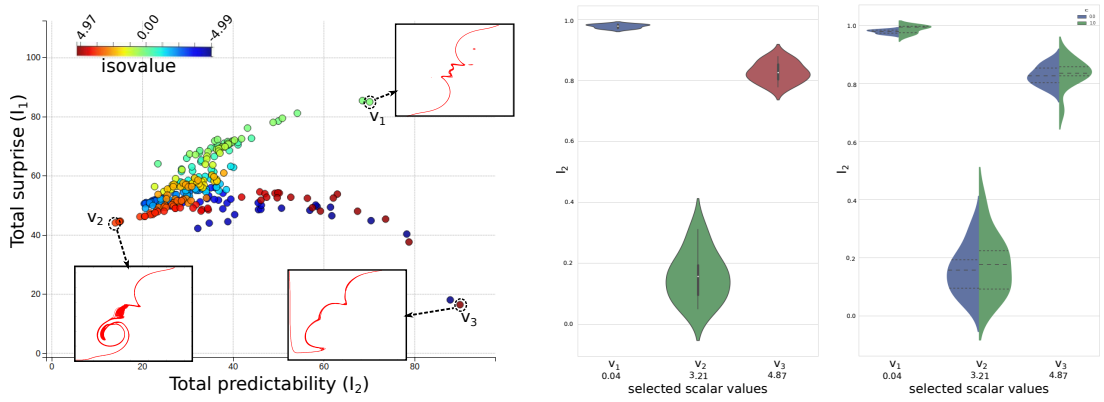
(a) (b) (c)

Figure 6.4: Informative isocontour selection in synthetic dataset: (a)Information gain curve for all 10 members. Each point on the plot corresponds to a member isocontour arranged from left to right in the descending order of their informativeness. The vertical axes show the maximum (cumulative) information gained about the system by selecting a sequence of members along the horizontal axis. (b)The isocontour plot of the top 5 most informative isocontours. (c)The spaghetti plot of all 10 isocontours. As can be seen, the top 5 isocontours (b) retain about 90% information of the complete system (c).

members the information gained is 4.42 which is 90% information of the whole system. The spaghetti plots of the 5 most informative contours and all the 10 member contours are shown in Figure 6.4(b) and (c) respectively. As can be seen, the 5 informative contour conveys almost the same uncertainty information as shown by all the members. This is particularly useful for systems with large number of members (which is shown in Section 5), as it helps in identifying the most important members relevant for uncertainty analysis for that isovalue.

6.5 Results

To demonstrate the effectiveness of our information-theoretic approach in exploring the scalar values and their ensemble isocontours, we test it on three different types of ensemble datasets. The datasets have varying degrees of uncertainty and are selected from the field of material sciences, weather-forecasting and ocean-modeling. All the experiments were conducted on a standard workstation PC powered by Intel Core i7-2600 quad-core CPU running at 3.40GHz with 16GB of RAM.



(a) (b) (c)

Figure 6.5: Scalar value exploration of material density ensemble. (a) Interactive Scatter-plot view of the total predictability vs total surprise of the scalar values. v_1 corresponds to a scalar value with high predictability and high surprise, v_2 corresponds to low predictability and low surprise i.e, high uncertainty, while, v_3 corresponds to high predictability but low surprise. (b) Violin-plot view showing the distribution of individual predictability values for selected scalar values. (c) Split violin-plot view showing the distribution of the predictability values for the two directions of the bi-directional information channel.

6.5.1 Material Density Ensemble:

Our first ensemble dataset is generated by performing multiple lock-exchange experiments [117] with different parameter setting. The experiment involves separating a light fluid from a heavy fluid with a barrier and then gradually letting them mix by releasing the barrier. We used a dataset with 100 ensemble simulation runs and a spatial resolution of 128×128.

Figure 6.5(a) shows the interactive scatter-plot view of the total predictability versus total surprise results for the scalar values, which are color-mapped to their values. Scalar values with high total predictability and high total surprise indicate that the corresponding ensemble isocontours of the scalar values are less uncertain, as is shown by the inset figure for the highlighted value v_1. On the other hand, for values with low predictability and surprise the uncertainty of the ensemble isocontours are relatively high, as is shown for the highlighted

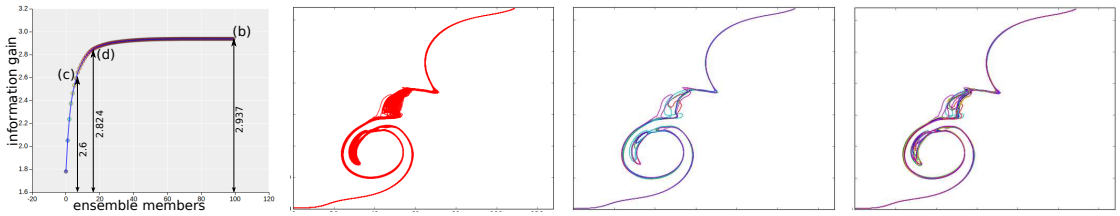
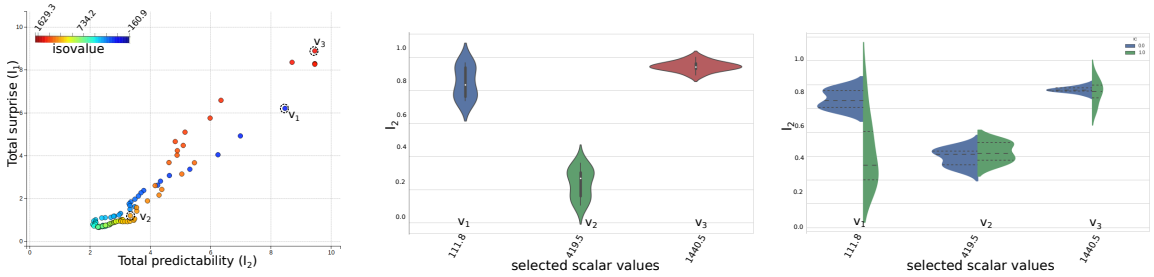


Figure 6.6: Informative isocontour exploration for material density value of 3.218. (a) The information gain curve for all 100 ensemble members. The vertical axes show the maximum information gained about the system by selecting a sequence of members in the plot.(b) The spaghetti plot of all 100 isocontours. (c) The isocontour plot of the top 7 most informative isocontours. (d) The plot of top 15 informative isocontours. (c) and (d) are able to reveal the spatial layout of the isovalue with lesser number of members.

value v_2. The third highlighted value v_3, acts as an example for another interesting set of scalar values with high total predictability but low surprise. The high predictability of v_3 indicates that the corresponding ensemble isocontours are less uncertain, but the low surprise indicates high frequency of the value in the scalar fields, thus representing a relatively less important feature in the data. This is interesting because the set of values with high predictability and low surprise for this experiment corresponds to the boundary fluid-density values i.e, the individual material densities of the two mixing fluids which dominates the scalar field. Regions with values close to these initial density values correspond to the locations where the materials have not yet mixed properly and is in fact not the feature that scientists are interested in studying in this experiment. Figure 6.5(b) and (c) shows the corresponding violin-plots for the user selected values in the scatter-plot. The Y-axis corresponds to the normalized predictability values of the individual members for the selected values which are plotted along the X-axis. The violin-plot conveys the contribution of individual member to the total predictability of a value. The shape of the violin glyph shows the distribution of the individual predictability values. The shape of the violins

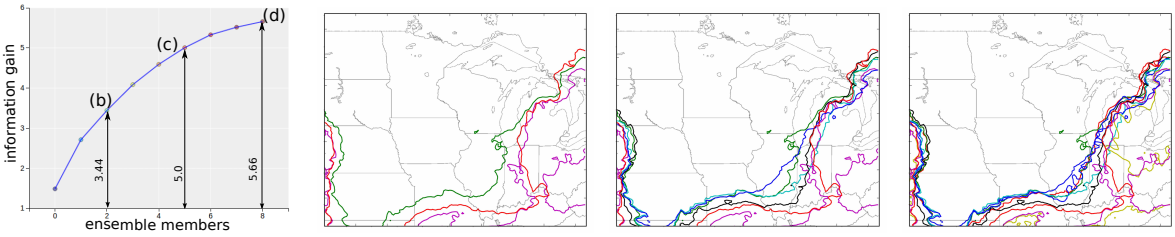
for values v_1 and v_3 in Figure 6.5(b) indicate a high agreement among the individual members, while, the narrow violin for v_2 indicates the wide variation of the members. The second view of the violin-plot as shown in Figure 6.5(c) shows the distribution of individual predictability members for both the directions of the information channel. The blue side of the violin shows the distribution of how predictable are the individual members when the corresponding isocontour in the mean field is known, while, the green side of the violin shows the distribution of predictability of the mean field when the corresponding isocontours in the individual members are known. A high variation in the distribution of the individual predictability values indicates that not all the isocontours contributed equally to the overall high uncertainty of that values. For, such scalar values we explore the importance of individual contours from an information-theoretic point-of-view.

Figure 6.6(a) shows the *information gain curve* of the 100 member ensemble isocontours for isovalues 3.21 (i.e, v_2). The total entropy/uncertainty of the system is 2.9375. As can been seen, in the corresponding information gain curve, by selecting the first 7 informative members the information gained is 2.6, which is about 88.5% information about the system, while, the information gained by selecting the first 15 informative members is 2.824 which is about 96% of the system. This says that the top 15 most informative isocontours are sufficient to represent the structural variation of the 100 ensemble isocontours. We show this claim in subsequent figures by drawing the spaghetti plots of all the 100 isocontours (Figure 6.6b), top 7 informative isocontours (Figure 6.6c) and the top 15 informative isocontours (Figure 6.6d). As shown in Figure 6.6(c) and Figure 6.6(d), the spatial spread of the isocontours can be easily viewed with lesser number of informative isocontours. This offers a quicker way of understanding the uncertainty of the ensemble isocontours without looking at all the 100 instances.



(a) (b) (c)

Figure 6.7: Scalar value exploration of Great Lakes WRF ensemble. (a) Interactive Scatter-plot view of the total predictability vs total surprise of the scalar values. (b) Violin-plot view showing the distribution of individual predictability values for selected scalar values. (c) Split violin-plot view showing the distribution of the predictability values for the two directions of the bi-directional information channel.



(a) (b) (c) (d)

Figure 6.8: Informative isocontour selection of Great Lakes WRF ensemble: (a) The information gain curve for isovalue 1440.5. (b) The spaghetti-plot of the top 3 informative isocontours which captures about 60% of the total uncertainty. (c) The spaghetti-plot of the top 6 informative isocontours which captures about 88% of the total uncertainty (d) The spaghetti-plot of all the isocontours.

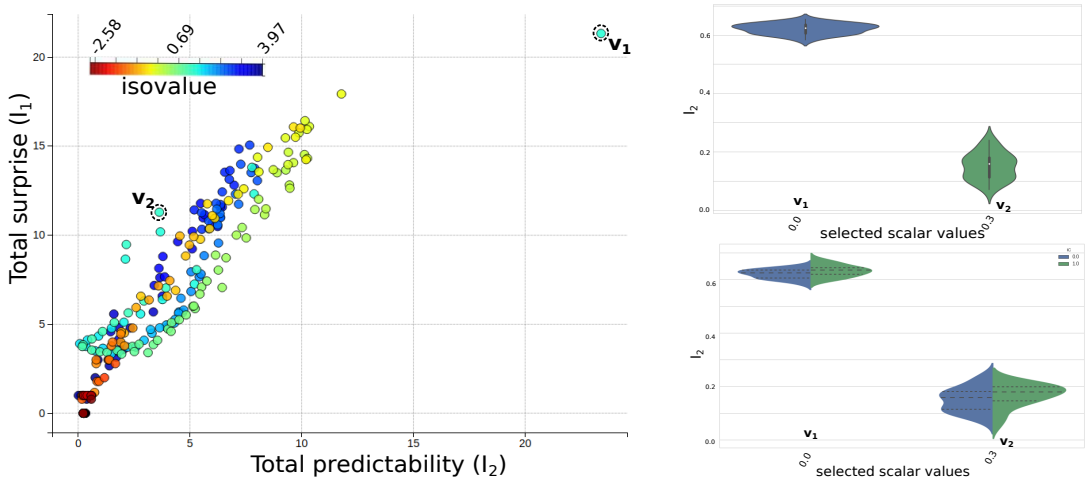
6.5.2 Great Lakes WRF Ensemble

Our second dataset is the Great Lakes WRF ensemble data, generated by the Atmospheric Sciences Program of the University of Wisconsin-Milwaukee. This is a 9-member ensemble of numerical weather forecasting across the Great Lakes region using the WRF-ARW forecasting model. The resolution of the dataset is 167×151. We used the pressure variable over the domain to perform our analysis.

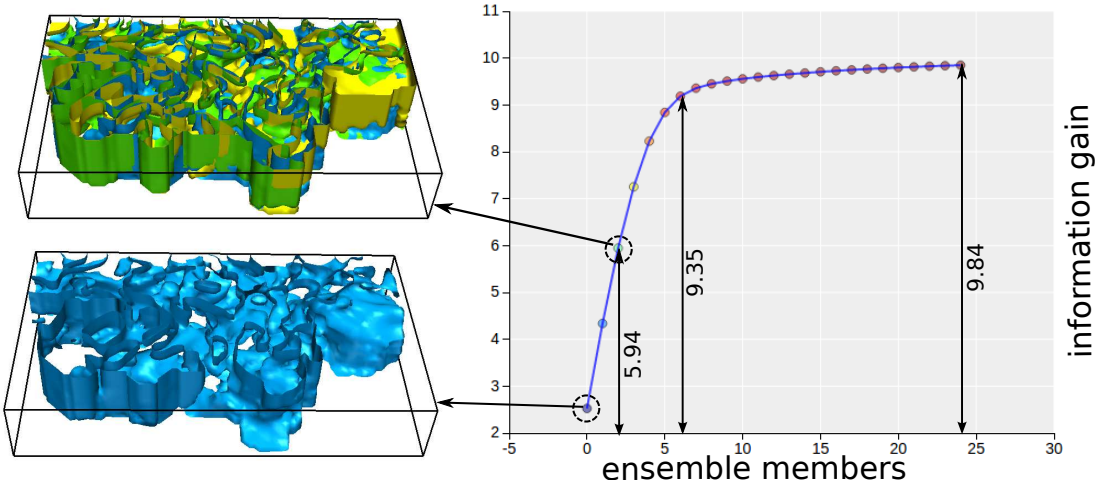
Figure 6.7(a) shows the total predictability versus surprise scatter-plot for the various pressure values. The values with high predictability and surprise corresponds to the less uncertain pressure values in the experiment. Figure 6.7(b) and (c) show the corresponding violin-plot views for the selected values in the scatter-plot, i.e, v_1(111.8Pa), v_2(419.5Pa) and v_3(1440.5Pa). These violin plots show the distribution of the predictability results of the individual simulation models for the selected pressure values. Figure 6.8(a) shows the information gain curve for isovalue 1440.5. As can be seen, the information gained about the ensemble isocontours is not significant by selecting just a few member. The total information/entropy of the system is 5.66 and the amount of information gained by selecting the top 3 informative contours is 3.44 (about 60%) while selecting the top 6 give an information gain of 5.0 (about 88%). Because of the high structural variations of the ensemble isocontours and relatively less number of ensemble members we cannot decide on a small subset of informative isocontours. As the spaghetti-plot of top 3 (Figure 6.8(b)) and top 6 (Figure 6.8(c)) shows, the complete structural information of all the 9 ensemble isocontours (Figure 6.8(d)) cannot be represented properly by a subset of isocontours. This implies that for such datasets with high uncertainty and very low number of simulation, it is important to consider all the members for analysis.

6.5.3 Massachusetts Bay Ocean Modeling Ensemble:

Our third dataset is a three-dimensional ensemble dataset, covering the region from the Massachusetts Bay to the Cape Cod area of the US east-coast [100, 101, 108]. This dataset is divided into 53×90 grid with 16 depth levels and consists of 25 ensemble simulations, generated by different sets of initial parameters and boundary conditions. The focus of this scientific experiment was to model the oceanic biodiversity of the selected region and to



(a) Interactive Scatter-plot view of the total predictability vs total surprise of the scalar values along with the violin-plot view for the selected values marked as v_1 and v_2. The violin-plot view (top right) shows the distribution of individual predictability values for v_1 and v_2. The split violin-plot view shows the distribution of the predictability values for the two directions of the bi-directional information channel.



(b) The information gain curve for isovalue 0.0 for 25 members along with the ensemble isosurfaces. The bottom left isosurface corresponds to the most informative surface. The top three informative isosurfaces are shown in top left image which comprises about 60% of the total uncertainty of all the members.

Figure 6.9: Information-theoretic exploration of ocean temperature values in the Massachusetts Bay ensemble dataset.

observe the effect of various variables on the life-forms. We use the oceanic temperature variable of the region to test our proposed technique for three-dimensional data.

Figure 6.9(a) shows the total predictability versus total surprise plot for the various temperature values along with the two violin-plot views corresponding to the selected points in the scatter-plot. For this dataset the structural variation of most of the values are very high. However, for the particular scalar value of 0.0 degree Celsius, the relative predictability and surprise is high. This indicates a low uncertainty of its corresponding ensemble isosurfaces. Figure 6.9(b) shows the information gain curve for isovalue 0.0 alongside the ensemble isosurfaces selected in the curve. The curve shows the information gained about the system by selecting an ensemble isosurface member. We show the isosurface (blue) corresponding to the maximum information gain i.e, the first point in the information gain curve. Since, it is difficult to visualize multiple isosurfaces, we restrict to visualizing the top three informative isosurfaces only; the blue surface being the most informative, the green is the second most informative isosurface and the yellow surface is the third most informative isosurface. These three isosurfaces represent about 60% of the information (structural variation) of the entire ensemble system. The inherent occlusion and clutter while drawing multiple isosurfaces make it a challenging problem to visually analysis the multiple surfaces. Therefore, the kind of analysis and exploration that we have proposed in this work helps us understand the structural variation without going through the trouble of rendering multiple surfaces.

6.6 Discussion

Our proposed specific information based scalar value exploration method is an efficient and effective method to understand the variations of the ensemble isocontours of all the scalar values in the data. To the best of our knowledge, there is no other existing work that explores the isocontour uncertainty of a range of scalar values at the same time. A partially related work is the isosurface similarity map proposed by Bruckner et al. [27]

that uses mutual information to compare the isosurfaces of all the scalar values in a single scalar field. Mutual information of the distance fields of the corresponding isosurfaces are computed for all the pairs of isovalues in the scalar field to create the similarity map. A similar approach for comparing the ensemble isocontours for all the scalar values will be exponentially expensive. However, to validate the predictability measures in our work, we compared our results with a similar brute force mutual information based approach for the 100 member material density dataset. We extracted the individual isocontours and their distance fields for all the corresponding scalar values and computed the pair-wise mutual information values across all the 100 members. Figure 6.10(a) shows the normalized average pair-wise mutual information values for all the scalar values, while Figure 6.10(b) shows the normalized total predictability (I_2) computed by our proposed method. We can see similar uncertainty trend for the range of scalar values. However, the brute force approach took about 49 minutes and our specific information based approach took about 1.4 minutes for all the scalar values across 100 ensemble members.

For a single scalar value, there are many ensemble isocontour visualization techniques [56, 144, 158, 183]. All this techniques can be used to understand the uncertainty associated with a selected isovalue. The first stage of our exploration helps the user to select scalar values for such analysis. However, current analysis methods do not offer any insights into how the individual member isocontours of a scalar value are contributing to the overall uncertainty. This is addressed in the second stage of our exploration, where we use conditional entropy to determine the informativeness of the individual member isocontours. To the best of our knowledge, there is no existing work in ensemble visualization literature that quantifies the uncertainty contribution of individual members. The informativeness results help us create samples/subsets of members that can reliably represent the uncertainty

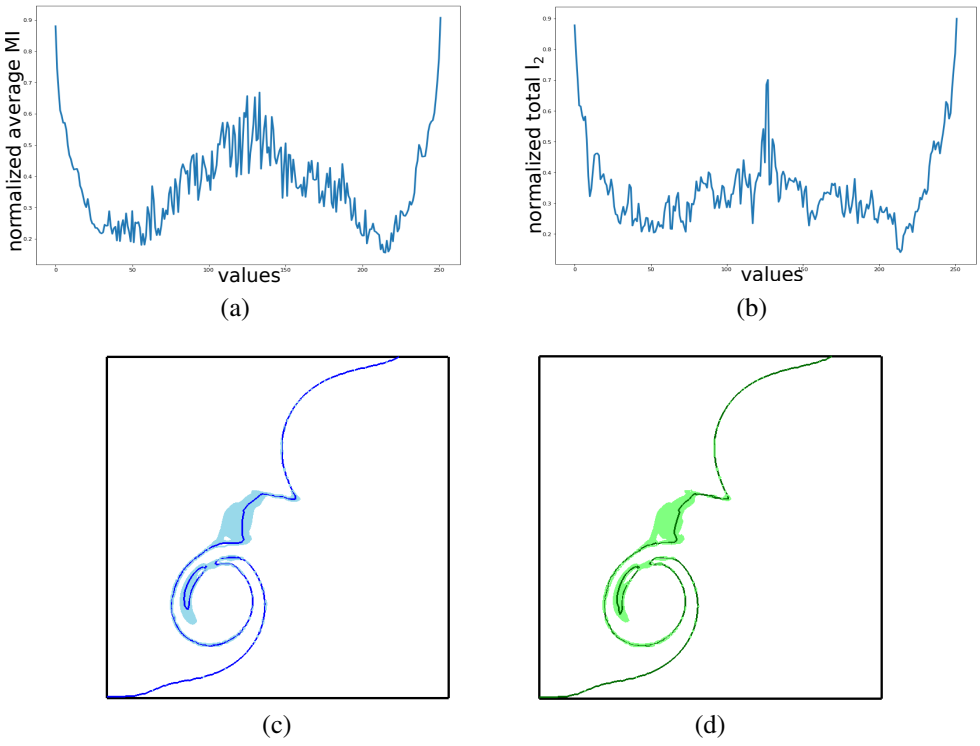


Figure 6.10: Validation: (a) The average pair-wise mutual information of ensemble isocontours for all the scalar values. (b) The total predictability results generated by our proposed method for all the scalar values. Both (a) and (b) reveals similar trend of uncertainty across the value range. (c) Contour variability band of all the 100 ensemble isocontours and (d) top 7 informative isocontours. Both (c) and (d) display similar variability band structure and average contour shape.

of the complete set of ensemble isocontours. This is especially useful for systems with large number of members. Apart from the visual validations shown via the spaghetti-plots in Section 5, we tried to validate whether the subsets we chose were indeed representative of the original ensemble. To do so, we visualized the distribution of isocontours by drawing bands using the algorithm proposed by Ferstl et al. [56]. Figure 6.10(c) shows the band and the geometric median created by the original 100 ensemble isocontours of the material density dataset, while, Figure 6.10(d) shows the band and the geometric median created by the top 7 informative isocontours selected by our algorithm. The band of top 7 contours

covers 91% of the original band-area. This shows that our approach selects samples which are good representation of the structural properties of the original ensemble.

In Table 6.1, we show the computation times of the major steps. The second column of the table shows the data resolution and corresponding number of ensemble member in parenthesis. The third column shows the computational time for the calculation of the I_1 and I_2 values across all the information channels. The fourth column shows the time for conditional entropy based informative isocontour computation for a single scalar value.

Datasets	Dim (ensembles)	I_1 and I_2 (secs)	Cond. Entropy (secs)
Material	$128 \times 128(100)$	84.3	47.3
Great Lake	$167 \times 151(9)$	1.2	20.5
Mass. Bay	$16 \times 53 \times 90(25)$	5.1	33.2

Table 6.1: Performance of various computational stages.

Though, these computations are one-time activities for a given ensemble dataset, we feel that there is room for improvement with respect to performance. Parallelization of the entire pipeline can help us in achieving better performances.

6.7 Conclusion

In this chapter, we presented an information-theoretic approach of exploring the uncertainties across a scalar value range for ensemble datasets. Further, for a single scalar value, we let the users explore the structural variations of the ensemble isocontours. Using specific information measures, I_1 and I_2, we proposed a method to understand the effect of uncertainty on the ensemble isocontours of all the scalar values. For a selected scalar

value, we let the users explore the structural variations of the ensemble isocontours using a conditional entropy based method. This exploration leads to the identification of structurally informative isocontours that can represent the spatial properties of the complete ensemble system. In future, we plan to extend this method to time-varying and multivariate ensemble datasets. The current method is targeted only for ensemble scalar datasets, however similar specific information measures can also be applied for vector datasets to understand the uncertainty of various features like streamlines and stream-surfaces.

Chapter 7: Neural Network Assisted Visual Analysis of Yeast Cell Polarization Simulation

7.1 Introduction

In the field of computational biology, scientists often design mathematical simulation models to offer quantitative descriptions of complex biological processes. These simulations are subsequently used to perform in-depth analyses of the real biological phenomenon. However, designing an optimal simulation model can be challenging. Scientists need to have a clear picture of how the different simulation input parameters are affecting the simulation output. For compute-intensive simulation models with high-dimensional input and output spaces, this can become a computationally prohibitive and non-trivial analysis task.

We collaborated with computational biologists to design an interactive visual analysis framework, which can assist them in analyzing and visualizing a complex *yeast cell polarization* simulation model. The model simulates the concentration of important protein molecules along the membrane of a yeast cell (single-cell microorganism) during its mating process. Cell polarization refers to asymmetric localization of protein concentration in a small region of the cell membrane and is a fundamental stage in the life-cycle of many microorganisms. Our experts are interested in exploring and analyzing the simulation input parameters, which can simulate varying levels of cell polarization results, particularly, the

ones with high polarization. However, there are 35 different unknown/uncalibrated input parameters for the simulation. Besides the high-dimensional nature of the problem, the simulation model itself is computationally expensive. It takes hours on a supercomputing cluster to complete a single execution of the model. This seriously hampers the possibility of performing any exploratory analysis task that requires frequent execution of the simulation on new and unseen parameter configurations to study its properties in detail. In the field of simulation sciences, a popular and effective strategy to address this issue has been to create a simpler statistical/mathematical *surrogate model*, mimicking the original expensive simulation model [58, 66, 84, 149, 151]. The surrogate is then utilized to perform detailed analysis tasks instead of the expensive simulation model. A well-trained surrogate model can greatly facilitate the analysis workflow of complex simulation models.

Compared to popular surrogate model options like *polynomial fitting* or *Gaussian Processes*, *neural networks* are particularly well-suited for designing interactive visual analysis systems. This is primarily because, besides accurately predicting the output of high-dimensional non-linear functions, they can also be utilized to extract and analyze interesting properties about the original simulation by opening up the *black-box* of the trained neural networks. Recent advances in the field of *interpretability* and *explainability* of neural network-based models [122] have resulted in many useful post-hoc analysis techniques, making them more transparent in the process. This has led to a surge in their usage as proper analysis tools in many application domains [10, 91, 162, 170]. In this work, we propose a *neural network*-assisted *visual analysis* system (NNVA), which utilizes a *neural network-based surrogate model* to perform exploratory analysis and visualization of the aforementioned yeast simulation model. The surrogate model acts as the backend analysis framework, facilitating various visual interactions and analysis activities in the system.

Of late, there is a growing interest in the usage of different neural network-based models to solve complex real-world problems. Our work, therefore, exemplifies how to build an interactive visual analysis system around these powerful models, i.e, *Machine Learning for Visual Analytics (ML4VA)*.

Our proposed system facilitates interactive exploratory analysis by allowing the experts to modify the input parameter values and immediately visualize the predicted simulation outcome. This helps them discover new parameter configurations without having the need to execute the original expensive simulation for every instance. Using different interactive selection brushes, experts can perform parameter sensitivity analysis at multiple levels of detail as well as get optimal parameter recommendations to produce desired simulation outcomes for selected regions of the membrane. To establish the trustworthiness of any visual analysis system, it is important to convey the underlying uncertainty associated with the predictions of the surrogate model. We utilize a recently proposed uncertainty quantification technique for neural networks using dropout layers [61] to incorporate uncertainty visualization in our system. We also allow the experts to validate the surrogate model itself, by analyzing the various weight matrices and extracting the knowledge learned by the surrogate during the training process. We performed extensive evaluations of the proposed framework by comparing with the original simulation model and the results of a previous analysis effort which used polynomial surrogate models [152].

To summarize, the major contributions of our work are as follows:

- We demonstrate how a trained neural network can act as an analysis backend to drive an interactive visual analysis system.
- We discovered multiple previously unknown parameter configurations, which generated strong cell polarization results in the original simulation model.

- We provide easy integration of our visual analysis workflow with the simulation modeling workflow of the experts by allowing them to store the discovered configurations in a file format, which can be used to directly execute the original simulation model.

7.2 Simulation Model Background

Traditional laboratory-based approach for studying yeast cell polarization consists of a laborious workflow. The cells are first cultured/grown in a growth media and then treated with various straining agents. They are then visualized using high-resolution microscopes. Fig. 7.1(a) shows the microscopic image of a highly polarized yeast cell. A mathematical simulation model, therefore, can significantly accelerate the study of such biological phenomena.

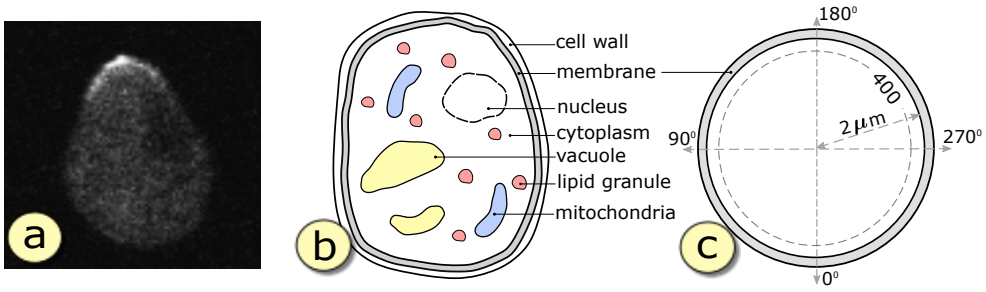


Figure 7.1: (a) Microscopic image of a highly polarization yeast cell. (b) Pedagogical illustration of the yeast cell structure. (c) The computational domain used in the simulation to model the cell membrane.

To capture the spatio-temporal dynamics of yeast cell polarization during the mating cycle, scientists created a mechanistic spatial model to simulate the concentration of important protein species along the cell membrane. In this work, we focus only on one important protein species called *Cdc42*. Protein concentration is measured as the number of molecules

per unit area of the membrane. The model simulates the cell membrane as a circle, centered at the origin with radius $2\mu m$. Fig. 7.1(b) shows a pedagogical diagram of the yeast cell structure with the peripheral cell membrane, while, Fig. 7.1(c) shows the corresponding computational domain of the simulation model used by the scientists. The computational domain (circle) has a spatial resolution of 400, parameterized by angles in the range $[0360$. Scientists are interested in simulating results with high degree of polarization, in particular of *Cdc42* protein. To quantify the extent of Cdc42 polarization, they constructed a scalar function of active Cdc42 (C42a) values called *polarization factor* (PF), denoted as;

$$PF = \left(1 - 2\frac{S_p(C42a)}{SA}\right) \times \frac{(ax)^5}{1+(ax)^5} \tag{7.1}$$

where, SA is the surface area of the membrane simulated by the model, $S_p(C42a)$ is the surface area at the front of the cell that encompasses half of the polarized component C42a, x is the maximum C42a concentration value and a is an experiment constant dependent on simulation parameter. An unpolarized cell would have a PF value of 0 and an infinitely polarized cell would have a PF value of 1.

However, the simulation model comprises of 35 different input parameters. For the model to be useful to study the biological process of yeast cell polarization, scientists need to have a clear understanding of how the simulation input parameters effect the simulation results. More specifically, they want to figure out the parameter configurations which can generate high cell polarization results in the model. Studying this high-dimensional parameter space is not trivial, especially when the individual simulation execution itself takes few hours to execute. Any analysis task that requires frequent execution of the model on new parameter configurations is effected by the long execution time of the simulation model.

7.2.1 Previous Simulation Model Analysis

Previous efforts into analyzing this simulation model involved creating a polynomial surrogate model [152]. The surrogate model was created by uniformly sampling the parameter space and fitting a polynomial function to the polarization factor (PF) values (Equation 7.1). The surrogate model facilitated in analyzing the parameter sensitivity of the model and helped estimate parameter configurations using a Markov Chain Monte-Carlo (MCMC) approach. However, the approach was not able to identify satisfactory parameter configurations with which the simulation can generate significantly high polarization results. Moreover, given a parameter configuration, the polynomial surrogate model only predicts the final PF value and not the Cdc42 protein concentration values across the membrane, which is the final output of the simulation. As a result, parameter sensitivity analysis performed with the polynomial surrogate model was not able to study the influence of the parameters on different regions of the membrane in finer details. Such analysis is important to get a better understanding of how the simulation parameters actually affect the protein concentration across the membrane and not just the final PF value. In this work, we train a neural network-based surrogate model to predict Cdc42 concentration values for the 400 spatial locations across the membrane as modeled by the computational domain (Fig. 7.1(c)). We use this neural network-based surrogate model to create a visual analysis system for the simulation model. We utilize some of the important findings in the previous work [152] to extensively evaluate and validate the results produced by our proposed system.

7.3 Requirement Analysis and Approach Overview

7.3.1 Requirements

Throughout the course of this project, we had multiple interactions with the scientists from computational biology to understand the various aspects of their simulation model and get a clear picture of their needs and requirements from a visual analysis system. Based on these discussions, the most important requirements are as follows:

R1 Discover new parameter configurations which can generate high Cdc42 polarization results in the simulation model. The system should have the ability to visually guide the users in the process of finding desired parameter configurations for the simulation.

R2 Ability to get a quick preview of the predicted simulation output for particular parameter configuration to facilitate model calibration. This helps them decide whether to execute the expensive simulation model with certain parameter configurations or not.

R3 Perform detailed sensitivity analysis of the input parameters with respect to Cdc42 concentration for different regions of the cell membrane.

R4 Analyze the distribution of protein concentration values across the computational domain of the model. This is required to decide on an ideal partitioning scheme for the computational domain.

R5 Ability to extract and validate the knowledge learned by the surrogate during its training process. This is required to make sure that the trained network is not making any random predictions.

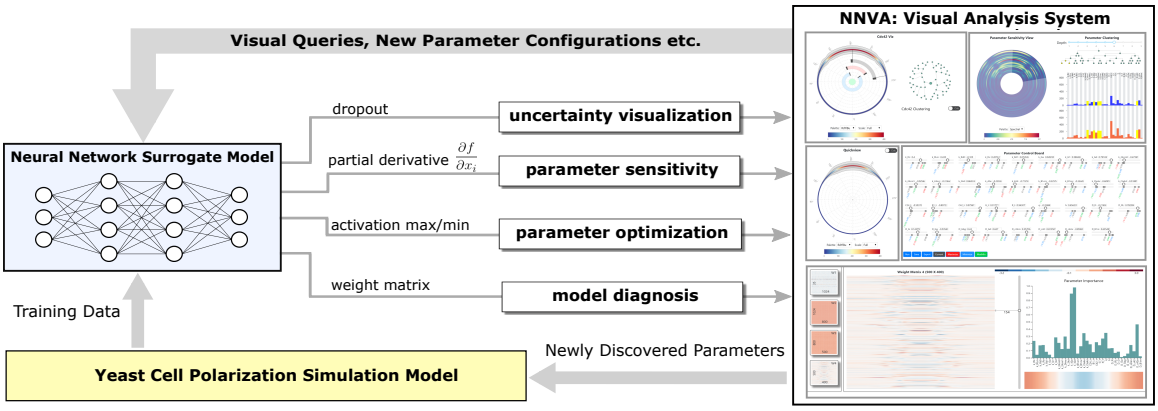


Figure 7.2: Approach Overview: A trained neural network-based surrogate model acts as the backend analysis framework, driving our interactive visual analysis system for analyzing a computationally expensive yeast simulation model.

7.3.2 Overview

Based on these requirements, we have proposed an interactive visual analysis system, which is driven by a neural network-based surrogate model. We first train a fully connected neural network on a finite set of training data, obtained by running the simulation on random parameter configurations. The neural network learns to predict the concentration of the Cdc42 protein along the cell membrane for a given input parameter configurations. We then design a visual analysis framework which allows the users to visually query for different properties about the simulation model to address the aforementioned user requirements. These queries are executed in the backend by the trained surrogate model to provide prompt feedback via the visual interface. The system visually guides the users to discover new parameter configurations, which can be later used to execute the original simulation model. Fig. 7.2 shows the high-level overview of the proposed system.

7.4 Neural Network-based Surrogate Model

Surrogate models are widely used in many areas of engineering and simulation science as cost effective alternatives to expensive simulation models for various analyses [58, 66, 84, 149, 151]. Also known as response surface models or emulators, they mimic the behavior of the actual simulation model as closely as possible while being computationally easier to evaluate. The surrogate model proposed in our work is a *multi-layer fully-connected feed-forward regression neural network*. In this section, we first discuss in details the network structure and the training process of our surrogate model and then elaborate on the various post-hoc analysis techniques that can be performed on trained networks to facilitate the eventual visual analysis system.

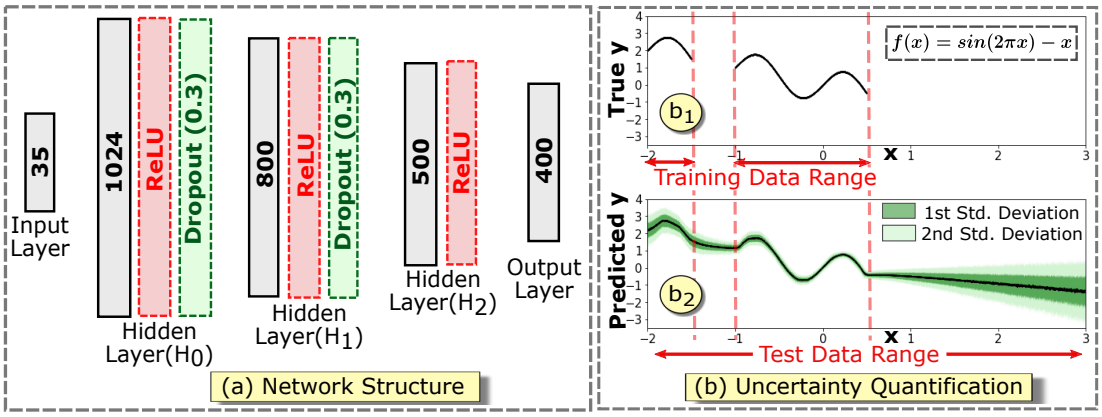


Figure 7.3: (a) Architecture of our surrogate model. (b) Dropout-based uncertainty visualization of neural networks for a synthetic dataset.

7.4.1 Network Structure and Training Process

As shown in Fig. 7.3(a), our surrogate model consists of 5 layers, comprising an input layer of 35 neurons and an output layer of 400 neurons, corresponding to the 35 simulation

input parameters and 400 uniformly distributed spatial locations along the cell membrane respectively. The intermediate hidden layers, H_0, H_1 and H_2 are composed of 1024, 800 and 500 neurons respectively. The network is fully-connected, therefore, the neurons of one layer are connected with all the neurons of its subsequent layer. The output of each neuron in the three hidden layers are passed through ReLU (rectified linear unit) activation functions [125] to model any non-linearity between the input parameters and the output responses. We also apply *dropout* regularization to the first (H_0) and second (H_1) hidden layers with a dropout rate of 0.3. Dropout regularization corresponds to randomly ignoring the activation results of the neurons in a layer during the training process to avoid over-fitted networks and achieve higher accuracy. A dropout rate of 0.3 refers to the fact that we randomly ignore 30% of the neuron outputs at layers H_0 and H_1.

We trained the surrogate model on a training dataset of size 3000. Our experts predetermined the ranges of the individual parameters and normalized them independently to the range $[-1, 1]$. The training data was created by first randomly sampling the 35 parameters in their normalized value ranges to create 3000 random parameters configurations and then running the simulation model to get the corresponding Cdc42 concentration values for each configuration ($\sim$ 27 hours on a supercomputing cluster). It was observed that a large fraction of the randomly sampled training data corresponds to instances with very low Cdc42 polarization. Therefore, to train the network to predict the high polarization instances (**R1**), we performed PF value weighted training of the neural network. During the training iterations, this assigned high weightage to the loss function values corresponding to training data instances with high PF values. After training for 5000 epochs, with a batch size of 32, the model achieved a stable RMSE (root mean square error) accuracy of 87.6%. Standard mean squared error (MSE) was used as the loss function for training the network and the

accuracy was tested on a separate validation dataset of size 500. We tested different network architectures before finalizing on the one described in Fig. 7.3(a) because it had the highest accuracy and stabilized relatively faster (i.e, around 3500^{th} epoch).

7.4.2 Uncertainty Quantification in Neural Network

Traditionally, neural networks do not provide any measure of the uncertainty associated with its predictions by default. However, in a recent work, Gal et al. [61] showed that traditional neural networks can also be made to quantify uncertainty by activating the dropout layers in the prediction (testing) phase. As discussed in Section 5.1 above, dropout layers are generally used as regularizers in the training phase to avoid over-fitting of training data by randomly ignoring the neuron activations at different layers of the network. Dropout is generally turned off when the network makes predictions. Gal et al. [61] showed that if we apply dropout during prediction and randomly ignore the activations of neurons in different layers, we get slightly varying predictions every time the network is run with the same set of input. By observing the variations of the predicted results for multiple instances, we can quantify the uncertainty associated with the predicted results. Gal et al. [61] further proved that dropout induced uncertainty for neural networks is actually an approximation of the uncertainty obtained in Bayesian models like Gaussian Processes. Therefore, using this feature in our trained neural network-based surrogate model, we can incorporate uncertainty visualization into our proposed visual analysis system.

Fig. 7.3(b) demonstrates the uncertainty visualization results using dropout layers in a simple 3-layer neural network on a synthetic dataset. Consider using a neural network to learn a simple sinusoidal function $f(x) = sin(2\pi x) - x$. Training data to learn this function was provided only for the x ranges of $[-2.0, -1.5]$ and $[-1.0, 0.5]$. Fig. 7.3(b_1) shows the

plot of the true function values $y = f(x)$ for the training data ranges. Fig. 7.3(b_2) shows the result of the trained neural network predictions for x values ranging from -2.0 to 3.0. The variation in the predicted result is captured using dropout layers and is visualized as standard deviation bands (in shades of green) around the mean predicted values. As can be seen, the uncertainty is high for regions where the training data was not provided to the network to learn from. The uncertainty visualization clearly shows that as we move away from the training data range (i.e, $x > 0.5$) the corresponding prediction uncertainty also increases. In order to avoid misleading the users in their decision making process and to add a sense of trustworthiness, it is vital for a visual analysis system to convey any underlying uncertainty in the model.

7.4.3 Parameter Sensitivity Analysis

Sensitivity analysis is a popular post-hoc analysis technique performed on trained neural network models to identify the most important/salient input features. It serves as an effective tool in driving many recent advances in the field of *explainable* machine learning [122].

Sensitivity analysis of a neural network corresponds to computing the partial derivative of the outputs with respect to the inputs. Consider the i-th neuron in the output layer, predicting the function $f_i(\mathbf{x})$ for an n-dimensional input vector $\mathbf{x} \sim \{x_1, ..., x_n\} \in \mathbb{R}^n$. The sensitivity of f_i with respect to the j-th input parameter can be denoted as $\left(\frac{\partial f_i}{\partial x_j}\right)^2$. A high sensitivity value corresponds to the fact that a small change in the value of the input x_j is going to have a significant change in the output value of $f_i(.)$. The architecture of neural network is such that the output of every neuron in the network is completely differentiable with respect to its inputs, as a result, we can easily compute the required partial derivatives for sensitivity analysis via chain-rule using the backpropagation technique [76]. In our work,

we utilize this to evaluate the sensitivity of the 35 simulation input parameters using the trained surrogate model. The visual analysis system provides an interface for the scientists to query for such parameter sensitivity information for different spatial regions of interest along the cell membrane. Another advantage of using neural network for sensitivity analysis is that we can also compute the sensitivity of the hidden layer activation values with respect to the input parameters as well. We utilize this to observe the parameter sensitivity towards interesting latent space data patterns learned by the hidden layers of the surrogate model during the training process.

7.4.4 Parameter Optimization

Another important category of post-hoc analysis operation performed on trained neural networks is called *activation maximization* (AM). Activation maximization corresponds to searching for an optimal input configuration in the high-dimensional input space that maximizes the output response function. It is often used to interpret a high-level concept learned by the neural network, for example, in the field of image classification, it can be used to create new images of what the trained network thinks a cat or a dog looks like [122, 127].

Neural networks are essentially optimization machines that try to find the optimal network configurations (various weight and bias values) during the training process that can best map a given input to the desired output. Once the training process is over, the network configuration is fixed and is used to predict outputs for new and unseen input configurations. Activation maximization corresponds to a reverse optimization process, where, keeping the network configuration fixed, we search for optimal input configurations that maximizes the function values of specific neurons in the output layer. For the i-th neuron in the output layer predicting the function $f_i(\mathbf{x})$ for an n-dimensional input vector $\mathbf{x} \in \mathbb{R}^n$, we can find an

optimal input configuration $\mathbf{x}^*$ by optimizing the following objective function

$$\max_{\mathbf{x}} f_i(\mathbf{x}) - \lambda ||\mathbf{x} - \mathbf{x}'||^2 \tag{7.2}$$

where, the rightmost term is an ℓ_2-norm regularizer to constrain the input search space within a known confinement $\mathbf{x}'$. This penalizes the optimizer from finding an arbitrarily different $\mathbf{x}^*$. The optimization involves a gradient ascent algorithm using the gradients $\frac{\partial f_i}{\partial x}$ to update the inputs to eventually find the optimal input configuration. A similar approach can also be employed to minimize the activation of a selected neuron, i.e, activation minimization, by negating the gradient values during the optimization steps. In our work, we utilize activation maximization and minimization principles to recommend simulations input parameter configurations to the scientists. They can visually selected the spatial regions in the cell membrane that they want to maximize/minimize the Cdc42 protein concentration for. By carefully choosing to maximize Cdc42 concentration in certain regions and minimize in the other regions scientists can query for parameter configurations that is likely to produce high polarization profiles in the original simulation.

7.5 Neural Network Assisted Visual Analysis

In this section, we introduce our proposed interactive visual analysis system. We first explain the primary visualizations and interaction techniques before looking at the high-level analysis views in our system.

7.5.1 Primary Visualizations and Interactions

***Cdc42 Visualization*:** To visualize the predicted Cdc42 concentration and preserve the circular context of the cell membrane structure, we opted for radial layout designs [44]. Fig. 7.4(a) visualizes the mean predicted concentration values for the 400 uniformly sampled

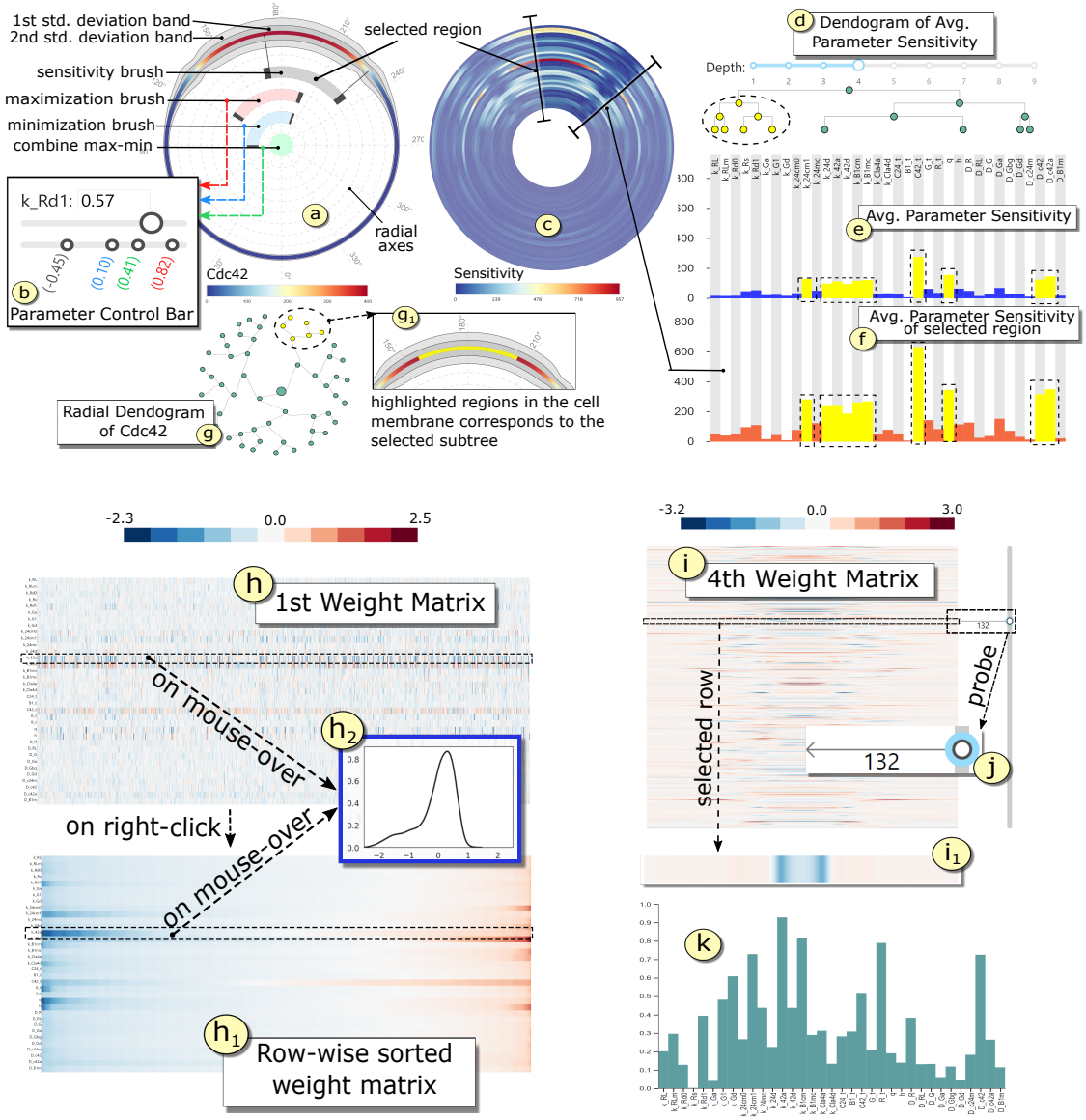


Figure 7.4: Primary Visualizations and Interaction techniques: (a) Predicted Cdc42 concentration across the membrane along with uncertainty bands and selection brushes. (b) Parameter control bar. (c) Spatial parameter sensitivity. (d) Linear cluster tree for average parameter sensitivity. (e,f) Average parameter sensitivities. (g) Radial cluster tree for predicted Cdc42. (h) First weight matrix. (i) Final weight matrix. (j) Row selection probe. (k) Average parameter sensitivity for selected pattern.

points across the membrane. By default, the values are color-mapped to the maximum and minimum Cdc42 values observed in the simulation by the experts. Radial coordinate

axes (dashed-gray lines) are provided in the backdrop as frame of reference to reflect the parameterization of the simulation domain in terms of angles (degree). To efficiently utilize the design space, we employed the popular design principle of *superposition*, i.e. visualizing multiple data subsets in the same coordinate system [123].

***Uncertainty Visualization*:** The uncertainty associated with the predicted values of the surrogate model is visualized using superimposed standard deviation bands around the circumference of the circular domain. The shapes of first (inner) and second (outer) standard deviation bands, as shown in Fig 7.4(a), highlight the deviation of the predicted values in the corresponding locations of the membrane.

***Selection Brushes and Interaction*:** As shown in Fig. 7.4(a), we superimpose multiple interactive radial selection brushes in the same coordinate system as the Cdc42 visualization. This facilitates performing various visual queries on different regions of the membrane. The *sensitivity brush* (gray) allows the users to select the regions of the membrane where they want to perform detailed parameter sensitivity analysis. This brush is linked with the sensitivity visualizations in Fig. 7.4(c) and (f). The *maximization brush* (red), selects the region where the users want to maximize the predicted concentration values (Section 7.4.4). On clicking this brush, the corresponding optimal parameter values, computed by the backend surrogate model, are reported in the parameter control bars (Fig. 7.4(b)). Similarly, the *minimization brush* (blue) selects the region to minimize the predicted values. The green circular button at the center performs a logical AND operation of the regions selected by the maximization and minimization brushes.

***Parameter Control Bars*:** Fig. 7.4(b) shows the parameter control bar for one of the simulation input parameters. The parameter name is followed by an input textbox to enter desired parameter values. The sliderbar, immediately below, can also be used to adjust the

parameter values. The parameter values corresponding to different configuration instances show up in the last bar. The value corresponding to currently loaded instance show up in black colored text, whereas, the optimal parameter values recommended by activation maximization, minimization and combined max-min show up as red, blue, and green colored texts respectively. Users can click on these texts to adjust the parameter value as well.

***Sensitivity Visualizations*:** For a given parameter configuration, the local sensitivity of the 400 spatial locations across the membrane with respect to the 35 parameters are visualized in a circular heatmap as shown in Fig. 7.4(c). The 35 parameters are laid out along the radial direction. A high sensitivity score for a parameter implies that a small change in its current value is going to trigger a relatively high change in the predicted Cdc42 value for the specific region of the membrane. Finer spatial selections can be made using the aforementioned *sensitivity brush*. The average sensitivity of the 35 different parameters across all the spatial locations is shown as a bar-chart in Fig. 7.4(e). The average parameter sensitivity information for a user selected region of interest (via sensitivity brush) is shown in Fig. 7.4(f), juxtaposed with (e) to convey the relative variation in the sensitivities.

***Cluster Visualization*:** To help analyze the current partitioning scheme of the simulation domain (**R4**), we perform hierarchical clustering [96] of the 400 uniformly partitioned points based on their Cdc42 values and associated uncertainty. The multi-level cluster information is visualized using a radial dendogram as shown in Fig. 7.4(g). Hovering over the nodes highlights the corresponding selected clusters in the simulation domain (Fig. 7.4(g_1)). Similar clustering is performed on the average sensitivities of the 35 parameter to study their associations. Fig. 7.4(d) shows a linear dendogram view of the parameter cluster information. Users can control the tree depth, which relates to the number of clusters desired for the

study. For selected nodes in the tree, the corresponding cluster members get highlighted in Fig. 7.4(e) and (f).

***Weight Matrix Visualization*:** To extract the knowledge learned by the trained neural network, and thereby, validate the surrogate model (**R5**), we present various techniques to analyze its weight matrices. Fig. 7.4(h) shows the weight matrix (35×1024) between the input layer and the first hidden layer (H_0). The rows correspond to the 35 input parameters and the columns correspond to the final weights assigned to the 1024 neurons in H_0 layer. On clicking the matrix, the weights are sorted in ascending order for each row of the matrix, which helps identify interesting weight distribution patterns as shown in Fig. 7.4(h_1). Detailed explanation about the interesting weight patterns and their importance is provided in Section 7.6.2. On hovering the mouse over the rows, we display the shape of the corresponding weight distribution for the respective parameter in a pop-up window (Fig. 7.4(h_2)). Based on the patterns observed in the final weight matrix (500×400) between the last hidden layer (H_2) and the output layer, as shown in Fig. 7.4(i), we offer a separate set of interactions to study the matrix. We provide a row selection probe in the form of a sliderbar with an arrowhead (Fig. 7.4(j)) to select the rows with interesting weight patterns. The selected row index is highlighted in the side and a zoomed-in view of the row is displayed (Fig. 7.4(i_1)). Fig. 7.4(k) shows the average parameter sensitivity chart corresponding to the neuron in the penultimate layer (H_2), which is responsible for the selected weight pattern.

***Colormap Adjustments*:** We offer two sets of value ranges for mappings the colors when visualizing the predicted Cdc42 concentration. The first set (default) is based on the minimum and maximum concentration values that the experts feel is effective for studying cell polarization behavior. For the current system, this range is set to $[0, 400]$. The

second set corresponds to the local minimum and maximum values for individual prediction results. This provides more control to the experts during the analysis period, because the concentration values for different instances can have varying dynamic ranges. We have included nine divergent and three sequential colormaps [7] in our system for the users to choose from.

7.5.2 Visual Analysis System

Using the visualizations and interaction techniques explained above, we design our visual analysis system with multiple high-level views to provide a structured and efficient analysis workflow for our experts. It comprises of the following high-level views, each addressing different facets of the analysis requirements set forth in Section 7.3.

***Instance View*:** This view loads the visualizations corresponding to a specific *instance* of input parameter configuration that the scientists wish to analyze in detail (**R3**, **R4**). As shown in Fig. 7.5(a), it comprises of two sub-views. *Cdc42 Viz* displays the predicted Cdc42 concentration result for the currently analyzed parameter instance, whereas, *Parameter Sensitivity View* displays the corresponding spatial sensitivity as well as the overall average parameter sensitivity for the specific instance. Both the sub-views have the corresponding cluster trees described in Section 7.5.1 for detail analysis. In *Cdc42 Viz*, there is a switch to toggle between the radial cluster trees corresponding to the Cdc42 values and the uncertainty values (**R4**).

***Parameter Control Board*:** This view serves as the main panel to visualize and interactively modify the input parameter configurations (**R1**). As shown in Fig. 7.5(c), the 35 different *parameter control bars* (Section 7.5.1) for the individual parameters are laid out across four rows. It visualizes the parameter values corresponding to the parameter instance

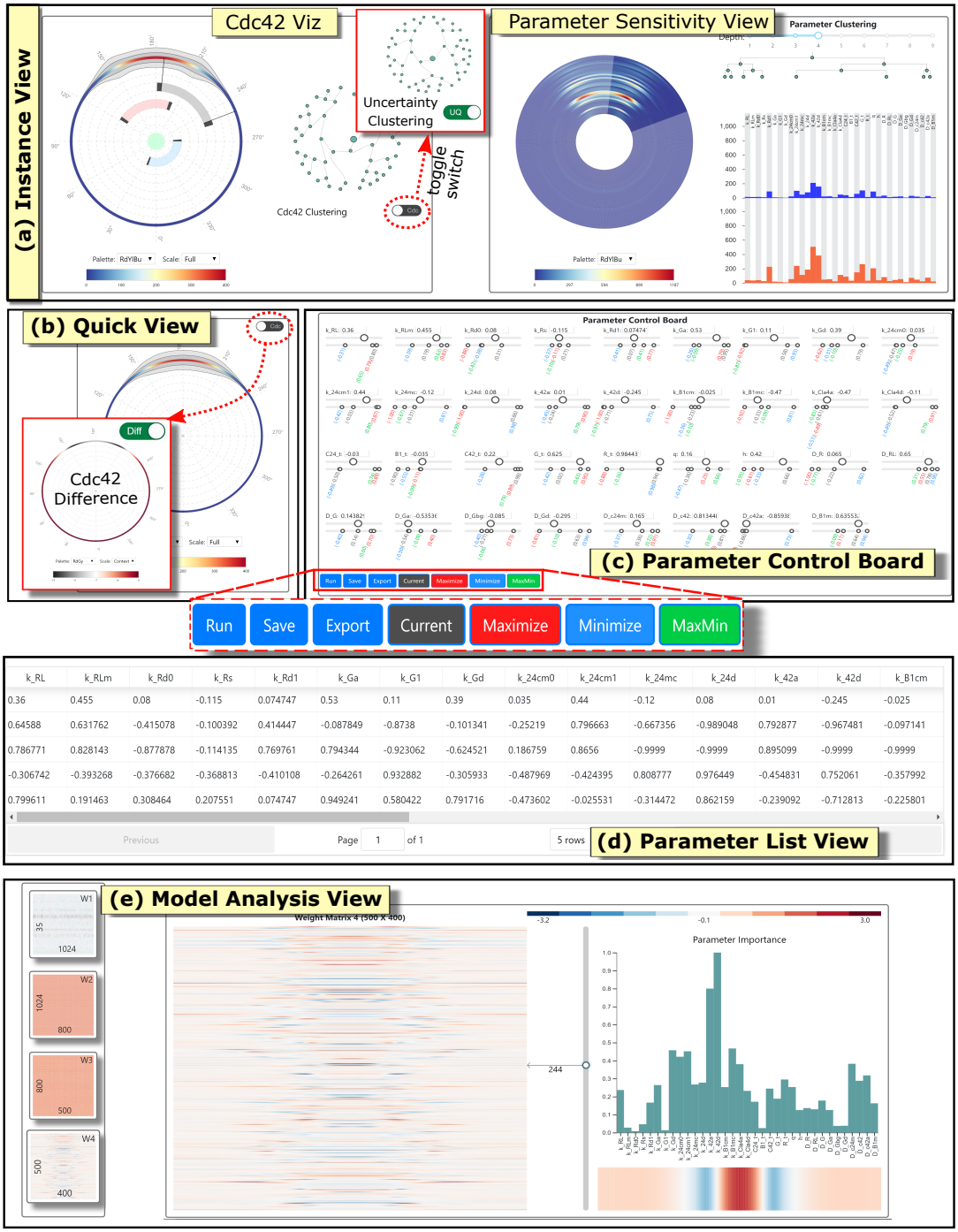


Figure 7.5: Multiple high-level analysis views of our visual analysis system.

currently analyzed in the *Instance View* as well as the optimal parameters recommended by the interactive optimization brushes. Users can modify the parameter values and click

the *Run* button to execute the neural network-based surrogate model in the backend to generate the corresponding simulation prediction, which is visualized in the *Quick View* (**R2**). The *Save* button lets us store the modified input configuration in the *Parameter List View*, whereas, the *Export* button downloads the list of saved configurations. Instead of manually adjusting the 35 different parameter sliders, users can click the *Current*, *Maximize*, *Minimize* or *MaxMin* buttons to automatically set the parameter sliders/values to the desired recommended configurations.

***Quick View*:** As shown in Fig. 7.5(b), it visualizes the predicted Cdc42 concentration for the user-modified parameter configurations via the *Parameter Control Board*. This offers a means to perform rapid prototyping of the expensive simulation using the surrogate model, thus facilitating exploratory analysis with new and unseen parameter configurations (**R1**, **R2**). In order to compare the predicted Cdc42 values vis-à-vis the results in the *Instance View*, we provide a toggle switch to change the visualization in *Quick View* to display the exact difference in Cdc42 concentration across the membrane.

***Parameter List View*:** As shown in Fig. 7.5(d), this view temporarily stores the newly discovered parameter configurations. Additionally, users can click on the rows in the list to load the selected configuration back in the *Parameter Control Board*. This list of configurations can be exported/downloaded in a file format which can be directly used to execute the original simulation model. This offers a seamless integration between the analysis workflow involving our visual analysis system and the actual simulation modeling workflow of the experts (**R1**).

***Model Analysis View*:** This view lets the users investigate the trained neural network-based surrogate model to extract useful insights about the simulation model (**R5**). As shown in Fig. 7.5(e), the left-most panel shows the thumbnail views of all the weight matrices of the

trained network. Users can click on the matrix images to open up the corresponding analysis views in the right panel. Detailed analysis, as explained in Section 7.5.1, can be performed on the selected weight matrix to extract any data patterns and validate the knowledge learned by the surrogate model.

7.6 Case Study and Evaluation

In this section, we perform two case studies using our visual analysis system and evaluate the results by comparing against the original simulation outcomes as well as the findings from a previous polynomial surrogate model based analysis of the simulation [152].

7.6.1 Discover New Parameter Configurations

One of the key requirements from the system is to visually guide the users towards discovering desired parameter configurations (**R1**), instead of having to perform random sampling of the high-dimensional parameter space. This also enables the experts to incorporate their domain knowledge into the parameter discovery process. In this case study, we use our system to identify new parameter configurations that can trigger high Cdc42 polarization in the original simulation model. We support two different visual parameter discovery approaches. According to the conceptual framework of Sedlmair et al. [163], both of these approaches can be categorized under *local-to-global* and *informed trial and error* visual parameter navigation strategies.

Guided by the Instance View: In this approach, users can center their parameter discovery process around a known instance of input parameter configuration, whose results get loaded in the *Instance View*. The *Instance View* offers in-depth analysis of the Cdc42 concentration as well as the corresponding parameter sensitivity information for the loaded parameter instance. Fig. 7.6(a), (b) and (c) show the zoomed-in views of the predicted protein

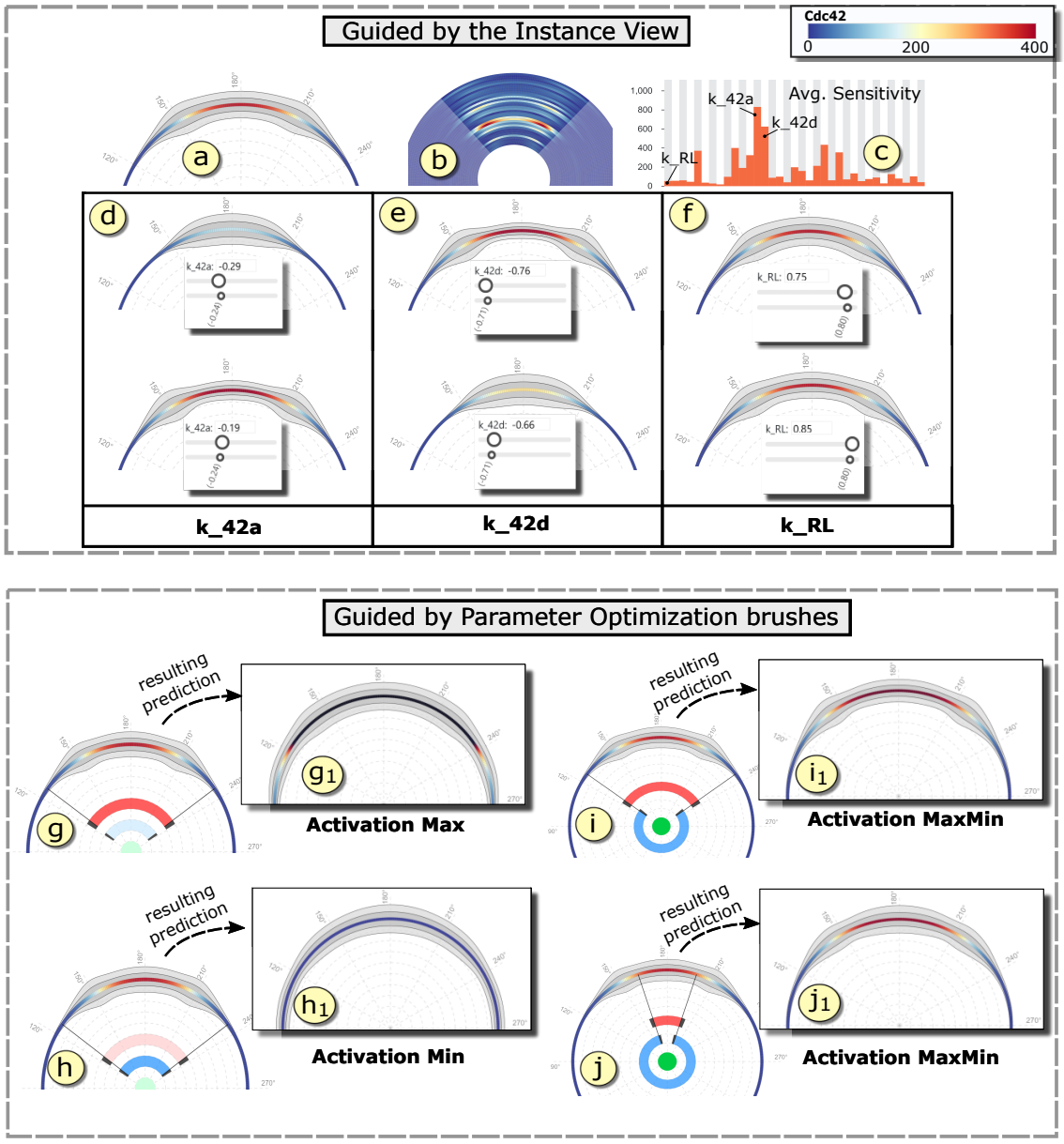


Figure 7.6: Discover new parameter configurations: (a) Predicted Cdc42 of a specific parameter instance with relatively high polarization profile. (b) Spatial parameter sensitivity of the parameter instance. (c) Corresponding average parameter sensitivities. Results for slightly changing the highly sensitive parameters k_42a(d), k_42d(e) and a less sensitive parameter k_RL(f). Maximizing (g) and minimizing (h) predicted Cdc42 values in the selected regions. (i,j) Maximizing and minimizing the predicted values for the selected regions *at the same time* to get highly polarized predictions (i_1, j_1).

concentration, spatial parameter sensitivity and average parameter sensitivity respectively of a selected region for the loaded parameter instance. As can be seen in Fig. 7.6(a), the loaded instance corresponds to a good polarization profile. Using this as the starting point, users can start modifying the parameter configurations based on the parameter sensitivity details of the loaded instance combined with their domain knowledge.

For example, from the sensitivity views in Fig. 7.6(b) and (c), we can infer that the parameters `k_42a` and `k_42d` are the most sensitive parameters for the loaded configuration. This implies that a small change to these parameter values will significantly change the current polarization profile. We use the *Quick View* panel to visualize the predicted polarization profile generated by the modified parameter values. Fig. 7.6(d) shows the results for decreasing and increasing the current `k_42a` value (-0.24) by a step-size of 0.05 in the top and bottom images respectively. Decreasing the parameter value significantly brings down the protein concentration at the top of the cell, while, increasing the parameter value increases the concentration. A reverse trend is observed for `k_42d` (Fig. 7.6(e)), where increasing the parameter values by 0.05 reduces the concentration and vice-versa.

From the simulation perspective, this behavior makes sense because `k_42a` corresponds to Cdc42 activation, whereas, `k_42d` corresponds to Cdc42 deactivation. Since the type of simulated protein is active Cdc42 (i.e, C42a), `k_42a` have a positive impact on its concentration while `k_42d` have a negative impact. Similarly, for a less sensitive parameter like `k_RL`, we can see in Fig. 7.6(f) that changing its parameter value by the same step-size does not result in a significant change in the polarization profile. Using this approach, we identified 8 new parameter configurations which are likely to produce high Cdc42 polarization results in the original simulation.

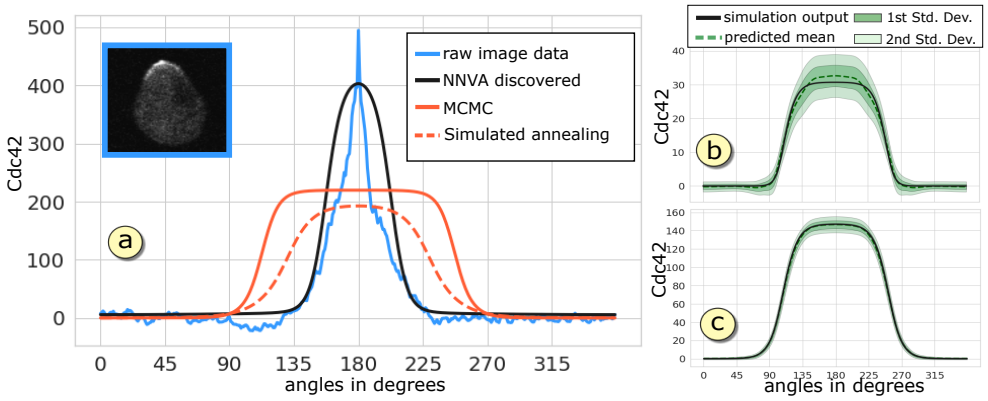


Figure 7.7: (a) Comparative evaluation of the simulation results using parameter configurations discovered by our system (black) and previous analysis work (red) [152]. Comparison curves of Cdc42 concentration for (b) a highly uncertain prediction and (c) a good prediction instance.

Guided by Parameter Optimization Brushes: We also recommend new parameter configurations based on the activation maximization/minimization analysis framework of neural network described in Section 7.4.4. Users can utilize the maximization brush, as shown in Fig. 7.6(g), to select the region of the cell membrane where they want to maximize the predicted Cdc42 value. The corresponding optimal parameter configurations computed by the surrogate is recommended in the *Parameter Control Board*. Fig. 7.6(g_1) shows the result of running the surrogate model with the recommended parameters. As can be seen in Fig. 7.6(g_1), the predicted Cdc42 values in the selected region sharply increases (even beyond the maximum value of 400, set by the experts). Similarly, a minimization brush, as shown in Fig. 7.6(h), recommends a parameter configuration that brings down the concentration values of the selected region close to the minimum concentration value (Fig. 7.6(h_1)). For the particular case of finding high polarization profiles, we are interested in maximizing the concentration at the top of the cell and minimizing the concentration across rest of the locations at the same time. We make the desired selection using the two brushes to make this query as shown in Fig. 7.6(i) and (j) for two different selection ranges.

The corresponding recommended parameters display high Cdc42 polarization predictions, as indicated by the *Quick View* results in Fig. 7.6(i_1) and (j_1) respectively for the two different selections. On top of the recommended parameters, users can further modify individual parameter values to create new sets of parameter configurations. Using this approach, we created a total of 7 new input parameter configurations.

Evaluation: We executed the simulation model using the 15 newly identified parameter configurations and observed high degrees of Cdc42 polarization (PF $>$ 0.5) for all the configurations. We found 5 parameter configurations with PF values exceeding 0.8, whereas, in the initial training data that was collected by randomly sampling the parameter space, we never found an instance with PF value close to 0.8. The highest PF value recorded among the newly discovered parameters was 0.82 and corresponds to the optimized configuration recommended by the neural network for the selection shown in Fig. 7.6(j). In Fig. 7.7(a), we compare the actual simulation result generated with our discovered optimal parameter configuration versus using the parameters estimated by previous polynomial surrogate model based analysis [152]. The sharp blue curve (PF = 0.87) corresponds to the protein concentration obtained by extracting the pixel intensity values of a real microscopic image of a highly polarized yeast cell as shown in the blue box. This acts as the ground truth for the simulation model to generate similar levels of polarization results. The black plot shows the simulation result produced using the optimal parameters discovered by our system (PF = 0.82). The solid red plot (PF = 0.57) and dashed red plot (PF = 0.64) corresponds to the simulation results generated by the parameter configurations estimated in previous work [152].

This is a significant improvement over the previous parameter analysis results for the same simulation model. This establishes that, given the right input parameter setting, the

simulation model is capable of generating sharp polarization results similar to real laboratory results. We also visually verified the predicted results of the surrogate model against the original simulation outputs by plotting them together as curves. Fig. 7.7(b) and (c) show the comparative plots for a highly uncertain prediction and a less uncertain prediction instance respectively. The green dashed-line shows the mean prediction curve, whereas, the solid black line is the original simulation output.

7.6.2 Knowledge Extraction from Surrogate Model

In this case study, we aim to extract the knowledge learned by the network during its training process by analyzing its various weight matrices. This also serves the purpose of validating the surrogate model to make sure that it is making predictions based on some reasonable domain-aligned logic rather than random ad-hoc predictions (**R5**).

First Weight Matrix: In the first weight matrix between the input layer and first hidden layer (Fig. 7.8(a)), we observe distinct patterns in the distribution of weights for certain parameters. Fig. 7.8(b) shows the full 35×1024 matrix with the weight values sorted in ascending order for each row (i.e, parameter) to highlight the patterns (Fig. 7.4(h) shows the original matrix view). We observed relatively high negative and positive weights for parameters `k_24cm0`, `k_24cm1`, `k_42a`, `k_42d`, `C42_t`, `q` and `h`. High positive and negative weights in the first matrix for a parameter corresponds to the fact that the parameter values had to be scaled by the weights for some neurons in the first hidden layers. This implies that the original range of values provided for the parameters is not sufficient. Similar observations were made for these parameters explicitly in the previous analysis work [152]. The experts feel that the range of these parameters need to be expanded to get better simulation results. Besides, the matrix also verified that the pairs (`k_24cm0`, `k_24cm1`), (`k_42a`, `k_42d`), and (`q`,

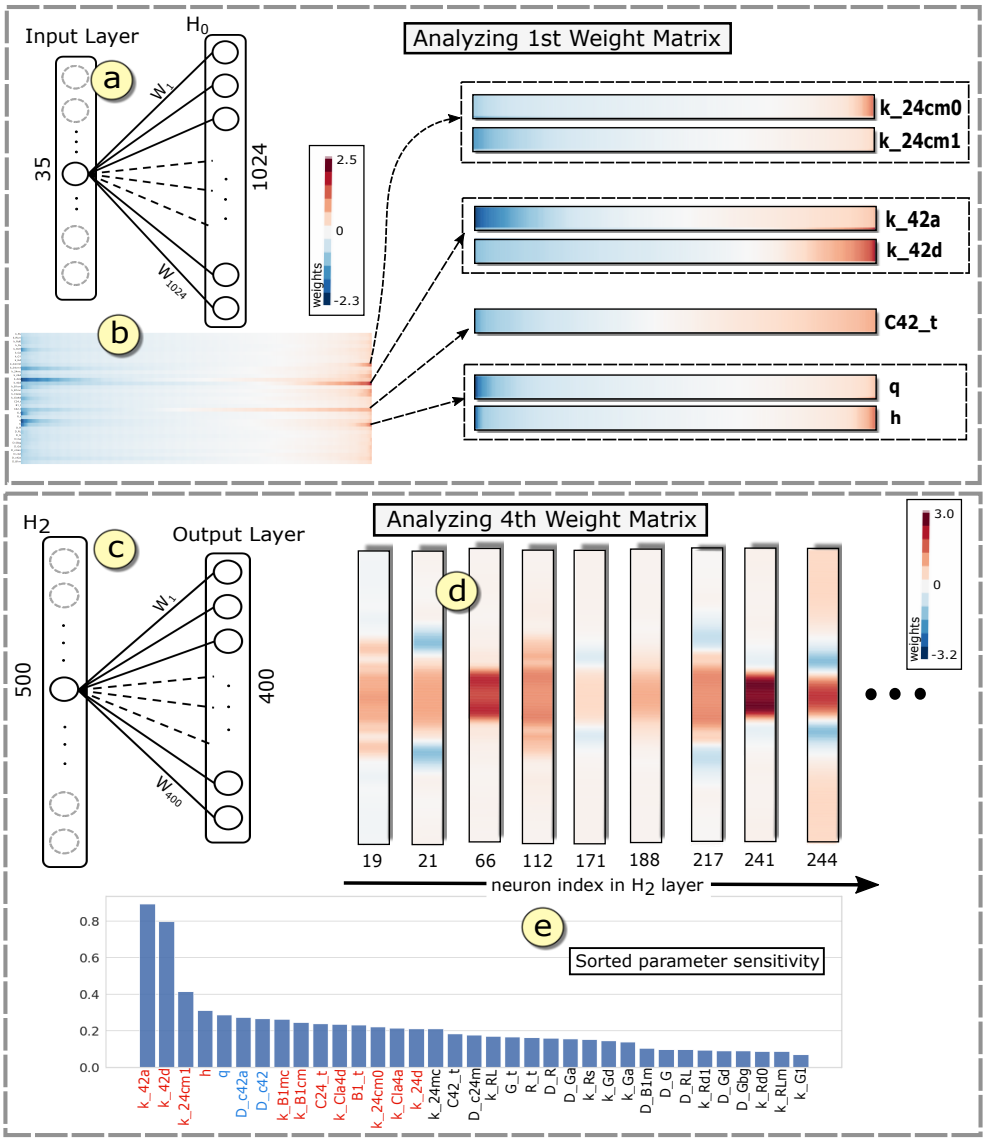


Figure 7.8: Knowledge extraction: (a) Connections of one parameter with H_0 layer. (b) Row-wise sorted first weight matrix. (c) Connections of a neuron in H_2 layer with the output layer. (d) Few selected weight patterns with high weights at the center. (e) Corresponding average parameter sensitivity sorted in descending order.

h) showed relatively strong correlation compared to the other parameters. The exact shape of the weight distributions for these parameters are provided in the supplementary materials.

Final Weight Matrix: We did not observe any distinct patterns in the second and the third weight matrices. However, in the final weight matrix of resolution 500×400,

corresponding to the third hidden layer and the output layer, we found multiple interesting weight distribution patterns (full matrix view is shown in Fig. 7.4(i)). Each row of this matrix corresponds to the weights assigned to individual neuron activation values in the penultimate layer (H_2) towards the 400 output neurons, as illustrated in Fig. 7.8(c). We observed that different neurons in H_2 layer assign high positive or high negative weights to different sets of output neurons. In the output layer, the neurons at the middle section corresponds to the top of the cell membrane modeled by the simulation. Therefore, it is interesting to identify which neurons in the H_2 layer assigns high positive weights to the middle section of the output layer, because those neurons are most likely to contribute towards producing high Cdc42 polarization results. We identified 95 such neurons in the H_2 layer (Fig. 7.8(d)) and evaluated their average parameter sensitivity (Section 7.4.3) to find out which parameters are more sensitive to produce the selected weight pattern in the penultimate layer.

Fig. 7.8(e) shows the sorted list of parameters based on descending order of their average normalized sensitivities. We compared this importance/sensitivity order of parameters for generating high polarization patterns with that of the list of highly sensitive parameters identified in the previous work of our experts [152]. We found that except the change in order of the 3 parameters `q`, `D_c42a`, and `D_c42` by one position (marked by blue texts), the top 15 sensitive parameters (red texts) were in the same order as previously identified by experts. Similar analysis can be perform for different weight distribution patterns in the penultimate layer of the network.

7.7 Domain Expert Feedback

Our experts from the field of computational biology comprise of a professor from the Department of Mathematics, who created the yeast simulation model, and two of her

graduate students. They feel that the proposed visual analysis system is a very useful tool to fine-tune their simulation model. They found the visual interface of our system to be simple and intuitive for users familiar with yeast simulation models. The ability to quickly prototype different parameter combinations and interactively visualize the predicted simulation output within seconds lets them easily calibrate the simulation model.

Previously, there was no interactive visualization created for the yeast simulation. Our experts feel that the visual analysis system will be a useful medium to communicate with the non-expert collaborators and stakeholders of the project, instead of explaining them the complex reaction-diffusion equations involved in the simulation. As discussed in Section 7.6.1, using our system we were able to discover new parameter configurations that can trigger high Cdc42 polarization in the original simulation model. This is a significant improvement over the previously estimated parameters using polynomial surrogate model analysis [152].

The experts feel that the system is flexible to work with other protein species besides Cdc42. The current backend, i.e, neural network-based surrogate model, predicting Cdc42 concentration, can be easily replaced with another neural network model predicting different species and still retain the same visual interactions to analyze the simulation. Since the backend analysis techniques are independent of the network structure, they can train a network with a different architecture and still utilize our visual analysis frontend. The experts also plan to utilize the radial clustering information to create an adaptive mesh for the computational domain rather than the current 400 uniformly spaced resolution.

The model analysis view was helpful to validate the trained network and see if the surrogate model is actually learning something relevant about the simulation rather than making random predictions. However, they feel that the model analysis view requires users

to have a good understanding of the neural network architecture to interpret the weight matrices. They feel it would be helpful to make the weight matrix analysis more intuitive for people without much machine learning background. Overall, the experts are satisfied with our neural network assisted visual analysis system and feel that it meets all of their expected requirements. They plan to use the findings from the visual analysis system to improve the simulation and report in a systems biology journal in future.

7.8 Discussion

Design Choices: The choice of using radial layouts for some of the visualizations is inspired from the circular shape of the computational domain (Fig. 7.1(c)). This helps retain the context of the cellular structure during the visual analysis workflow. Throughout the course of this project, we iterated over different design choices for various elements of our visual analysis system. Following are some of the key design decisions that we had to make in the process.

- Initially, we used 400 colored circles along the circumference of the simulation domain to visualize the concentration values (Fig. 7.9(a)). However, as the overall system grew in size, we had to reduce the size of the individual views, which significantly shrunk the size of the 400 small circles. Increasing the size of the small circles led to overlapping among the spatially neighboring points (Fig. 7.9(b)). Therefore, we changed the visual design to use contiguous rectangular boxes instead of circles at each point (Fig. 7.9(c)). As a result, we can scale the boxes radially without worrying about overlapping (Fig. 7.9(d)).

- Deciding on an optimal design layout for the *Parameter Control Board* with 35 different parameter control bars was a challenging task. One straight-forward choice

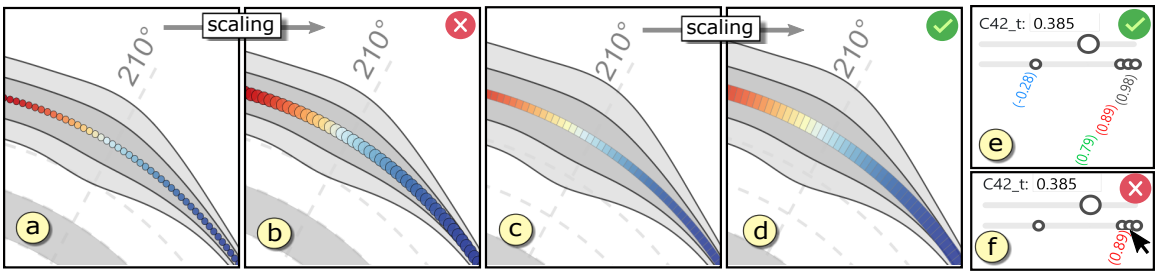


Figure 7.9: Design study: Using circles (a,b) versus using rectangular boxes (c,d) across the membrane. Parameter control bar with (e) all the values displayed versus using (f) mouse-hovering.

was to place the control bars for 35 parameters in a long vertical/horizontal panel with scrollbars to scroll through the list of individual control bars. However, the experts felt that it is important to have all the 35 bars visible at the same time, without the need to scroll around during the model calibration process.

- The next challenge was to find the best way of showing multiple recommended parameter values in the same view. Fig. 7.9(e) shows an instance in our current system with 3 recommended parameter values very close to each other. We lay them out vertically to make the close-by values standout. This helps the users in clicking on the texts and adjusting the parameter slider to that precise recommended value. Another choice to save space in the *Parameter Control Board* panel was to show the values only when the mouse hovers over the recommended nodes as shown in Fig. 7.9(f). However, we went with the first choice as it helps the experts to see all the recommended parameter values in the same view and not just the nodes in the bar. We believe that there could be better designs to address this, and we plan to explore other alternatives in the next version of our system.

Comparison with Previous Works: Previous visual analysis systems for simulation parameter exploration were all specifically designed to meet the requirements of their

respective domain applications. However, one popular choice among some of the systems [16, 130, 139] was to project the high-dimensional space to a low-dimensional space for visualization. We did not opt for this choice because our experts were interested in directly manipulating in the exact parameter space rather than in some latent space. This helps them to explicitly map the effect of the parameters to different output regions. Using the reduced latent space can be confusing to interpret the meaning in the high-dimensional space. One unique feature of our proposed approach is that all the *analysis tasks* and *navigation strategies* [163] supported in our system are carried out using a single analysis framework in the backend, i.e, the trained neural network. Whereas, when using other surrogate models for prediction, most of these backend activities have to be carried out separately [16, 20, 22, 163]. In this work, we analyzed a simulation with 35 input parameters. To the best of our knowledge, previous visual parameter analysis systems [20, 130, 163, 180] did not have to deal with such large number of simulation parameters.

Implementation and Performance: We used the *Keras* python library (*v2.2.4*) [5], with *TensorFlow* backend, to implement and train our neural network-based surrogate model. We trained the model on a NVIDIA Pascal P100 GPU for 5000 epochs. It took 59.31 minutes to train the network for 5000 epochs and the final accuracy of the model was 87.6%. To perform post-hoc operations like sensitivity analysis and activation maximization on the trained network, we used the *Keras-Vis* [95], which is a high-level analysis library for neural networks. To perform uncertainty quantification for neural networks, we wrote custom code to turn on the dropout layers for trained *Keras* models during prediction/testing phase. Our frontend visual analysis system was designed using *d3.js*. We used *flask* framework [2] to interact with the trained neural network from our visual analysis system. The average time to get the predicted simulation output from the trained neural network for a new parameter

configuration is 0.25 seconds, whereas, running the original simulation model for one configuration took 2.3 hours in a supercomputing cluster.

7.9 Conclusion

In this chapter, we have proposed an interactive visual analysis system to study and analyze a complex yeast cell polarization simulation model. The proposed system uses a trained neural network-based surrogate model as the backend analysis framework to facilitate interactive visual analysis. It allows the experts to interactively calibrate the simulation input parameters as well visually guide them towards discovering new parameter configurations. We also analyze the surrogate model to extract interesting insights about the original simulation model. We hope that our proposed approach can motivate researchers to look at neural networks as more than a prediction tool, and start utilizing them to conduct interesting analysis activities. In future, we would like to extend the visual analysis framework to facilitate more complex analysis tasks like the recently proposed *testing with concept activation vectors* (TCAV) [92]. This will allow the experts to validate high-level domain specific concepts in the surrogate model. We plan to apply similar visual analytic approach for analyzing simulations from other application domains as well. As suggested by our experts, we also plan to simplify the model analysis and validation methods so that it is more intuitive for people without much machine learning background.

Chapter 8: Conclusion and Future Work

8.1 Conclusion

In this dissertation, we proposed different statistical and machine learning based approaches to address the challenges of visualizing and analyzing large-scale simulation data. We covered three broad categories of data analysis challenges: (i) multivariate distribution-based data summarization, (ii) uncertainty analysis in ensemble simulation data, and (iii) simulation parameter analysis and exploration. We first proposed a copula-based approach to model multivariate distribution in scientific simulation data. In Chapter 3, we explained in detail the concept of copula function and its application as a flexible multivariate distribution modeling tool. For multivariate simulations, in Chapter 4, we described our copula-based distribution modeling framework to create multivariate data summaries in an *in situ* environment [72]. The stored data summaries are later used to perform post-hoc analysis activities like sampling-based scalar field visualization and probabilistic query-driven analysis. For ensemble simulation data, in Chapter 5, we showed how to use our proposed copula-based strategy to create a mixed distribution field to model data uncertainty, while maintaining the spatial correlations [71]. Such uncertainty representations were used to visualize uncertain features like isosurfaces and vortices in ensemble simulation data. In Chapter 6, we proposed a two-stage information-theoretic framework for the exploration of scalar values as well as

their corresponding ensemble isocontours [75]. Using *specific information* measures like *predictability* and *surprise* of specific scalar values, we evaluated the ensemble isocontour uncertainty of all the scalar values in an efficient way. We also proposed a *conditional entropy* based approach to identify the contribution of individual members towards the overall uncertainty of the ensemble isocontours of a selected scalar value. Finally, in Chapter 7, we proposed an interactive visual analysis system to analyze a computationally expensive yeast cell polarization simulation [74]. We utilized a trained neural network-based surrogate model as the backend analysis framework to facilitate interactive visual analysis. Using our system, the experts were able to interactively calibrate the simulation input parameters as well as discover new parameter configurations of interest.

8.2 Future Research Directions

In this section, we discuss some potential future research directions in the field of scientific data analysis and visualization that can be built on top of the statistical and machine learning approaches proposed in this dissertation. One promising direction is the application of deep learning-based models to solve the challenges of visualizing large-scale simulation data. In particular, we can use models like *variational autoencoders* (VAE) to create an end-to-end *in situ* framework to estimate important low-dimensional feature distributions of high-resolution scalar fields generated by large-scale scientific simulations. It is often computationally prohibitive to store the high-resolution scalar fields for each and every timestep. Therefore, scientists have to sacrifice a lot of the temporal information due to storage limitations. We can learn a latent space distribution (low-dimensional in nature) of the high-resolution scalar fields by training a VAE over the data generated from sparsely sampled time-steps of the simulation. The trained encoder can be then applied in an *in situ*

environment to create the low-dimensional feature distributions at finer temporal resolutions which the scientists can effort to write to the storage device. Using the corresponding trained decoder of the VAE, scientist can reconstruct the complete scalar field from the feature distribution, thus, facilitating efficient post-hoc analysis of finer temporal resolution. Another interesting research direction is the extension of interpretability methods for deep learning models. The neural network-based surrogate model proposed in this dissertation is a simple multilayer feed-forward network. We can also train complex deep neural networks as surrogates to large-scale scientific simulations. By opening up the "black-box" of the trained surrogate model using different interpretation techniques, we hope to offer novel analysis workflow for the scientists in future.

Appendix A: Theorems and Proofs

A.1 Sklar's Theorem

This relationship between copula functions and general multivariate distribution functions was formalized by *Sklar's theorem*, which can be stated as follows:

Theorem 1. ***(Sklar's Theorem)***

1. *Let F be a joint CDF with marginals $F_1, ...F_d$. Then, there exists a copula $C : [0,1]^d \rightarrow [0,1]$ such that*

$$F(x_1, ...x_d) = C(F_1(x_1), ...F_d(x_d)), \forall x_i \in [-\infty, +\infty] \tag{A.1}$$

   Furthermore, if the marginals are continuous, then the copula is unique.

2. *Conversely, if C is a copula and $F_1, ..., F_d$ are univariate CDFs, then F defined as in equation A.1 is a multivariate CDF with margins $F_1, ..., F_d$ and copula C.*

A.2 Proofs for Distribution Transformation Properties

In this section, we provide the proofs for the two CDF transformation properties illustrated in Figure 3.1 in the dissertation.

***Property 1*:** If U is a uniform random variable (i.e, $U \sim U[0,1]$) and F_X is a univariate CDF then its inverse function, $F_X^{-1}(U)$, corresponds to the random variable X, (i.e, $F_X^{-1}(U) \sim X$)

$$P(F_X^{-1}(U) \leq x) = F_X(x) \tag{A.2}$$

Proof:

$$\forall x \in \mathbf{R}, P(F_X^{-1}(U) \leq x) = P(U \leq F_X(x)) = F_X(x) \tag{A.3}$$

where the first equality is using the right-continuity property of $F_X(x)$, which always hold for every distribution function. ■

***Property 2*:** If a real valued random variable X has a continuous cumulative distribution function F_X then

$$F_X(X) \sim U[0,1] \tag{A.4}$$

Proof: Assume, F_X is a continuous CDF of random variable X. Consider, another random variable $Z = F_X(X)$. Note that Z will have values in the range $[0,1]$. Then,

$$F_Z(x) = P(F_X(X) \leq x) = P(X \leq F_X^{-1}(x)) = F_X(F_X^{-1}(x)) = x \tag{A.5}$$

On the other hand, if U is a uniform random variable,

$$F_U(x) = \int_R f_U(u)du = \int_0^x du = x \tag{A.6}$$

From Equation A.5 and Equation A.6 we see that $F_Z(x) = F_U(x) \forall x \in [0,1]$. Therefore, this proves that we can always transform a continuous CDF to a uniform distribution. ■

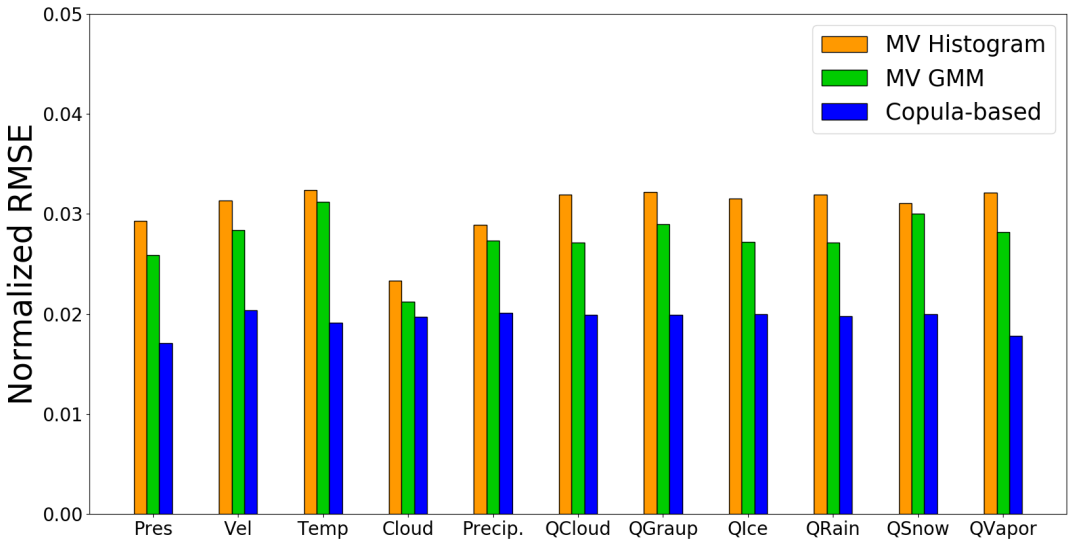


Figure B.1: Accuracy of the sample scalar fields for all 11 variables in Isabel for block size of 5^3.

Appendix B: Additional Results

The chart in Figure B.1 shows the normalized RMSE values of the sample scalar fields for all the 11 variables in the Isabel dataset. It compares the results of using our copula-based hybrid distribution framework against standard multivariate histogram and multivariate GMMs.

Figure B.2 shows the distribution of the other variables in the queried region for the Isabel dataset. For the multivariate query of $-2000 < \mathit{Pressure} < 500$ and $40 < \mathit{Velocity} < 50$, Figure B.2(a) shows the corresponding probability field for the queried range (i.e.,

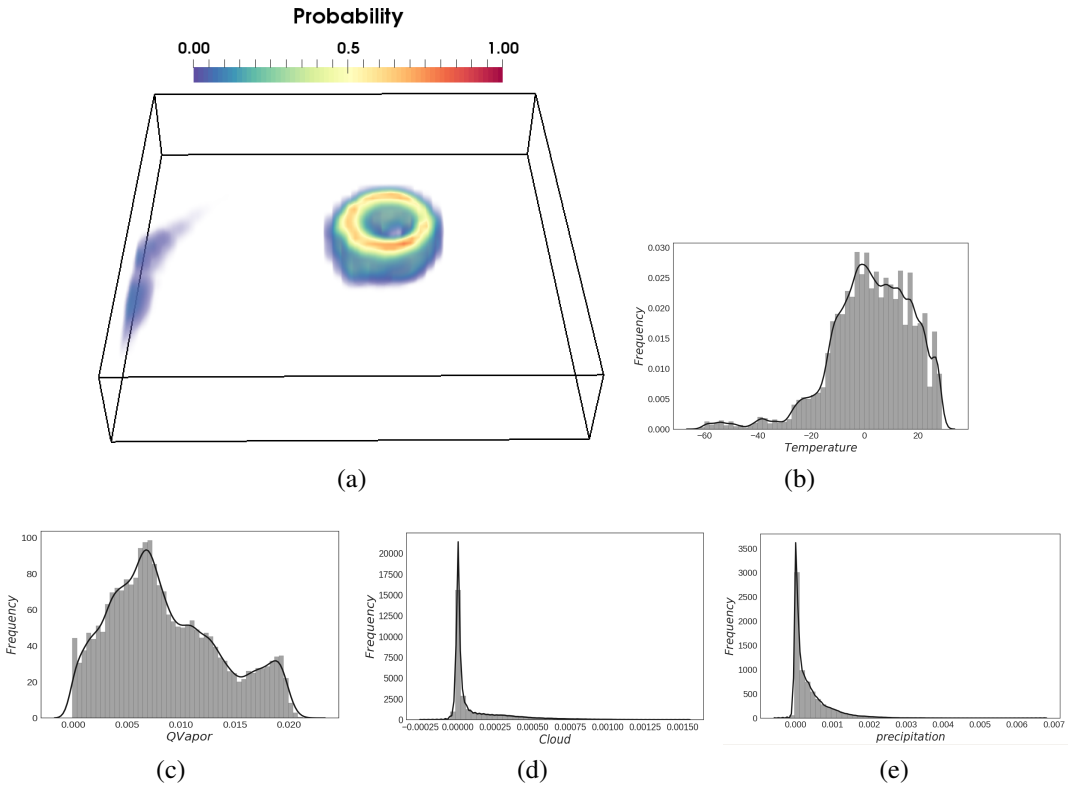
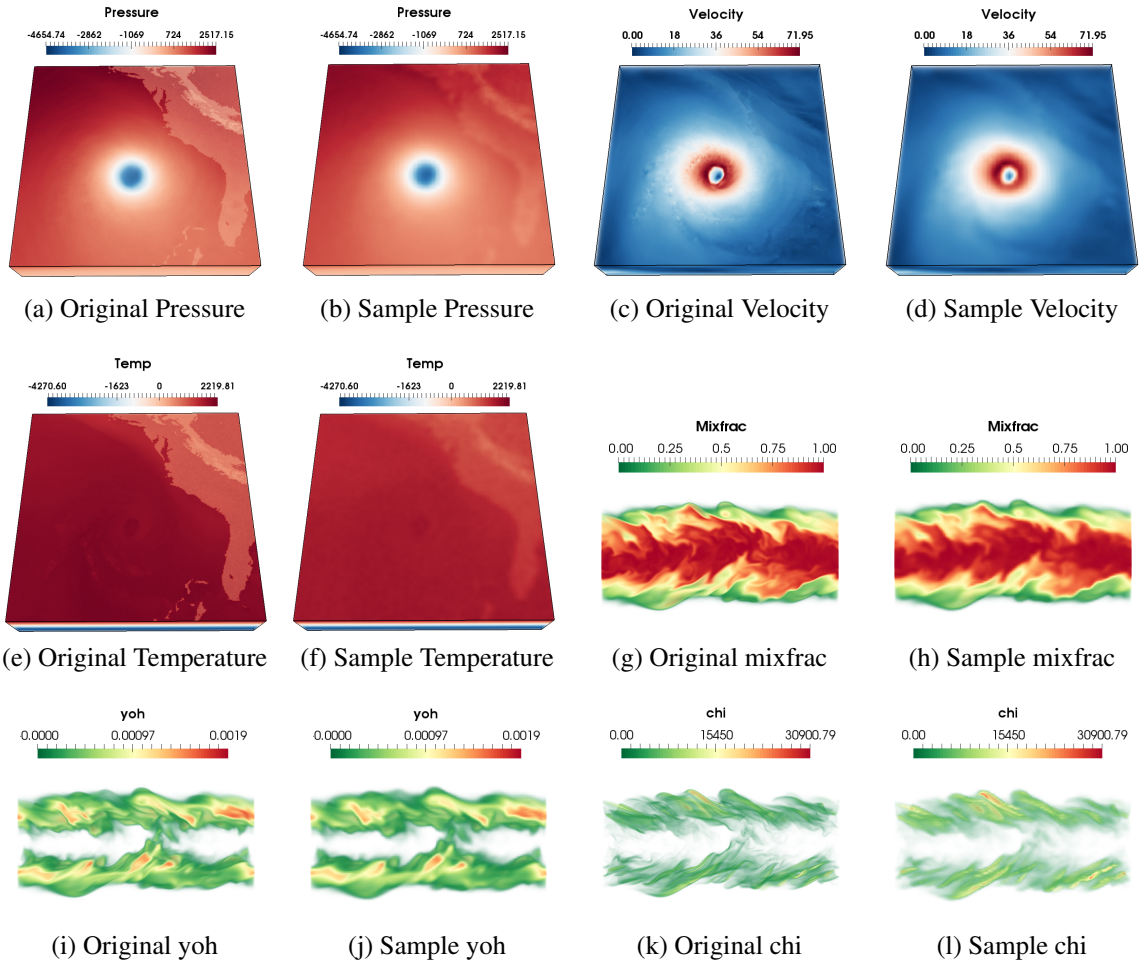


Figure B.2: Query-drive analysis in Isabel dataset: (a) $P(-2000 < Pres < 500 \text{ AND } 40 < Vel < 50)$. (b) $P(Temp| -2000 < Pres < 500 \text{ AND } 40 < Vel < 50)$. (c) $P(Qvapor| -2000 < Pres < 500 \text{ AND } 40 < Vel < 50)$. (d) $P(Cloud| -2000 < Pres < 500 \text{ AND } 40 < Vel < 50)$. (e) $P(Precip.| -2000 < Pres < 500 \text{ AND } 40 < Vel < 50)$.

$P(-2000 < Pressure < 500 \text{ AND } 40 < Velocity < 50))$. Figure B.2(b,c,d,e) show the distribution of Temperature, Qvapor, Cloud and Precipitation values respectively in the queried region.

Figure B.3 show few more sample scalar fields generated by our strategy along with their original raw fields. Figure B.3(a,c,e) show the original Pressure, Velocity and Temperature fields respectively from the Isabel dataset, while, their corresponding sample scalar fields generated by our copula-based sampling strategy are shown in B.3(b,d,f) respectively. Similarly, the original fields for variables mixfrac, yoh and chi in Combustion dataset is



(a) Original Pressure (b) Sample Pressure (c) Original Velocity (d) Sample Velocity

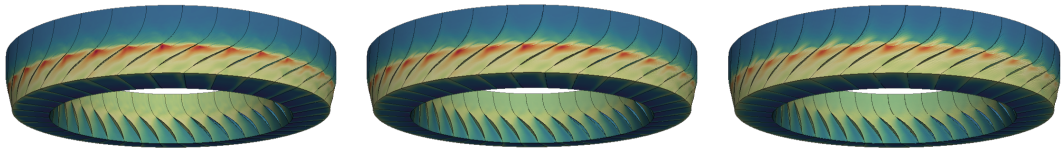
(e) Original Temperature (f) Sample Temperature (g) Original mixfrac (h) Sample mixfrac

(i) Original yoh (j) Sample yoh (k) Original chi (l) Sample chi

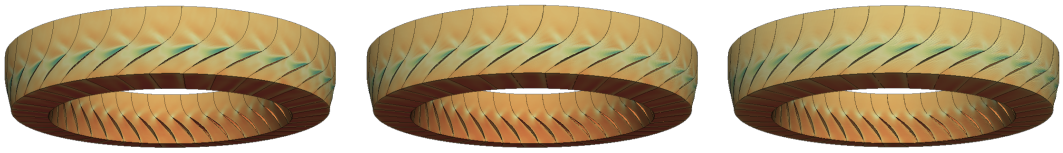
Figure B.3: Visual validation of the sample scalar fields for the variables in Isabel (a - f) and Combustion (g - l), generated in the same resolution as the original raw field.

shown in Figure B.3(g,i,k) respectively. The corresponding sample scalar fields are shown in B.3(h,j,l) respectively.

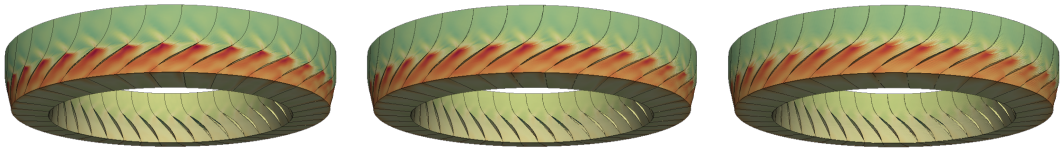
The sample scalar fields generated by our method can be in any arbitrary user-specified grid resolutions. Figure B.5 shows the results of the sample scalar fields for Pressure and Velocity variables at different resolution levels. Figure B.5(a,b,c) show the original Pressure field sub-sampled to resolutions of $125 \times 125 \times 25$, $100 \times 100 \times 20$ and $50 \times 50 \times 10$ respectively. Figure B.5(d,e,f) shows the sample scalar fields generated by our method with



(a) High Res (b) Medium Res (c) Low Res



(d) High Res (e) Medium Res (f) Low Res

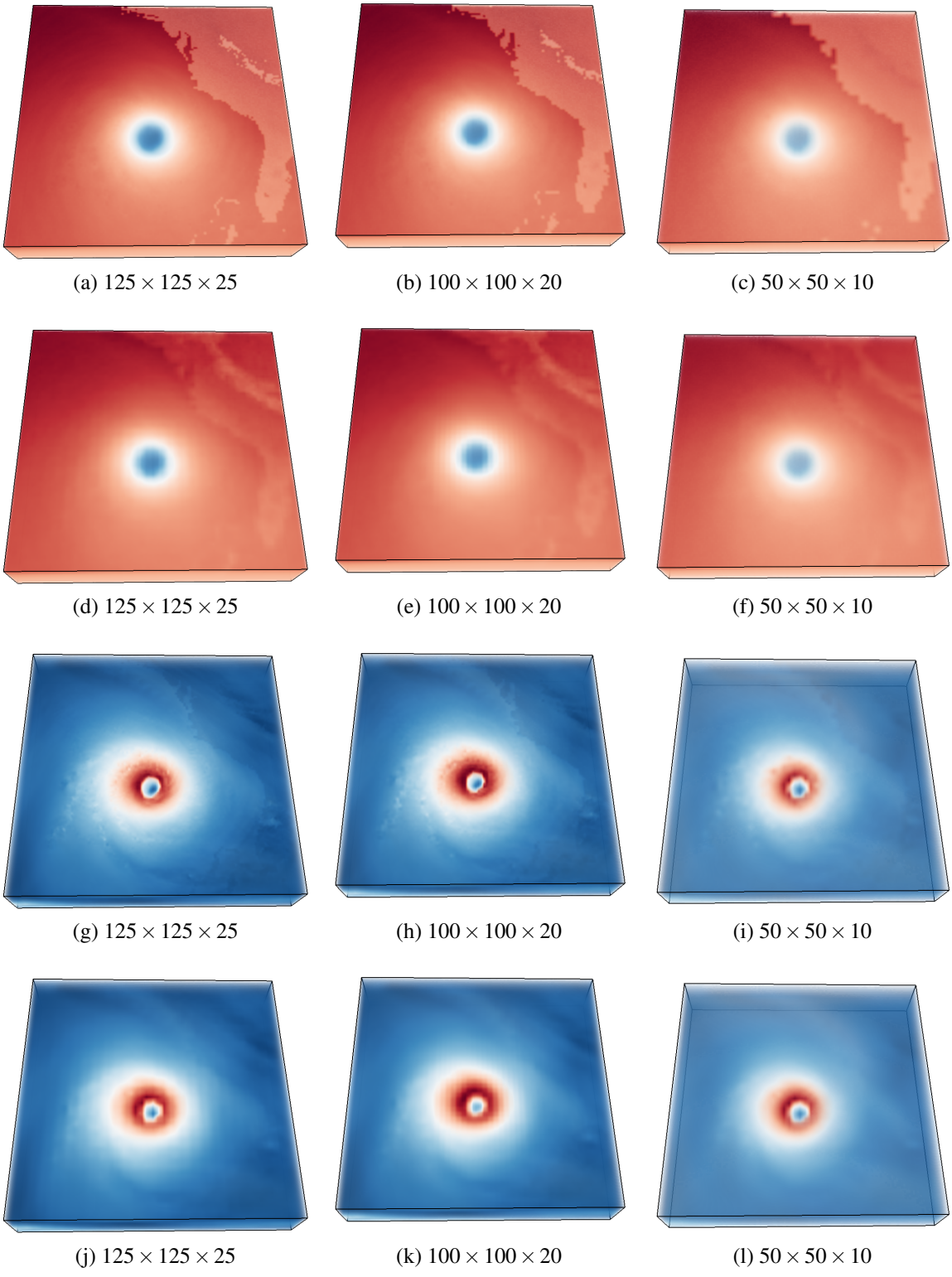


(g) High Res (h) Medium Res (i) Low Res

Figure B.4: Arbitrary grid resolutions for Jet turbine dataset: (a,b,c) Three different resolutions of Entropy variable. (d,e,f) Three different resolutions of Uvelocity variable. (g,h,i) Three different resolutions of Temperature variable.

the corresponding grid resolutions. Similarly, Figure B.5(g,h,i) shows the sub-sampled raw fields of Velocity variable in the respective resolutions, while Figure B.5(j,k,l) shows the corresponding sample scalar fields.

For the jet turbine dataset, which is in multi-block format with each block (36 turbine passage in total) of resolution $151 \times 71 \times 56$, we generate the sample fields in three resolutions; first in the original grid resolution, second in one-third the original resolutions i.e., $50 \times 23 \times 18$ (medium resolution) and third, in one-fifth the original resolution i.e., $30 \times 14 \times 11$ (low resolution). Figure B.4 shows the sample scalar fields for the three variables Entropy, Uvelocity and Temperature in three different resolution levels.



(a) $125 \times 125 \times 25$ (b) $100 \times 100 \times 20$ (c) $50 \times 50 \times 10$

(d) $125 \times 125 \times 25$ (e) $100 \times 100 \times 20$ (f) $50 \times 50 \times 10$

(g) $125 \times 125 \times 25$ (h) $100 \times 100 \times 20$ (i) $50 \times 50 \times 10$

(j) $125 \times 125 \times 25$ (k) $100 \times 100 \times 20$ (l) $50 \times 50 \times 10$

Figure B.5: Arbitrary grid resolution for Isabel:(a,b,c) Sub-sampled Pressure fields from the original raw data. (d,e,f) Corresponding sample Pressure fields generated by our method. (g,h,i) Sub-sampled Velocity fields from the original raw data. (j,k,l) Corresponding sample Velocity fields generated by our method.

Bibliography

[1] Exascale Computing Project. https://www.exascaleproject.org/.

[2] Flask (last accessed: 03-30-2019). http://flask.pocoo.org/.

[3] GEFS Global Ensemble Forecast System. https://www.ncdc.noaa.gov/.

[4] IEEE Scivis Contest 2016. http://www.uni-kl.de/sciviscontest/.

[5] Keras (last accessed: 03-30-2019). https://keras.io/.

[6] Ohio Supercomputer Center, Oakley. http://osc.edu/ark:/19495/hpc0cvqn.

[7] Scivis colors (last accessed: 03-30-2019). https://sciviscolor.org/home/colormaps/.

[8] James Ahrens, Sébastien Jourdain, Patrick O'Leary, John Patchett, David H. Rogers, and Mark Petersen. An image-based approach to extreme scale in situ visualization and analysis. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '14, pages 424–434, Piscataway, NJ, USA, 2014. IEEE Press.

[9] Oluwafemi S. Alabi, Xunlei Wu, Jonathan M. Harter, Madhura Phadke, Lifford Pinto, Hannah Petersen, Steffen Bass, Michael Keifer, Sharon Zhong, Chris Healey, and Russell M. Taylor II. Comparative visualization of ensembles using ensemble surface slicing. volume 8294, pages 82940U–82940U–12, 2012.

[10] Babak Alipanahi, Andrew Delong, Matthew T. Weirauch, and Brendan J. Frey. Predicting the sequence specificities of dna- and rna-binding proteins by deep learning. *Nature Biotechnology*, 33:831–838, 2015.

[11] T. Athawale and A. Entezari. Uncertainty quantification in linear interpolation for isosurface extraction. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2723–2732, Dec 2013.

[12] Tushar Athawale, Elham Sakhaee, and Alireza Entezari. Isosurface visualization of data with nonparametric models for uncertainty. *IEEE Trans. Vis. Comput. Graph.*, 22(1):777–786, 2016.

[13] Michaël Baudin, Anne Dutfoy, Bertrand Iooss, and Anne-Laure Popelin. *OpenTURNS: An Industrial Software for Uncertainty Quantification in Simulation*, pages 1–38. Springer International Publishing, Cham, 2016.

[14] A. C. Bauer, H. Abbasi, J. Ahrens, H. Childs, B. Geveci, S. Klasky, K. Moreland, P. O'Leary, V. Vishwanath, B. Whitlock, and E. W. Bethel. In situ methods, infrastructures, and applications on high performance computing platforms. *Computer Graphics Forum*, 35(3):577–597, 2016.

[15] K Bensema, L Gosink, H Obermaier, and K Joy. Modality-driven classification and visualization of ensemble variance. *IEEE transactions on visualization and computer graphics*, (5), 2015-12-10 00:00:00.0.

[16] Wolfgang Berger, Harald Piringer, Peter Filzmoser, and Eduard Gröller. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. In *Computer Graphics Forum*, volume 30, pages 911–920. Wiley Online Library, 2011.

[17] W. Bethel, L. Gosink, K. Joy, and J. Anderson. Variable interactions in query-driven visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13:1400–1407, 09 2007.

[18] T. G. Bever and D. Poeppel. Analysis by synthesis: a (re-) emerging program of research for language and vision,. *Biolinguistics*, 4(2-3):174–200, 2010.

[19] Alsallakh Bilal, Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. Do convolutional neural networks learn class hierarchy? *IEEE Transactions on Visualization and Computer Graphics*, 24(1):152–162, 2018.

[20] Ayan Biswas, Christopher M Biwer, David J Walters, James Ahrens, Devin Francom, Earl Lawrence, Richard L Sandberg, D Anthony Fredenburg, and Cynthia Bolme. An interactive exploration tool for high-dimensional datasets: A shock physics case study. *Computing in Science & Engineering*, 2018.

[21] Ayan Biswas, Soumya Dutta, Han-Wei Shen, and Jonathan Woodring. An information-aware framework for exploring multivariate data sets. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2683–2692, 2013.

[22] Ayan Biswas, Guang Lin, Xiaotong Liu, and Han-Wei Shen. Visualization of time-varying weather ensembles across multiple resolutions. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):841–850, 2017.

[23] Georges-Pierre Bonneau, Hans-Christian Hege, Chris R. Johnson, Manuel M. Oliveira, Kristin Potter, Penny Rheingans, and Thomas Schultz. *Overview and State-of-the-Art of Uncertainty Visualization*, pages 3–27. 2014.

[24] R. Bramon, I. Boada, A. Bardera, J. Rodriguez, M. Feixas, J. Puig, and M. Sbert. Multimodal data fusion based on mutual information. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1574–1587, Sept 2012.

[25] Ken Brodlie, Rodolfo Allendes Osorio, and Adriano Lopes. *A Review of Uncertainty in Data Visualization*, pages 81–109. Springer London, London, 2012.

[26] Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *J. Mach. Learn. Res.*, 13:27–66, January 2012.

[27] Stefan Bruckner and Torsten Moller. Isosurface similarity maps. In *Proceedings of the 12th Eurographics / IEEE - VGTC Conference on Visualization*, EuroVis'10, pages 773–782, 2010.

[28] J. Chanussot, A. Clement, B. Vigouroux, and J. Chabod. Lossless compact histogram representation for multi-component images: application to histogram equalization. In *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No.03CH37477)*, volume 6, pages 3940–3942 vol.6, July 2003.

[29] A. Chaudhuri, T. H. Wei, T. Y. Lee, H. W. Shen, and T. Peterka. Efficient range distribution query for visualizing scientific data. In *2014 IEEE Pacific Visualization Symposium*, pages 201–208, March 2014.

[30] C. M. Chen, S. Dutta, X. Liu, G. Heinlein, H. W. Shen, and J. P. Chen. Visualization and analysis of rotating stall for transonic jet engine simulation. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):847–856, Jan 2016.

[31] Jen-Ping Chen, Michael D. Hathaway, and Gregory P. Herrick. Prestall behavior of a transonic axial compressor stage via time-accurate numerical simulation. *Journal of Turbomachinery*, 130(4):041014, 2008.

[32] Jenping Chen, Robert Webster, Michael Hathaway, Gregory Herrick, and Gary Skoch. Numerical simulation of stall and stall control in axial and radial compressors. In *44th AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, 2006.

[33] M. Chen and H. Jänicke. An information-theoretic framework for visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1206–1215, 2010.

[34] Min Chen, Mateu Sbert, Han-Wei Shen, Ivan Viola, Anton Bardera, and Miquel Feixas. Information Theory in Visualization. In Augusto Sousa and Kadi Bouatouch, editors, *EG 2016 - Tutorials*. The Eurographics Association, 2016.

[35] U. Cherubini and E. Luciano. Bivariate option pricing with copulas. *Applied Mathematical Finance*, 9:69–85, 2002.

[36] Hank Childs. Data exploration at the exascale. *Supercomputing frontiers and innovations*, 2(3), 2015.

[37] Dane Coffey, Chi-Lun Lin, Arthur G Erdman, and Daniel F Keefe. Design by dragging: An interface for creative forward and inverse design with simulation ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2783–2791, 2013.

[38] Paulo Cortez and Mark J Embrechts. Using sensitivity analysis and visualization techniques to open black box data mining models. *Information Sciences*, 225:1–17, 2013.

[39] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[40] Ralph B. D'agostino, Albert Belanger, and Ralph B. D'agostino Jr. A suggestion for using powerful and informative tests of normality. *The American Statistician*, 44(4):316–321, 1990.

[41] I. Demir, C. Dick, and R. Westermann. Multi-charts for comparative 3d ensemble visualization. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2694–2703, Dec 2014.

[42] Michael R Deweese and Markus Meister. How to measure the information gained from one symbol. *Network: Computation in Neural Systems*, 10(4):325–340, 1999.

[43] Suzana Djurcilov, Kwansik Kim, Pierre Lermusiaux, and Alex Pang. Visualizing scalar volumetric data with uncertainty. *Computers and Graphics*, 26:239–248, 2002.

[44] Geoffrey M. Draper, Yarden Livnat, and Richard F. Riesenfeld. A survey of radial methods for information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(5):759–776, September 2009.

[45] S. Dutta, C. M. Chen, G. Heinlein, H. W. Shen, and J. P. Chen. In situ distribution guided analysis and visualization of transonic jet engine simulations. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):811–820, Jan 2017.

[46] S. Dutta, H. W. Shen, and J. P. Chen. In situ prediction driven feature analysis in jet engine simulations. In *2018 IEEE Pacific Visualization Symposium (PacificVis)*, pages 66–75, April 2018.

[47] S. Dutta, J. Woodring, H. W. Shen, J. P. Chen, and J. Ahrens. Homogeneity guided probabilistic data summaries for analysis and visualization of large-scale data sets. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pages 111–120, April 2017.

[48] Soumya Dutta, Xiaotong Liu, Ayan Biswas, Han-Wei Shen, and Jen-Ping Chen. Pointwise information guided visual analysis of time-varying multi-fields. In *SIGGRAPH ASIA 2017 Symposium on Visualization*, SA '17. ACM, 2017.

[49] Soumya Dutta and Han-Wei Shen. Distribution driven extraction and tracking of features for time-varying data analysis. *IEEE Trans. on Vis. and Comp. Graphics*, 22(1):837–846, 2016.

[50] Gal Elidan. *Copulas in Machine Learning*, pages 39–60. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[51] Paul D. Ellis. *The essential guide to effect sizes: statistical power, meta-analysis, and the interpretation of research results*. Cambridge University Press, Cambridge ; New York, 2010.

[52] Matt Elsey, Selim Esedoğlu, and Peter Smereka. Large-scale simulations and parameter study for a simple recrystallization model. *Philosophical Magazine*, 91(11):1607–1642, 2011.

[53] Paul Embrechts, Filip Lindskog, and Alexander McNeil. Modelling dependence with copulas and applications to risk management. *Handbook of heavy tailed distributions in finance*, 8(1):329–384, 2003.

[54] A. Endert, W. Ribarsky, C. Turkay, B.L. William Wong, I. Nabney, I. Daz Blanco, and F. Rossi. The state of the art in integrating machine learning into visual analytics. *Computer Graphics Forum*, 36(8):458–486, 2017.

[55] N. Fabian, K. Moreland, D. Thompson, A. C. Bauer, P. Marion, B. Gevecik, M. Rasquin, and K. E. Jansen. The paraview coprocessing library: A scalable, general purpose in situ visualization library. In *2011 IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pages 89–96, 2011.

[56] Florian Ferstl, Mathias Kanzler, Marc Rautenhaus, and Rüdiger Westermann. Visual analysis of spatial variability and global correlations in ensembles of iso-contours. *Computer Graphics Forum (Proc. EuroVis)*, 35(3):221–230, 2016.

[57] David F. Findley. Counterexamples to parsimony and bic. *Annals of the Institute of Statistical Mathematics*, 43(3):505–514, 1991.

[58] Alexander I. J. Forrester, Andras Sobester, and Andy J. Keane. *Engineering Design via Surrogate Modelling - A Practical Guide.* Wiley, 2008.

[59] R. Fuchs and H. Hauser. Visualization of multivariate scientific data. *Computer Graphics Forum*, 28(6):1670–1690.

[60] Ryohei Fujimaki, Yasuhiro Sogawa, and Satoshi Morinaga. Online heterogeneous mixture modeling with marginal and copula selection. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 645–653, New York, NY, USA, 2011. ACM.

[61] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1050–1059. JMLR.org, 2016.

[62] A. Ghasemi and S. Zahediasl. Normality tests for statistical analysis: A guide for non-statisticians. *Int J Endocrinol Metab*, 10(2):486–489, 2012.

[63] Ayana Ghosh, Lydie Louis, Kapildev K Arora, Bruno C Hancock, Joseph F Krzyzaniak, Paul Meenan, Serge Nakhmanson, and Geoffrey PF Wood. Assessment of machine learning approaches for predicting the crystallization propensity of active pharmaceutical ingredients. *CrystEngComm*, 21(8):1215–1223, 2019.

[64] Ayana Ghosh, Filip Ronning, Serge Nakhmanson, and Jian-Xin Zhu. Understanding magnetic properties of actinide-based compounds from machine learning. *arXiv preprint arXiv:1907.10587*, 2019.

[65] John R Goodall, Eric D Ragan, Chad A Steed, Joel W Reed, G David Richardson, Kelly MT Huffer, Robert A Bridges, and Jason A Laska. Situ: Identifying and explaining suspicious behavior in networks. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):204–214, 2019.

[66] Dirk Gorissen, Ivo Couckuyt, Piet Demeester, Tom Dhaene, and Karel Crombecq. A surrogate modeling and adaptive sampling toolbox for computer based design. *J. Mach. Learn. Res.*, 11:2051–2055, August 2010.

[67] L.J. Gosink, C. Garth, J.C. Anderson, E.W. Bethel, and K.I. Joy. An application of multivariate statistical analysis for query-driven visualization. *IEEE Trans. on Vis. and Comp. Graphics*, 17(3):264–275, 2011.

[68] DM Hamby. A review of techniques for parameter sensitivity analysis of environmental models. *Environmental monitoring and assessment*, 32(2):135–154, 1994.

[69] F. Han and H. Liu. High dimensional semiparametric scale-invariant principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(10):2016–2032, Oct 2014.

[70] Fang Han and Han Liu. Semiparametric principal component analysis. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 171–179. Curran Associates, Inc., 2012.

[71] 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, Jan 2018.

[72] S. Hazarika, S. Dutta, H. Shen, and J. Chen. Codda: A flexible copula-based distribution driven analysis framework for large-scale multivariate data. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):1214–1224, Jan 2019.

[73] S. Hazarika, S. Dutta, and H. W. Shen. Visualizing the variations of ensemble of isosurfaces. In *2016 IEEE Pacific Visualization Symposium (PacificVis)*, pages 209–213, April 2016.

[74] S. Hazarika, H. Li, K. Wang, H. Shen, and C. Chou. Nnva: Neural network assisted visual analysis of yeast cell polarization simulation. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2019.

[75] Subhashis Hazarika, Ayan Biswas, Soumya Dutta, and Han-Wei Shen. Information guided exploration of scalar values and isocontours in ensemble datasets. *Entropy*, 20(7), 2018.

[76] Robert Hecht-Nielsen. Neural networks for perception (vol. 2). chapter Theory of the Backpropagation Neural Network, pages 65–93. Harcourt Brace & Co., Orlando, FL, USA, 1992.

[77] Christian Hentschel and Harald Sack. What image classifiers really see–visualizing bag-of-visual words models. In *International Conference on Multimedia Modeling*, pages 95–104. Springer, 2015.

[78] Jerry L. Hintze and Ray D. Nelson. Violin plots: A box plot-density trace synergism. *The American Statistician*, 52(2):181–184, 1998.

[79] Fred Hohman, Minsuk Kahng, Robert Pienta, and Duen Horng Chau. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Transactions on Visualization and Computer Graphics*, 2018.

[80] Xiaolei Huang, Nikos Paragios, and Dimitris N. Metaxas. Shape registration in implicit spaces using information theory and free form deformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(8):1303–1318, August 2006.

[81] Aleks Jakulin, Martin Možina, Janez Demšar, Ivan Bratko, and Blaž Zupan. Nomograms for visualizing support vector machines. In *Proceedings of the eleventh ACM*

SIGKDD international conference on Knowledge discovery in data mining, pages 108–117. ACM, 2005.

[82] H. Jänicke, A. Wiebel, G. Scheuermann, and W. Kollmann. Multifield visualization using local statistical complexity. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1384–1391, Nov 2007.

[83] M. Jarema, I. Demir, J. Kehrer, and R. Westermann. Comparative visual analysis of vector field ensembles. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 81–88, Oct 2015.

[84] Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61 – 70, 2011.

[85] C. Johnson. Top scientific visualization research problems. *IEEE Computer Graphics and Applications*, 24(4):13–17, July 2004.

[86] C. R. Johnson and A. R. Sanderson. A next step: Visualizing errors and uncertainty. *IEEE Computer Graphics and Applications*, 23(5):6–10, Sept 2003.

[87] Mark W. Jones, J. Andreas Baerentzen, and Milos Sramek. 3d distance fields: A survey of techniques and applications. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):581–599, July 2006.

[88] Minsuk Kahng, Pierre Y Andrews, Aditya Kalro, and Duen Horng Polo Chau. Activis: Visual exploration of industry-scale deep neural network models. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):88–97, 2018.

[89] Minsuk Kahng, Nikhil Thorat, Duen Horng Polo Chau, Fernanda B Viégas, and Martin Wattenberg. Gan lab: Understanding complex deep generative models using interactive visual experimentation. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):310–320, 2019.

[90] David Kao, Alison Luo, Jennifer L. Dungan, and Alex Pang. Visualizing spatially varying distribution data. In *Proceedings of the Sixth International Conference on Information Visualisation, 2002*, pages 219–225, 2002.

[91] Javed Khan, Jun S. Wei, Markus Ringner, Lao H. Saal, Marc Ladanyi, Frank Westermann, Frank Berthold, Manfred Schwab, Cristina R. Antonescu, Carsten Peterson, and Paul S. Meltzer. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat Med*, 7(6):673–679, june 2001.

[92] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV) . *ICML*, 2018.

[93] Sergey Kirshner and Barnabás Póczos. Ica and isa using schweizer-wolff measure of dependence. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 464–471, New York, NY, USA, 2008. ACM.

[94] Jack PC Kleijnen, Greet van Ham, and Jan Rotmans. Techniques for sensitivity analysis of simulation models: a case study of the co2 greenhouse effect. *Simulation*, 58(6):410–417, 1992.

[95] Raghavendra Kotikalapudi and contributors. keras-vis (last accessed: 03-30-2019). `https://github.com/raghakot/keras-vis`, 2017.

[96] Marjan Kuchaki Rafsanjani, Zahra Asghari, and Nasibeh Emami. A survey of hierarchical clustering algorithms. *The Journal of Mathematics and Computer Science*, 5:229–240, 10 2012.

[97] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statistics*, 22:79–86, 1951.

[98] Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John Stasko, and Haesun Park. ivisclustering: An interactive visual document clustering via topic modeling. In *Computer graphics forum*, volume 31, pages 1155–1164. Wiley Online Library, 2012.

[99] H. Lehmann and B. Jung. In-situ multi-resolution and temporal data compression for visual exploration of large-scale scientific simulations. In *IEEE 4th Symposium on Large Data Analysis and Visualization (LDAV), 2014*, pages 51–58, 2014.

[100] Pierre FJ Lermusiaux. Uncertainty estimation and prediction for interdisciplinary ocean dynamics. *Journal of Computational Physics*, 217(1):176–199, 2006.

[101] Pierre FJ Lermusiaux, Ching-Sang Chiu, Glen G Gawarkiewicz, Phil Abbot, Allan R Robinson, Robert N Miller, Patrick J Haley, Wayne G Leslie, Sharanya J Majumdar, Alex Pang, et al. Quantifying uncertainties in ocean predictions. Technical report, DTIC Document, 2006.

[102] Yuan Liang, Xiting Wang, Song-Hai Zhang, Shi-Min Hu, and Shixia Liu. Photorecomposer: Interactive photo recomposition by cropping. *IEEE Transactions on Visualization and Computer Graphics*, 24(10):2728–2742, 2018.

[103] T. Liebmann and G. Scheuermann. Critical points of gaussian-distributed scalar fields on simplicial grids. *Computer Graphics Forum*, 35(3):361–370, 2016.

[104] Mengchen Liu, Jiaxin Shi, Zhen Li, Chongxuan Li, Jun Zhu, and Shixia Liu. Towards better analysis of deep convolutional neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):91–100, 2017.

[105] Shixia Liu, Changjian Chen, Yafeng Lu, Fangxin Ouyang, and Bin Wang. An interactive method to improve crowdsourced annotations. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):235–245, 2019.

[106] Shixia Liu, Michelle X Zhou, Shimei Pan, Yangqiu Song, Weihong Qian, Weijia Cai, and Xiaoxiao Lian. Tiara: Interactive, topic-based visual text summarization and analysis. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2):25, 2012.

[107] Shusen Liu, J.A. Levine, P. Bremer, and V. Pascucci. Gaussian mixture model based volume visualization. In *2012 IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pages 73–77, 2012.

[108] A. L. Love, A. Pang, and D. L. Kao. Visualizing spatial multivalue data. *IEEE Computer Graphics and Applications*, 25(3):69–79, May 2005.

[109] Kewei Lu and Han-Wei Shen. A compact multivariate histogram representation for query-driven visualization. In *Proceedings of the 2015 IEEE 5th Symposium on Large Data Analysis and Visualization (LDAV)*, LDAV '15, pages 49–56, 2015.

[110] Zhicong Lu, Mingming Fan, Yun Wang, Jian Zhao, Michelle Annett, and Daniel Wigdor. Inkplanner: Supporting prewriting via intelligent visual diagramming. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):277–287, 2019.

[111] Claes Lundstrom, Patric Ljung, and Anders Ynnerman. Local histograms for design of transfer functions in direct volume rendering. *IEEE Trans. on Vis. and Comp. Graphics*, 12(6):1570–1579, 2006.

[112] Alison Luo, David Kao, and Alex Pang. Visualizing spatial distribution data sets. In *Proceedings of the Symposium on Data Visualisation 2003*, VISSYM '03, pages 29–38, 2003.

[113] Jian Ma and Zengqi Sun. Copula component analysis. *CoRR*, abs/cs/0703095, 2007.

[114] Abhijit Mahalanobis, Bhagavatula Vijaya, and Alan Nevel. Volume correlation filters for recognizing patterns in 3d data, 2001.

[115] G Elisabeta Marai, Chihua Ma, Andrew Burks, Filippo Pellolio, Guadalupe M Canahuate, David M Vock, Abdallah SR Mohamed, and Clifton David Fuller. Precision risk analysis of cancer therapy with interactive nomograms and survival plots. *IEEE Transactions on Visualization and Computer Graphics*, 2018.

[116] Alexander J. McNeil, Rdiger Frey, and Paul Embrechts. *Quantitative risk management: concepts, techniques and tools*. Princeton series in finance. Princeton University Press, Princeton (N.J.), 2005.

[117] Diane Micard, Y. Dossmann, and Louis Gostiaux. Mixing Efficiency in a Lock Exchange Experiment. In *VIII th Int. Symp. on Stratified Flows*, Proceedings of the VIIIth Int. Symp. on Stratified Flows, 2016.

[118] Mihaela Mihai and Rdiger Westermann. Visualizing the stability of critical points in uncertain scalar fields. *Computer Graphics Forum*, 41(0):13–25, 2014.

[119] Yao Ming, Shaozu Cao, Ruixiang Zhang, Zhen Li, Yuanzhe Chen, Yangqiu Song, and Huamin Qu. Understanding hidden memories of recurrent neural networks. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 13–24. IEEE, 2017.

[120] T. Miyoshi, K. Kondo, and K. Terasaki. Big ensemble data assimilation in numerical weather prediction. *Computer*, 48(11):15–21, Nov 2015.

[121] F. Molteni, R. Buizza, T. Palmer, and T. Petroliagis. The ecmwf ensemble prediction system: Methodology and validation. *Q.J.R. Meteorol. Soc*, 122(11):73119, 1996.

[122] Grgoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1 – 15, 2018.

[123] T. Munzner. *Visualization Analysis and Design*. AK Peters Visualization Series. CRC Press, 2015.

[124] James M Murphy, David MH Sexton, David N Barnett, Gareth S Jones, Mark J Webb, Matthew Collins, and David A Stainforth. Quantification of modelling uncertainties in a large ensemble of climate change simulations. *Nature*, 430(7001):768, 2004.

[125] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress.

[126] Roger B. Nelsen, José Juan Quesada-Molina, José Antonio Rodriguez-Lallena, and Manuel Úbeda-Flores. On the construction of copulas and quasi-copulas with given diagonal sections. *Insurance: Mathematics and Economics*, 42:473–483, 2008.

[127] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, pages 3387–3395, 2016.

[128] Xuan Vinh Nguyen, Jeffrey Chan, Simone Romano, and James Bailey. Effective global approaches for mutual information based feature selection. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 512–521, New York, NY, USA, 2014. ACM.

[129] Henriette Obermaier, Kenneth Joy, et al. Future challenges for ensemble visualization. *Computer Graphics and Applications, IEEE*, 34(3):8–11, 2014.

[130] Daniel Orban, Daniel F Keefe, Ayan Biswas, James Ahrens, and David Rogers. Drag and track: A direct manipulation interface for contextualizing data instances within a continuous parameter space. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):256–266, 2019.

[131] R.S. Allendes Osorio and K.W. Brodlie. Contouring with uncertainty. In *6th Theory and Practice of Computer Graphics Conference*, pages 59–66, 2008.

[132] Mathias Otto, Tobias Germer, Hans-Christian Hege, and Holger Theisel. Uncertain 2d vector field topology. *Computer Graphics Forum*, 29(2):347–356, 2010.

[133] Mathias Otto and Holger Theisel. Vortex analysis in uncertain vector fields. In *Computer Graphics Forum*, volume 31, pages 1035–1044. Blackwell Publishing Ltd, 2012.

[134] Alex Pang, Craig Wittenbrink, and Suresh Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 13(8):370–390, Nov 1997.

[135] Tom Peterka, Hadrien Croubois, Nan Li, Esteban Rangel, and Franck Cappello. Self-adaptive density estimation of particle data. *SIAM Journal on Scientific Computing*, 38(5):S646–S666, 2016.

[136] Christoph Petz, Kai Pthkow, and Hans-Christian Hege. Probabilistic local features in uncertain vector fields with spatial correlation. *Computer Graphics Forum*, 31(3pt2):1045–1054, 2012.

[137] T. Pfaffelmoser, M. Reitinger, and R. Westermann. Visualizing the positional and geometrical variability of isosurfaces in uncertain scalar fields. In *Computer Graphics Forum*, volume 30, pages 951–960. Wiley Online Library, 2011.

[138] Tobias Pfaffelmoser and Rdiger Westermann. Visualization of Global Correlation Structures in Uncertain 2D Scalar Fields. *Computer Graphics Forum*, 2012.

[139] Harald Piringer, Wolfgang Berger, and Jürgen Krasser. Hypermoval: Interactive visual validation of regression models for real-time simulation. In *Computer Graphics Forum*, volume 29, pages 983–992. Wiley Online Library, 2010.

[140] Josien P. W. Pluim, J. B. Antoine Maintz, and Max A. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE Transcations on Medical Imaging*, pages 986–1004, 2003.

[141] 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, Oct 2011.

[142] Kai Pöthkow and Hans-Christian Hege. Nonparametric models for uncertainty visualization. *Computer Graphics Forum*, 32(3pt2):131–140, 2013.

[143] Kai Pöthkow, Christoph Petz, and Hans-Christian Hege. Approximate level-crossing probabilities for interactive visualization of uncertain isocontours. *International Journal for Uncertainty Quantification*, 3(2), 2013.

[144] Kai Pöthkow, Britta Weber, and Hans-Christian Hege. Probabilistic marching cubes. In *Proceedings of the 13th Eurographics / IEEE - VGTC Conference on Visualization*, EuroVis'11, pages 931–940, Chichester, UK, 2011. The Eurographs Association ; John Wiley ; Sons, Ltd.

[145] Kristin Potter, Joe Kniss, Richard Riesenfeld, and Chris R. Johnson. Visualizing summary statistics and uncertainty. *Computer Graphics Forum (Proceedings of Eurovis 2010)*, 29(3):823–831, 2010.

[146] Kristin Potter, Jens Krüger, and Christopher Johnson. Towards the visualization of multi-dimensional stochastic distribution data. In *Proceedings of The International Conference on Computer Graphics and Visualization (IADIS) 2008*, 2008.

[147] Kristin Potter, Paul Rosen, and Chris R Johnson. From quantification to visualization: A taxonomy of uncertainty visualization approaches. In *Uncertainty Quantification in Scientific Computing*, pages 226–249. Springer, 2012.

[148] Kristin Potter, Andrew Wilson, Peer timo Bremer, Dean Williams, Charles Doutriaux, Valerio Pascucci, and Chris Johhson. Visualization of uncertainty and ensemble data: Exploration of climate modeling and weather forecast data with integrated visus-cdat systems, 2009.

[149] Nestor V. Queipo, Raphael T. Haftka, Wei Shyy, Tushar Goel, Rajkumar Vaidyanathan, and P. Kevin Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1 – 28, 2005.

[150] Nornadiah Mohd Razali, Yap Bee Wah, et al. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *Journal of statistical modeling and analytics*, 2(1):21–33, 2011.

[151] Saman Razavi, Bryan A. Tolson, and Donald H. Burn. Review of surrogate modeling in water resources. *Water Resources Research*, 48(7), 2012.

[152] Marissa Renardy, Tau-Mu Yi, Dongbin Xiu, and Ching-Shan Chou. Parameter uncertainty quantification using surrogate models applied to a spatial model of yeast mating polarization. *PLOS Computational Biology*, 14(5):1–26, 05 2018.

[153] Mélanie Rey. Copula models in machine learning. 2015.

[154] Mélanie Rey and Volker Roth. Copula Mixture Model for Dependency-seeking Clustering. *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 927–934, 2012.

[155] Mélanie Rey and Volker Roth. Meta-gaussian information bottleneck. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012.*, pages 1925–1933, 2012.

[156] Oliver Ruebel, E. Wes Bethel, Mr. Prabhat, and Kesheng Wu. Query-driven visualization and analysis. 2012.

[157] S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Visualization and Computer Graphics*, 21(3):660–674, 1991.

[158] Jibonananda Sanyal, Song Zhang, Jamie Dyer, Andrew Mercer, Philip Amburn, and Robert J Moorhead. Noodles: A tool for visualization of numerical weather model ensemble uncertainty. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1421–1430, 2010.

[159] N. Sauber, H. Theisel, and H. p. Seidel. Multifield-graphs: An approach to visualizing correlations in multifield scalar data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):917–924, Sept 2006.

[160] 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, Dec 2012.

[161] Thorsten Schmidt. Coping with Copulas. *Copulas - From Theory to Applications in Finance*, (15):1–23, 2006.

[162] Kristof Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus-Robert Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature Communications*, 8, 01 2017.

[163] Michael Sedlmair, Christoph Heinzl, Stefan Bruckner, Harald Piringer, and Torsten Möller. Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2161–2170, 2014.

[164] John Shalf, Sudip Dosanjh, and John Morrison. Exascale computing technology challenges. In *Proceedings of the 9th International Conference on High Performance Computing for Computational Science*, VECPAR'10, pages 1–25, Berlin, Heidelberg, 2011. Springer-Verlag.

[165] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52:591–611, 1965.

[166] A. Sklar. *Fonctions de repartition a n dimensions et leurs marges*. 1959.

[167] M. A. Stephens. Edf statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69(347):730–737, 1974.

[168] Hendrik Strobelt, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander M Rush. Seq2seq-vis: A visual debugging tool for sequence-to-sequence models. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):353–363, 2019.

[169] Hendrik Strobelt, Sebastian Gehrmann, Hanspeter Pfister, and Alexander M Rush. Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):667–676, 2018.

[170] Irene Sturm, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Interpretable deep neural networks for single-trial EEG classification. *Journal of Neuroscience Methods*, 274:141 – 145, 2016.

[171] A. Tewari, M. J. Giering, and A. Raghunathan. Parametric characterization of multimodal distributions with non-gaussian modes. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 286–292, Dec 2011.

[172] D. Thompson, J. A. Levine, J. C. Bennett, P. T. Bremer, A. Gyulassy, V. Pascucci, and P. P. Pbay. Analysis of large-scale scalar data using hixels. In *Large Data Analysis and Visualization (LDAV), 2011 IEEE Symposium on*, pages 23–30, 2011.

[173] V. Vishwanath, M. Hereld, and M. E. Papka. Toward simulation-time data analysis and i/o acceleration on leadership-class systems. In *2011 IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pages 9–14, 2011.

[174] Carl Vondrick, Aditya Khosla, Tomasz Malisiewicz, and Antonio Torralba. Hoggles: Visualizing object detection features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, 2013.

[175] Chaoli Wang, Hongfeng Yu, and Kwan-Liu Ma. Importance-driven time-varying data visualization. *IEEE Trans. on Vis. and Comp. Graphics*, 14(6):1547–1554, 2008.

[176] J. Wang, L. Gou, W. Zhang, H. Yang, and H. Shen. Deepvid: Deep visual interpretation and diagnosis for image classifiers via knowledge distillation. *IEEE Transactions on Visualization and Computer Graphics*, 25(6):2168–2180, June 2019.

[177] J. Wang, S. Hazarika, C. Li, and H. Shen. Visualization and visual analysis of ensemble data: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 25(9):2853–2872, Sep. 2019.

[178] Junpeng Wang, Liang Gou, Han-Wei Shen, and Hao Yang. Dqnviz: A visual analytics approach to understand deep q-networks. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):288–298, 2019.

[179] Junpeng Wang, Liang Gou, Hao Yang, and Han-Wei Shen. Ganviz: A visual analytics approach to understand the adversarial game. *IEEE Transactions on Visualization and Computer Graphics*, 24(6):1905–1917, 2018.

[180] Junpeng Wang, Xiaotong Liu, Han-Wei Shen, and Guang Lin. Multi-resolution climate ensemble parameter analysis with nested parallel coordinates plots. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):81–90, 2017.

[181] K. C. Wang, Kewei Lu, T. H. Wei, N. Shareef, and H. W. Shen. Statistical visualization and analysis of large data using a value-based spatial distribution. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pages 161–170, April 2017.

[182] Tzu-Hsuan Wei, Teng-Yok Lee, and Han-Wei Shen. Evaluating isosurfaces with level-set-based information maps. In *Proceedings of the 15th Eurographics Conference on Visualization*, EuroVis '13, pages 1–10. The Eurographs Association ; John Wiley ; Sons, Ltd., 2013.

[183] Ross T Whitaker, Mahsa Mirzargar, and Robert M Kirby. Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2713–2722, 2013.

[184] Brad Whitlock, Jean M. Favre, and Jeremy S. Meredith. Parallel in situ coupling of simulation with a fully featured visualization system. In *Proceedings of the 11th Eurographics Conference on Parallel Graphics and Visualization*, EGPGV '11, pages 101–109. Eurographics Association, 2011.

[185] P. C. Wong, H. W. Shen, C. R. Johnson, C. Chen, and R. B. Ross. The top 10 challenges in extreme-scale visual analytics. *IEEE Computer Graphics and Applications*, 32(4):63–67, July 2012.

[186] Pak Chung Wong and R. Daniel Bergeron. 30 years of multidimensional multivariate visualization. In *Scientific Visualization, Overviews, Methodologies, and Techniques*, pages 3–33, Washington, DC, USA, 1997. IEEE Computer Society.

[187] Kanit Wongsuphasawat, Daniel Smilkov, James Wexler, Jimbo Wilson, Dandelion Mané, Doug Fritz, Dilip Krishnan, Fernanda B Viégas, and Martin Wattenberg. Visualizing dataflow graphs of deep learning models in tensorflow. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):1–12, 2018.

[188] J. Woodring, J. Ahrens, J. Figg, J. Wendelberger, S. Habib, and K. Heitmann. In-situ sampling of a large-scale particle simulation for interactive visualization and analysis. In *Proceedings of the 13th Eurographics / IEEE - VGTC Conference on Visualization*, pages 1151–1160. Eurographics Association, 2011.

[189] J. Woodring, M. Petersen, A. Schmeißer, J. Patchett, J. Ahrens, and H. Hagen. In situ eddy analysis in a high-resolution ocean climate model. *IEEE Trans. on Vis. and Comp. Graphics*, 22(1):857–866, 2016.

[190] Cong Xie, Wei Xu, and Klaus Mueller. A visual analytics framework for the detection of anomalous call stack trees in high performance computing applications. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):215–224, 2019.

[191] Xiao Xie, Xiwen Cai, Junpei Zhou, Nan Cao, and Yingcai Wu. A semantic-based method for visualizing large image collections. *IEEE Transactions on Visualization and Computer Graphics*, 2018.

[192] Lijie Xu, Teng-Yok Lee, and Han-Wei Shen. An information-theoretic framework for flow visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1216–1224, 2010.

[193] Huiping Yan, Y Qian, Guang Lin, L Leung, Ben Yang, and Q Fu. Parametric sensitivity and calibration for the kainfritsch convective parameterization scheme in the wrf model. *Climate Research*, 59:135–147, 03 2014.

[194] Y. C. Ye, T. Neuroth, F. Sauer, K. L. Ma, G. Borghesi, A. Konduri, H. Kolla, and J. Chen. In situ generated probability distribution functions for interactive post hoc visualization and analysis. In *2016 IEEE 6th Symposium on Large Data Analysis and Visualization (LDAV)*, pages 65–74, Oct 2016.

[195] H. Yu, C. Wang, R. W. Grout, J. H. Chen, and K. L. Ma. In situ visualization for large-scale combustion simulations. *IEEE Computer Graphics and Applications*, 30(3):45–57, 2010.

[196] Jiawei Zhang, Yang Wang, Piero Molino, Lezhi Li, and David S Ebert. Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):364–373, 2019.

[197] Zhiguang Zhou, Linhao Meng, Cheng Tang, Ying Zhao, Zhiyong Guo, Miaoxin Hu, and Wei Chen. Visual abstraction of large scale geospatial origin-destination movement data. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):43–53, 2019.