Efficient Large-Scale Multivariate Data Analytics and Visualization using Statistical Approaches

Kaushik Raj V. Nadar

April 2025

MTH700 Project Report

Roll No.: 208160499

Under the Guidance of

Prof. Soumya Dutta (Computer Science & Engineering), and

Prof. Satya Prakash Singh (Mathematics & Statistics),

Indian Institute of Technology Kanpur

Soumya Dutta.

April 11, 2025

Satya

Declaration

I hereby declare that the work presented in the project report entitled **"Efficient Large-Scale Multivariate Data Analytics and Visualization using Statistical Approaches"** is written by me in my own words and contains my own or borrowed ideas. At places, where ideas and words are borrowed from other sources, proper references and acknowledgements, as applicable, have been provided. To the best of my knowledge this work does not emanate from or resemble work created by person(s) other than those mentioned and acknowledged herein.

Name and Signature: Kaushik Raj Vaikuntaraman Nadar

Date: April 11, 2025 

Abstract

Scientific simulations such as those used in climate modeling and fluid dynamics routinely generate high-dimensional, multivariate datasets at massive spatial and temporal resolutions, often resulting in terabytes of output. Due to practical constraints in storage bandwidth and computational overhead, storing raw simulation data in full fidelity is increasingly infeasible. In situ summarization—where the simulation domain is partitioned and compact statistical summaries are computed per block during runtime—has emerged as a viable solution, enabling significant data reduction while preserving essential distributional characteristics for post hoc analysis. Yet, conventional summarization techniques face critical limitations: multivariate histograms scale poorly in high dimensions, and Gaussian Mixture Models, though compact, are computationally intensive to fit and inflexible in marginal modeling. Copula-based methods offer a principled alternative by separating the modeling of univariate marginals from inter-variable dependencies. This allows for tailored marginal distributions per variable and a flexible copula function to capture their joint structure. Building on the CoDDA framework, which uses Gaussian copulas and arbitrary marginals, this work expands the methodological space in two key directions. First, we incorporate non-Gaussian copulas—such as the Clayton copula—to better capture tail dependencies and asymmetric correlations not addressed by linear models. Second, we enrich the repertoire of marginal models to include parametric forms like Gamma, Gaussian, and truncated Gaussians, in addition to empirical histograms. All components are estimated using statistical techniques such as maximum likelihood or EM, avoiding black-box models in favor of transparent, interpretable summaries. We apply our method to the Hurricane Isabel simulation dataset—a 3D climate dataset with 14 scalar fields over a 250×250×50 grid—and show that our extended copula framework achieves superior reconstruction fidelity (lower NRMSE, higher PSNR) compared to both independent and standard Gaussian approaches, while maintaining a compact storage footprint and runtime efficiency suitable for high-performance computing environments.

Contents

1 Introduction

Large-scale scientific simulations (e.g., climate models, fluid dynamics) produce multivariate data at extremely high resolution (often terabytes or more). Storing and analyzing such massive data in full detail is impractical due to storage I/O bottlenecks and computational overhead. A proven strategy is in situ data summarization: partition the simulation domain and store compact statistical summaries per block instead of raw data. These summaries, capturing distributional information, drastically reduce data size and enable faster post-hoc analysis on commodity hardware. However, standard multivariate summaries have limitations. For example, a naive multivariate histogram suffers exponential growth in bins with dimensionality, often yielding storage sizes comparable to raw data. Gaussian Mixture Models (GMMs) are more compact, but fitting a full multivariate GMM via Expectation-Maximization is computationally expensive as dimension increases. Moreover, fixed-form multivariate models lack flexibility – they assume each variable's marginal follows a certain form (e.g. fixed number of mixture components), even if a simpler distribution would suffice.

Copula-based modeling addresses these challenges by decoupling multivariate distribution into two parts: (1) arbitrary univariate marginal distributions for each variable, and (2) a copula function that models their dependency structure. A copula C "couples" the marginals into a joint distribution. This allows great flexibility: one can model each variable with a distribution suited to its shape (normal, gamma, etc.) and choose a copula to capture inter-variable dependencies. Prior work introduced CoDDA (Copula-based Distribution Driven Analysis) as a framework leveraging Gaussian copulas for in situ multivariate summarization[1]. CoDDA demonstrated that even using a Gaussian copula (which captures linear correlation) with flexible marginals yields accurate reconstructions and query results for complex simulation data.

In this paper, we extend the copula-based approach to improve efficiency and fidelity in summarizing large-scale multivariate data. Specifically, we incorporate alternative copula families (e.g., Clayton) to capture tail dependencies beyond what a Gaussian copula can represent. We also enrich the set of permissible univariate marginal models,

including not only histograms as in CoDDA, but also Gamma distributions for skewed positive variables, Gaussian, and even truncated Gaussians for naturally bounded variables. Only statistical methods for parameter estimation (e.g. maximum likelihood, EM for mixtures) are employed – no black-box machine learning model is used for the summarization itself. The goal is a flexible, interpretable statistical summary that balances modeling accuracy, storage footprint, and computational efficiency. We focus on a case study with the Hurricane Isabel dataset (from the IEEE Visualization 2004 Contest), a 3D simulation of a hurricane with 14 physical variables at resolution 250×250×50. We demonstrate that our extended copula-based framework yields lower error (as measured by Normalized RMSE and higher Peak Signal-to-Noise Ratio) compared to baselines like Independent approach, while remaining feasible for in situ deployment. We also show visual comparisons of reconstructed scalar fields.

2 Fundamentals of Copula Theory

2.1 Definition

A copula $C(u_1, \ldots, u_d)$ is a d-dimensional cumulative distribution function defined on the unit hypercube $[0,1]^d$ with uniform$(0,1)$ marginals[2]. By definition, $C(1, \ldots, 1) = 1$, and each marginal $C(u_1, \ldots, u_{i-1}, 1, u_{i+1}, \ldots, 1) = u_i$. Sklar's Theorem (1959)[3] formally states that for any set of continuous random variables $X_1, \ldots, X_d$ with joint CDF $F(x_1, \ldots, x_d)$ and marginal CDFs $F_i(x_i)$, there exists a copula C such that :

$$F(x_1, \ldots, x_d) = C\big(F_1(x_1),\, F_2(x_2), \ldots, F_d(x_d)\big).$$

If we denote $u_i = F_i(x_i)$ (which is uniform on $[0,1]$ by probability integral transform), then $C(u_1, \ldots, u_d) = F(x_1, \ldots, x_d)$. Differentiating Sklar's equation gives the relationship for probability density functions (PDFs) :

$$f(x_1, \ldots, x_d) = c(u_1, \ldots, u_d) \prod_{i=1}^{d} f_i(x_i),$$

where $f_i(x_i)$ are the marginal PDFs and $c(u_1, \ldots, u_d) = \frac{\partial^d C(u_1,\ldots,u_d)}{\partial u_1 \cdots \partial u_d}$ is the copula density. Thus the joint density factorizes into independent marginals times the copula density. Intuitively, the copula density $c(u_1, \ldots, u_d)$ captures all the dependence between variables (it equals 1 if variables are independent, and deviates from 1 to represent correlation or more complex dependence).

2.2 Elliptical Copulas:

Gaussian Copula The Gaussian copula is derived from the multivariate normal distribution. Given a correlation matrix $\rho \in \mathbb{R}^{d\times d}$, the Gaussian copula is defined as

$$C_\rho^G(u_1, \ldots, u_d) = \Phi_\rho\big(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d)\big),$$

where Φ^{-1} is the inverse CDF of a standard normal (applied to each u_i) and Φ_ρ is the CDF of a d-variate normal with zero mean and covariance ρ. In practice, ρ is the Pearson correlation matrix of the variables after a suitable marginal transform (e.g. using the observed data ranks). The meta-Gaussian distribution constructed by applying arbitrary marginals F_i^{-1} to a Gaussian copula sample is sometimes called a Gaussian copula model. Gaussian copulas capture linear correlation but no tail dependence – extreme values of X_i and X_j are no more likely to co-occur than normal assumption predicts.

Sampling a Gaussian copula is straightforward:

1. Generate d i.i.d. uniform random numbers $\{U_1^*, U_2^*, \ldots, U_d^*\} \sim \mathcal{U}(0,1)$ using a multiplicative congruential generator (MCG).

2. Convert the uniforms into independent standard normal variables $Z = (Z_1, Z_2, \ldots, Z_d)^\top$, using the inverse CDF method: $Z_i = \Phi^{-1}(U_i^*)$ or use Box–Muller method for pairs.

3. Let $\Sigma = \rho$ be a positive-definite correlation matrix. Compute its Cholesky factor L such that $\Sigma = LL^\top$. Then transform the i.i.d. normals to correlated normals:

$$V = LZ \sim \mathcal{N}(0, LIL^\top) \equiv \mathcal{N}(0, \Sigma).$$

4. Map each V_i to a uniform via the standard normal CDF:

$$U_i = \Phi(V_i) \sim \mathcal{U}(0,1), \quad \text{for } i = 1, \ldots, d.$$

5. Apply the inverse marginal transforms to each coordinate:

$$X_i = F_i^{-1}(U_i), \quad \text{for } i = 1, \ldots, d.$$

Output the sample $(X_1, X_2, \ldots, X_d)$ as a realization from the target multivariate distribution with marginals F_i and dependence structure induced by Σ.

Student–t Copula: The t-copula is an elliptical copula like the Gaussian, but with an additional degree of freedom parameter ν that controls tail thickness. It is derived from the multivariate Student–t distribution. If T_ν^{-1} is the inverse CDF of univariate t with ν degrees of freedom and $T_{\nu,\rho}$ the CDF of multivariate t_ν with correlation ρ, the t-copula is

$$C^t_{\rho,\nu}(u_1, \ldots, u_d) = T_{\nu,\rho}\big(T_\nu^{-1}(u_1),\, \ldots,\, T_\nu^{-1}(u_d)\big).$$

As $\nu \to \infty$, T_ν approaches Φ and the t-copula converges to the Gaussian copula. For finite ν, the t-copula exhibits tail dependence: both left and right tails of the variables are correlated (extreme co-movements are more likely than under a Gaussian copula). The t-copula is thus useful for modeling variables with joint extreme events. Its parameters are ρ (correlation matrix, which can be estimated via rank correlation as in Gaussian case) and ν. A common approach is to fit ν by maximizing the copula log-likelihood or matching empirical tail dependence. Sampling is analogous to Gaussian: sample t-distributed vector $z \sim t_\nu(0, \rho)$, then $u_i = T_\nu(z_i)$ and $x_i = F_i^{-1}(u_i)$. We don't discuss t-copula since, we haven't implemented and generated results for the same.

2.3 Archimedean Copulas

Archimedean copulas are defined via a generator function $\psi : [0,1] \to [0,\infty]$ that is convex, decreasing, with $\psi(1) = 0$. The copula is $C(u_1, \ldots, u_d) = \psi^{-1}(\psi(u_1) + \cdots + \psi(u_d))$. They provide simple closed forms with a single dependence parameter. Two popular ones are:

Clayton Copula: $C_\theta^{Clayton}(u_1, \ldots, u_d) = \max\left(u_1^{-\theta} + \cdots + u_d^{-\theta} - (d-1), 0\right)^{-1/\theta}$, for $\theta > 0$. In the bivariate case ($d = 2$), this simplifies to $C(u,v) = \left(u^{-\theta} + v^{-\theta} - 1\right)^{-1/\theta}$. Clayton's copula exhibits strong left-tail dependence – it favors joint lower extremes (simultaneously small u_i, corresponding to small values in original variables). The dependence parameter θ is related to Kendall's tau by $\tau = \theta/(\theta + 2)$; thus one can estimate θ from data via the empirical τ.

Sampling a Clayton copula can be done via the Marshall-Olkin method: e.g., for $d = 2$, one can draw $U \sim \text{Uniform}(0,1)$ and $W \sim \text{Exp}(1/\theta)$, then set $u_1 = (1+W)^{-1/\theta}$ and $u_2 = \left(1 + W(1-U)^{-\theta}\right)^{-1/\theta}$. This yields $(u_1, u_2) \sim C_\theta^{Clayton}$. (For larger d, one can sample a common $W \sim \text{Gamma}(d^{-1/\theta}, 1)$ and i.i.d. standard exponentials E_i, then set $u_i = (1 + WE_i)^{-1/\theta}$ for $i = 1, \ldots, d$.)

Gumbel Copula: $C_\theta^{\text{Gumbel}}(u_1, \ldots, u_d) = \exp\left\{-\left[(-\ln u_1)^\theta + \cdots + (-\ln u_d)^\theta\right]^{\frac{1}{\theta}}\right\}$, for $\theta \geq 1$. In $d = 2$, $C(u,v) = \exp -[(-\ln u)^\theta + (-\ln v)^\theta]^{1/\theta}$. Gumbel's copula exhibits right-tail dependence – it emphasizes joint large values of u_i (i.e. upper extremes of original variables). Its θ relates to Kendall's tau as $\tau = 1 - 1/\theta$. Sampling Gumbel copulas can be done using the relationship with minimum of i.i.d. exponential variables or via numerical inversion methods. Both Clayton and Gumbel are asymmetric: they capture one-side tail coupling. In contrast, the t-copula (and Gaussian, which is a special case with zero tail dependence) are symmetric in tail behavior. However, we have not completely implemented Gumbel in this project due to limitation of time.

2.4 Mixture Copula

Gaussian Mixture Copula Assume we have a GMM PDF $\psi(\mathbf{z};\Theta)$ in $\mathbb{R}^d$ with parameters $\Theta = \{\alpha^l, \boldsymbol{\mu}^l, \boldsymbol{\Sigma}^l\}_{l=1}^m$, meaning

$$\psi(\mathbf{z};\Theta) = \sum_{l=1}^{m} \alpha^l \mathcal{N}\big(\mathbf{z};\boldsymbol{\mu}^l,\boldsymbol{\Sigma}^l\big),$$

where each component has weight α^l, mean $\boldsymbol{\mu}^l$, and covariance $\boldsymbol{\Sigma}^l$. For each dimension $r \in \{1,\dots,d\}$, let $\psi_r(z_r;\Theta^r)$ be the univariate marginal PDF of the GMM, with Θ^r denoting the subset of parameters relevant to dimension r. Likewise, let $\Psi_r(z_r;\Theta^r) = \int_{-\infty}^{z_r}\psi_r(t;\Theta^r)\,dt$ be its marginal CDF, and Ψ_r^{-1} its corresponding quantile (inverse CDF) function[4].

Gaussian Mixture Copula Density A Gaussian Mixture Copula is then defined on $\mathbf{u} \in [0,1]^d$ (where each u_r is uniform on $[0,1]$) via:

$$\zeta(\mathbf{u};\Theta) = \frac{\psi\big(\Psi^{-1}(\mathbf{u});\,\Theta\big)}{\prod_{r=1}^{d}\psi_r\Big(\,\Psi_r^{-1}(u_r)\,;\,\Theta^r\Big)},$$

where: 1. $\Psi^{-1}(\mathbf{u}) \triangleq \big[\Psi_1^{-1}(u_1),\,\dots,\,\Psi_d^{-1}(u_d)\big]^{\mathsf{T}}$. 2. $\psi\big(\Psi^{-1}(\mathbf{u});\,\Theta\big)$ is simply the GMM PDF evaluated at the point whose r-th coordinate is $\Psi_r^{-1}(u_r)$. 3. The denominator factors out the product of the univariate GMM marginals ψ_r, evaluated at those same coordinates, ensuring each dimension is mapped to Uniform(0,1).

Hence, $\zeta(\mathbf{u};\Theta)$ is a valid copula density on $[0,1]^d$ encoding the dependence structure of the original GMM, while each coordinate u_r is uniform marginally.

Fitting the GMC Let $\{u^{(i)}\}i = 1^n \subseteq [0,1]^d$ be your data points (each coordinate is uniform on $[0,1]$). In practice, you typically start with real-valued data $\{x^{(i)}\}_{i=1}^n \subseteq \mathbb{R}^d$ and transform each dimension to $[0,1]$ using pre-chosen or pre-fitted marginal CDFs $F_r(x_r)$. Once you have these $\{u^{(i)}\}$, you fit the GMC by finding parameters $\Theta = \{\alpha^l, \boldsymbol{\mu}^l, \boldsymbol{\Sigma}^l\}_{l=1}^m$ that maximize the log-likelihood

$$\ell(\Theta) \;=\; \sum_{i=1}^{n} \log\Big[\zeta\big(u^{(i)};\,\Theta\big)\Big] \quad \text{where} \quad \zeta(\mathbf{u};\Theta) \;=\; \frac{\psi\big(\Psi^{-1}(\mathbf{u});\,\Theta\big)}{\prod_{r=1}^{d}\,\psi_r\Big(\,\Psi_r^{-1}(u_r)\,;\,\Theta^r\Big)}.$$

- Inverse of the marginal GMM CDF $\Psi_r^{-1}(\cdot)$ does not have a closed form. You typically solve for z_r in $u_r \;=\; \Psi_r(z_r;\,\Theta^r)$ using a root-finding method (e.g., Secant or Brent). This step is repeated for each dimension and each data point.

- Optimization can be done via: A gradient-based approach (e.g., Adam or LBFGS). You need either numerical or "semi-analytical" derivatives – i.e., partial derivatives of Ψ_r^{-1} can be derived by hand and inserted into an autodiff system or a Generalized EM (GEM) algorithm: though no closed-form M-step exists, one can form the auxiliary function (the "Q-function") and use a gradient-based partial maximization each iteration. This ensures a monotonic ascent in the log-likelihood.

- Identifiability Constraint: Gaussian Mixture Copulas have an inherent non-identifiability (scaling of means and covariances can leave the copula unchanged). To obtain a unique solution, you can enforce mild constraints or regularizing priors (e.g., constraints on the marginal means/variances across mixture components) to lock down that extraneous freedom.

- Number of components to be used in the GMC is decided based on Bayesian Information Criterion (BIC).

$$\text{BIC} = -2 \cdot \log L(\hat{\Theta}) + k \cdot \log n$$

where:

$L(\hat{\Theta})$ = Maximum likelihood of the GMM under estimated parameters $\hat{\Theta}$

k = Number of free parameters in the model

n = Number of data points

$$k = m \cdot \left(d + \frac{d(d+1)}{2} \right) + (m-1)$$

where:

- d parameters for the mean vector of each component
- $\frac{d(d+1)}{2}$ parameters for the covariance matrix (symmetric)
- $(m-1)$ parameters for the mixture weights (they sum to 1)

The BIC is calculated for 1-5 components of GMC model and the lowest BIC model is selected.

Sampling from the Fitted GMC Once you have the final parameter set $\hat{\Theta}$, you can draw uniform samples $\mathbf{u} \in [0,1]^d$ from the GMC via the following:

1. Sample $\mathbf{z}$ from the underlying GMM:

$$\mathbf{z} \sim \psi(\mathbf{z}; \hat{\Theta}) = \sum_{l=1}^{m} \alpha^l \mathcal{N}\big(\mathbf{z}; \boldsymbol{\mu}^l, \boldsymbol{\Sigma}^l\big).$$

2. Convert to copula space by computing $u_r = \Psi_r\big(z_r; \hat{\Theta}^r\big)$ for $r = 1, \ldots, d$. Because Ψ_r is each dimension's marginal CDF, each u_r is uniform on $[0,1]$. Thus $\mathbf{u}$ is a draw from $\zeta(\mathbf{u}; \hat{\Theta})$.

If the ultimate goal is a multivariate real-valued sample $\mathbf{x}$ with user-chosen marginals F_r, then once you have $\mathbf{u}$ above, apply $x_r = F_r^{-1}(u_r)$ for $r = 1, \ldots, d$. This yields samples $\mathbf{x}$ having the same dependence structure as your Gaussian Mixture Copula but with (potentially different) margins F_r.

2.5 Marginal Distributions

Any marginal distrbution having a closed form CDF can be used as marginal for copula modelling. We are using the following marginals since these are easier to fit, and the other models (eg. Beta, loglaplace), faced some issue during optimisation.

Uniform (Univariate) A Uniform distribution on $[a, b]$ has the simplest possible PDF: constant over the specified interval. Its PDF is

$$f_{\text{Uniform}}(x;\, a, b) \;=\; \begin{cases} \frac{1}{b-a} & \text{if } a \le x \le b, \\ 0 & \text{otherwise.} \end{cases}$$

It has the parameters: a (lower bound), b (upper bound) with $b > a$, and Support: $[a, b]$.

In our implementation, we often denote the location parameter as a and a scale parameter as $(b - a)$.

Gaussian (Normal) Univariate The standard Gaussian (Normal) distribution with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 > 0$ has PDF:

$$f_{\text{Gauss}}(x;\, \mu, \sigma) \;=\; \frac{1}{\sqrt{2\pi}\,\sigma}\; \exp\Big(-\frac{(\,x-\mu\,)^2}{2\,\sigma^2}\Big).$$

Parameters are: μ (location), $\sigma > 0$ (scale), and Support: $(-\infty, \infty)$.

Truncated Gaussian (Normal) A Truncated Gaussian distribution is a Normal distribution $N(\mu, \sigma^2)$ restricted to lie within the interval $[a, b]$. The PDF is given by:

$$f_{\text{TGauss}}(x;\, \mu, \sigma, a, b) \;=\; \begin{cases} \dfrac{\phi\Big(\dfrac{x-\mu}{\sigma}\Big)}{\sigma\,\Big[\Phi\Big(\dfrac{b-\mu}{\sigma}\Big) \;-\; \Phi\Big(\dfrac{a-\mu}{\sigma}\Big)\Big]} & \text{if } a < x < b, \\ 0 & \text{otherwise,} \end{cases}$$

where $\phi(z) = \frac{1}{\sqrt{2\pi}}\, e^{-\,z^2/2}$ and $\Phi(z) = \int_{-\infty}^{z} \phi(t)\, dt$ are the standard Normal PDF and CDF, respectively. It has Parameters: μ (location), $\sigma > 0$ (scale), and truncation bounds $a < b$, and Support: $[a, b] \subseteq (-\infty, \infty)$.

Gamma (Univariate) The Gamma distribution Gamma(α, β) is typically parameterized by a shape $\alpha > 0$ and scale $\beta > 0$. Its PDF is:

$$f_{\Gamma}(x;\,\alpha,\beta) \;=\; \begin{cases} \frac{x^{\alpha-1}\,\exp\!\left(-\frac{x}{\beta}\right)}{\beta^{\alpha}\,\Gamma(\alpha)} & x > 0, \\ \\ 0 & \text{otherwise,} \end{cases}$$

where $\Gamma(\alpha)$ is the Gamma function: $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1}e^{-t}\,dt$. Parameters are: α (shape), β (scale), and Support: $(0, \infty)$.

Histogram (Piecewise-Constant) A Histogram is a simple empirical approximation of a PDF, constructed by dividing the variable's range into bins. Suppose we have n bins with edges $\{e_0, e_1, \ldots, e_n\}$, where $e_0 < a < e_1 < \cdots < e_{n-1} < e_n = b$. Let $\Delta_i = e_i - e_{i-1}$ be the width of bin i. If the bin's count is c_i out of a total sample size N, then the piecewise-constant PDF is:

$$f_{\text{Hist}}(x) \;=\; \begin{cases} \frac{c_i}{N\,\Delta_i} & \text{if } e_{i-1} \le x < e_i, \\ \\ 0 & \text{otherwise.} \end{cases}$$

Parameters are: A set of bin edges $\{e_0, \ldots, e_n\}$ and the count in each bin c_i, and Support: $[e_0, e_n]$. In a copula context, one uses the normalized heights of these bins to define an empirical CDF for marginal transformations, though it's more common to store cumulative counts or to do a midpoint-based approximation.

2.6 From Copula to Joint Samples

Once a copula C and marginals F_i are determined for a block, one can generate synthetic data points $(x_1, \ldots, x_d)$ as follows : (1) Sample $u = (u_1, \ldots, u_d) \sim C$ on $[0,1]^d$ (using the methods above for the chosen copula). (2) Apply the inverse marginal CDFs to transform each u_i into $x_i = F_i^{-1}(u_i)$. This yields a sample consistent with the modeled joint distribution. Repeating this can populate an arbitrary-resolution grid with samples for visualization . Copula-based sampling is central to how we reconstruct approximate fields for visualization and to how we answer range queries (by Monte Carlo integration; see Methodology). Importantly, any error in the summary (e.g., imperfect capture

of dependencies or marginal shape) will manifest as differences between these sampled reconstructions and the original data. We quantify these differences using metrics like Normalized RMSE (NRMSE) and Peak Signal-to-Noise Ratio (PSNR) in our Results.

3 Methodology

3.1 Dataset

The Hurricane Isabel dataset is a high-resolution simulation output capturing the complex atmospheric dynamics of Hurricane Isabel, a Category 5 hurricane that occurred in September 2003. Generated using the Weather Research and Forecasting (WRF) model, the dataset was provided for the IEEE Visualization Contest 2004 and has since become a standard benchmark in the visualization and scientific data analysis communities.

The dataset represents a volumetric grid of dimensions $500 \times 500 \times 100$ (however, we use a subsampled $250 \times 250 \times 50$ version), corresponding to a three-dimensional spatial domain with fine-grained resolution in the horizontal and vertical directions. At each grid point, multiple meteorological variables are recorded, capturing key physical phenomena such as moisture content, temperature, pressure, and wind velocities. These multivariate fields characterize the structure, intensity, and evolution of the hurricane system at a specific simulation timestep.

Due to its size, complexity, and scientific relevance, the Isabel dataset serves as a canonical testbed for evaluating techniques in multivariate data summarization, uncertainty quantification, in situ analysis, and copula-based modeling, especially in the context of high-performance computing and large-scale scientific visualization.

3.2 Overall Workflow

During the simulation run, at each output interval we partition the domain into blocks of a chosen size (e.g. 5^3 subregions for a 3D grid) . For each block, we perform: (a) Univariate Distribution Fitting for each variable (and spatial coordinates), and (b) Copula Fitting for the joint dependency. The resulting parameters constitute the multivariate summary

Table 1: Summary of Hurricane Isabel Dataset Variables

Variable	**Description**	**Range**
QCLOUD	Cloud moisture mixing ratio (kg water/kg dry air)	[0.00000, 0.00332]
QGRAUP	Graupel mixing ratio (hail, sleet, snow pellets)	[0.00000, 0.01638]
QICE	Cloud ice mixing ratio	[0.00000, 0.00099]
QSNOW	Snow mixing ratio	[0.00000, 0.00135]
QVAPOR	Water vapor mixing ratio	[0.00000, 0.02368]
CLOUD	Total cloud moisture (QCLOUD + QICE)	[0.00000, 0.00332]
PRECIP	Total precipitation (QGRAUP + QRAIN + QSNOW)	[0.00000, 0.01672]
QRAIN	Rainwater mixing ratio	[0.00000, 0.00742]
P	Pressure (Pa; deviation from base state)	[–5471.85791, 3225.42578]
TC	Temperature (°C)	[–83.00402, 31.51576]
U	X-direction wind speed (west → east)	[–79.47297, 85.17703]
V	Y-direction wind speed (south → north)	[–76.03391, 82.95293]
W	Z-direction wind speed (upward positive)	[–9.06026, 28.61434]
Velocity	Magnitude of wind vector $\sqrt{U^2 + V^2 + W^2}$	[0.00000, 76.39613]

Grid Dimensions: $250 \times 250 \times 50$ **Total Variables:** 14

for that block . These summaries are then written to storage (a tiny fraction of full data size). Later, for analysis, we load the summaries and use them to either reconstruct approximate data fields or directly answer queries. We emphasize that our extensions remain within this general workflow, with improvements in the fitting procedures.

Each spatial coordinate (X, Y, Z) in a block is treated as a special variable with uniform distribution over that block's extent. In our implementation, we simply record the block's min and max along each axis (or equivalently, the block index within the grid) to represent the spatial variable's distribution – assuming data points are uniformly distributed in space within the block. Including spatial coords in the joint model means each sample drawn from the copula summary carries a spatial position, which preserves spatial context in our reconstructions.

After this step, each block has d univariate models (where $d = V + 3$, for V physical variables plus 3 spatial). For example, in Hurricane Isabel $V = 14$ physical variables, so $d = 17$ including (x, y, z) spatial. We denote these marginal CDFs as $F_1, \ldots, F_d$.

3.3 Copula Dependency Modeling

Next, we model the joint dependence of the d variables in the block using a copula $C(u_1, \ldots, u_d; \Theta)$. Here Θ denotes the copula parameters (correlation matrix, etc., depending on family). Gaussian copula is computationally light and is effective when block size is small. We compute the $d \times d$ sample correlation matrix $\hat{\rho}$ of the variables (after transforming each sample value x_i to $u_i = F_i(x_i)$ to get uniform marginals). Only the correlations need storage; for $d = 17$, there are $\frac{17\times16}{2} = 136$ unique correlations. We store them in a compressed format. In practice, we found that for small block sizes like 5^3, the empirical correlation estimates are robust (each block has 125). The Gaussian copula parameter fitting is thus $\mathrm{O}(d^2)$ per block for correlation calculation.

If we detect significant asymmetry in tail dependence (for instance, one variable's lower tail correlates with others' lower tails, but upper tail doesn't), we consider an Archimedean copula. Specifically, if lower-tail dependence is stronger (we compare lower tail dependence coefficients), we fit a Clayton copula by solving $\hat{\tau} = \theta/(\theta + 2)$ for θ using the average Kendall's tau of variable pairs that seem lower-tail dependent. Similarly, for upper-tail,

We have also implemented the gaussian mixture copula owing to its superior capabilities to capture multiple covariance structures among the variables. However, it is highly expensive on resources and storage, since stores the means, covariances and the weights of each component in the mixture model. Therefore, we didn't get enough time to collect results for this model.

We take into account the number of parameters: Archimedean copulas like Clayton and Gumbel each add just 1 parameter θ (which is very storage-friendly), but they assume a single parameter can describe the dependency of all variable pairs. In blocks with complex dependency (different pairs have different correlation), a one-parameter copula may be too restrictive and yield higher error. In such cases, a Student–t (which, like Gaussian, uses a full correlation matrix plus one ν) might be preferable despite more parameters.

Our framework can in principle use any copula. In this work, we implement Gaussian

copula fully, and provide Clayton and GMC as options for exploratory analysis on selected pairs of variables.

3.4 Parameter Estimation

For Gaussian copulas, $\Theta = \rho$ is obtained by sample correlation.

$$r_{X,Y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

For Clayton, we directly compute θ from Kendall's tau (since this method-of-moments is known to be efficient and yields a consistent estimator for Archimedean copulas). The overall cost of copula fitting is dominated by computing the correlation matrix ($O(d^2)$ per block). In our experiments, even for $d = 14$, this was under 0.01 seconds per block on average – negligible compared to the cost of simulating or even disk I/O. GMCs (Gaussian Mixture Copulas) do not have a closed form CDF, hence it requires numerical methods and sophistiated algorithms like EM(Expectation-Maximization) for estimation of parameters.

3.5 Storage and Data Structures

The binary file produced by `compress.py` stores a complete summary of the Copula model for in situ or post hoc analysis. Consider the example of Gaussian mixture copula (GMC), its structure is as follows:

- **Global Metadata (Header):**
  - $3 \times$ `uint16`: Original grid dimensions (d_x, d_y, d_z); total 6 bytes.
  - $1 \times$ `uint8`: Copula type code (e.g., Gaussian Mixture Copula = 3); 1 byte.
- **Per Block:**
  - $1 \times$ `uint8`: Number of mixture components m (i.e., number of Gaussians).
  - For each component $l = 1, \ldots, m$:

    - $1 \times$ `float32`: Component weight α^l.
  - For each component $l = 1, \ldots, m$:
    - $\frac{n(n-1)}{2} \times$ `float32`: Upper-triangular entries of the correlation matrix $\mathbf{\Sigma}^l$ (excluding the diagonal); this assumes n scalar variables.
  - For each scalar variable $r = 1, \ldots, n$ (i.e., marginal distributions):
    - $1 \times$ `uint8`: Marginal distribution type code.
    - If marginal is **Histogram**:
      - $1 \times$ `uint8`: Number of bins b.
      - $b \times$ `float32`: Bin densities.
      - $2 \times$ `float32`: Bin edges.
    - Otherwise (e.g., Gaussian, Gamma, Truncated Gaussian, etc.):
      - $k \times$ `float32`: Distribution parameters (e.g., mean/variance for Gaussian).

Each block thus occupies on the order of a few hundred bytes.

3.6 Post-hoc Reconstruction

Given the stored summary of a block, we can sample synthetic values as described earlier to reconstruct a field. A naive approach is to generate one sample per grid point in the block. However, that may introduce high-frequency noise because samples are random. CoDDA proposed an improved sampling-based visualization where many more samples N are generated in each block (e.g. $N = 2$–$5\times$ the number of grid points), then assigning the value to the closest grid point. We take it a step further approach: for each block and each variable, we generate N multivariate samples $(x_1, \ldots, x_d) \sim$ our fitted joint distribution. Each sample includes spatial coordinates (X, Y, Z) and a value for each physical variable. We then rasterize these N samples onto the block's grid (e.g., if block is $5\times5\times5$, we have 125 grid positions) by assigning the value to eight nearest grid points to produce the final scalar field. To do so, we compute a distance-weighted average of

sample values around each grid point. Essentially, we treat the samples as points and do a simple smoothing: each grid cell takes the average of nearby sample values of the variable, weighted by inverse distance (a form of scatter–gather interpolation). This smooths out the noise and yields a reconstructed scalar field for that block. We see that the copula-based reconstruction preserves the coherent structures (e.g., the eye of the hurricane's low pressure region) much better than the histogram approach, which loses correlation between neighboring cells (the multivariate histogram treats each block independently with random sampling, yielding salt-and-pepper noise). By incorporating spatial variables in the copula, our method respects that nearby points in space have related values.

Algorithm 1 Simplified Field Reconstruction from Copula Blocks

Require: List of copula blocks $\mathcal{C}$, Original grid shape (N_x, N_y, N_z), Target grid shape (T_x, T_y, T_z), Scalar variable list $\mathcal{V}$, and Samples per block S_b
Ensure: Reconstructed 3D field for each scalar variable at the target resolution

Initialize arrays S_v and W_v with zeros for each $v \in \mathcal{V}$
for each copula block in $\mathcal{C}$ **do**
  Sample S_b points from the block $\rightarrow \mathcal{D}$
  for each point in $\mathcal{D}$ **do**
    Convert (x, y, z) to new grid coordinates using proportional scaling
    Determine the base index (i, j, k) by flooring the rescaled coordinates
    for each of the $2 \times 2 \times 2$ neighboring grid points **do**
      Compute the distance from sample to the neighbor
      Compute weight as inverse of distance
      for each scalar variable v **do**
        Add weighted value to S_v at neighbor
        Add weight to W_v at neighbor
      end for
    end for
  end for
end for
for each scalar variable v **do**
  Normalize S_v by dividing by W_v
end for
return Reconstructed fields $\{S_v\}$

4 Results

Our technique is evaluated by comparing PSNR (Peak Signal to Noise Ratio), NRMSE (Normalised Root Mean Square), and Compression Ratio (Original Size/Compressed Size) across all the Copula models with various marginal distributions. Some Copula models may not have enough data for every field/marginals due to lack of time.

1. **Peak Signal-to-Noise Ratio (PSNR)**

   Let x_i and $\hat{x}_i$ be the original and reconstructed values, respectively, for a scalar field of length n. Let MAX denote the maximum possible value of the signal (or the dynamic range).

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2} \right)$$

   Normalized Root Mean Square Error (NRMSE)

   Let $x = \{x_1, x_2, \ldots, x_n\} and \hat{x} = \{\hat{x}_1, \hat{x}2, \ldots, \hat{x}n\}$ be the original and reconstructed values. Let $x\text{max}$ and $x\text{min}$ denote the maximum and minimum values of the original signal, respectively.

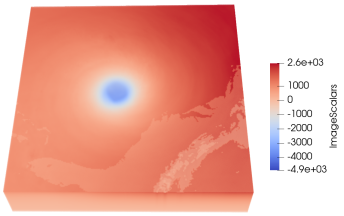
$$\text{NRMSE} = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{x}i)^2}}{x\text{max} - x_{\text{min}}}$$

   Compression Ratio (CR)

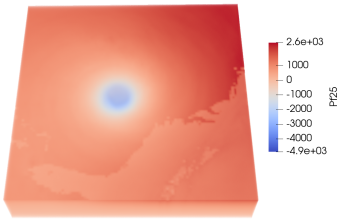
   Let S_{orig} be the size of the original data, and $S_{\text{compressed}}$ be the size after compression.

$$\text{Compression Ratio} = \frac{S_{\text{orig}}}{S_{\text{compressed}}}$$

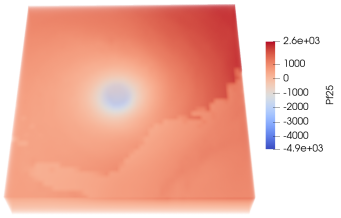
The analysis reveals that while the GMC copula outperforms other copulas in terms of reconstruction accuracy, as evidenced by higher PSNR values, this performance is achieved at the expense of a lower compression ratio due to the requirement for a higher number of parameters. The Histogram marginals are particularly noteworthy; they tend



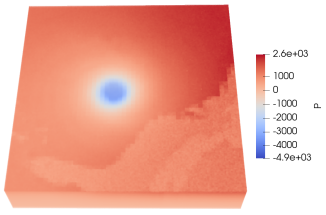
(a) Original Pressure field in (250x250x50) Resolution



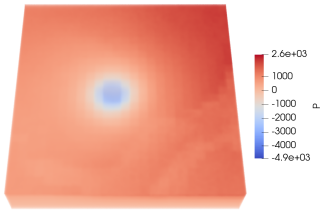
(b) Original Pressure field Subsampled to (100x100x20) Resolution



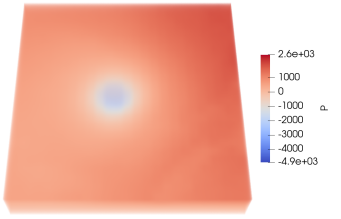
(c) Original Pressure field Subsampled to (50x50x10) Resolution



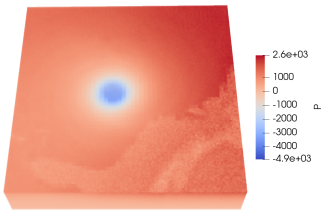
(d) Reconstructed to (250x250x50) Resolution using Truncated Normal



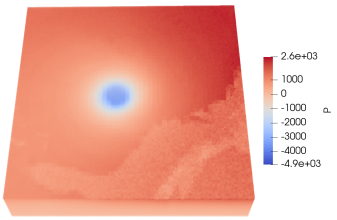
(e) Reconstructed to (100x100x20) Resolution using Truncated Normal



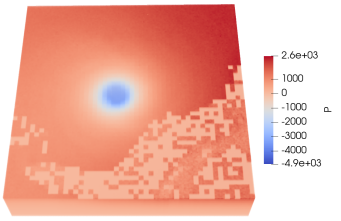
(f) Reconstructed to (50x50x10) Resolution using Truncated Normal



(g) Reconstructed to (250x250x50) Resolution using Histogram

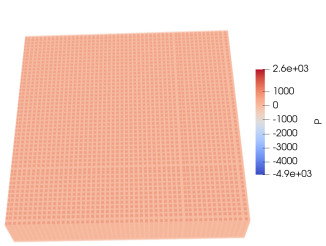


(h) Reconstructed to (250x250x50) Resolution using Gaussian

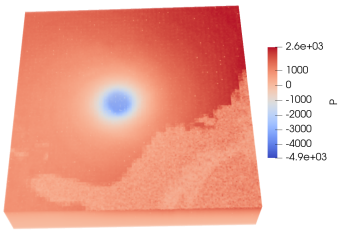


(i) Reconstructed to (250x250x50) Resolution using Gamma

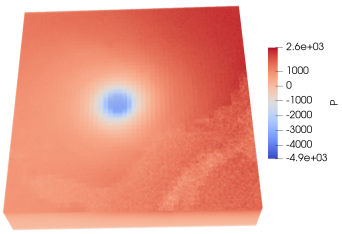
Figure 1: (a), (b), (c) Top images show the Original Pressure field. (d), (e), (f) show reconstructed Pressure field using Gaussian copula with Truncated Normal marginals. (g) uses Histogram as marginal distribution, (h) uses Gaussian distribution as marginals and (i) uses Gamma distribution as marginals, all with a Gaussian Copula and x,y,z as Uniform marginals and block size 5^3.



(a) Reconstructed to (250x250x50) Resolution using Clayton Copula

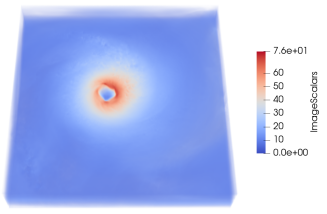


(b) Reconstructed to (250x250x50) Resolution using Gaussian Mixture Copula

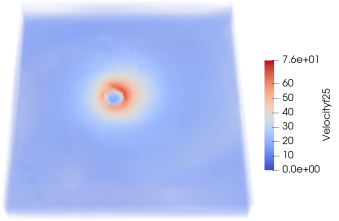


(c) Reconstructed to (250x250x50) Resolution using Independent Copula

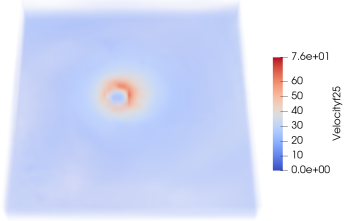
Figure 2: Similar to previous case, Pressure field is reconstructed. Block size 5^3 was used and Histogram was used as marginals for scalar variables, and uniform for x,y,z. It is quite easy to observe that Clayton does not perform well, GMC automatially chooses 1 component via BIC and performs well equivalent to single modal gaussian copula.



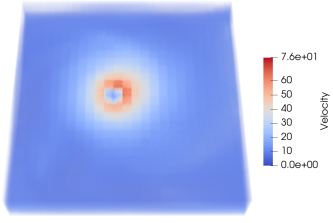
(a) Original Velocity field in (250x250x50) Resolution



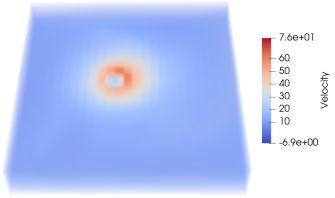
(b) Original Velocity field in (100x100x20) Resolution



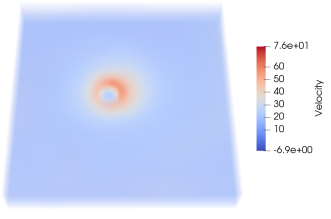
(c) Original Velocity field in (50x50x10) Resolution



(d) Reconstructed to (250x250x50) Resolution using Normal

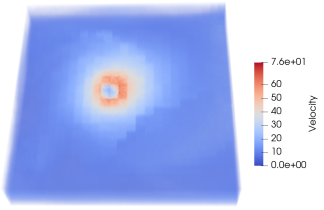


(e) Reconstructed to (100x100x20) Resolution using Normal

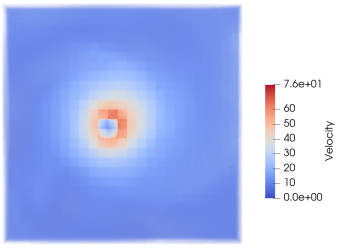


(f) Reconstructed to (50x50x10) Resolution using Normal

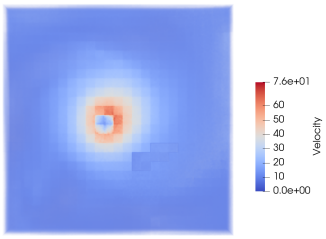
Figure 3: Velocity Field is modelled using Gaussian Copula, block size 10^3 was used and Normal distribution was used as marginals for scalar variables, and uniform for x,y,z.



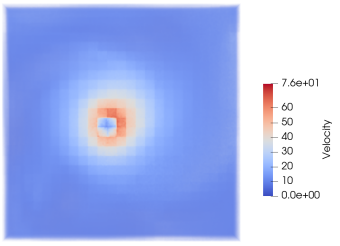
(a) (250x250x50) Resolution using Gaussian Copula with Uniform Marginals



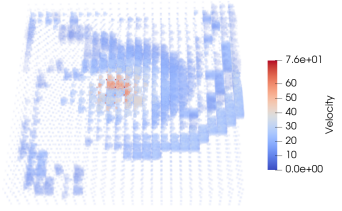
(b) (100x100x20) Resolution using Gaussian Copula with Truncated Normal Marginals



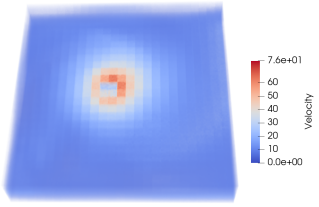
(c) (50x50x10) Resolution using Gaussian Copula with Gamma Marginals



(d) (250x250x50) Resolution using Gaussian Copula with Histogram Marginals

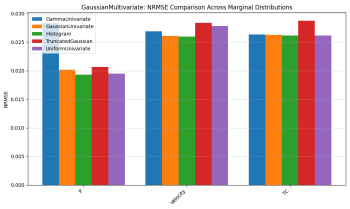


(e) (250x250x50) Resolution using Clayton Copula with Histogram Marginals

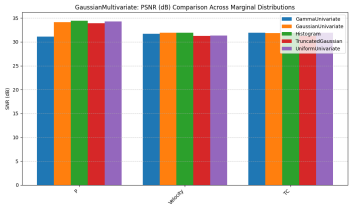


(f) (250x250x50) Resolution using Indpendent Copula with Histogram Marginals

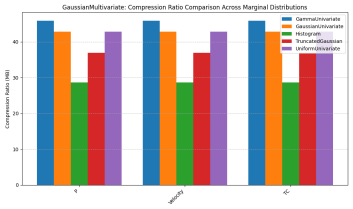
Figure 4: Velocity Field is modelled using various Copulas, block size 10^3 was used, and uniform marginals for x,y,z.



(a) Average NRMSE

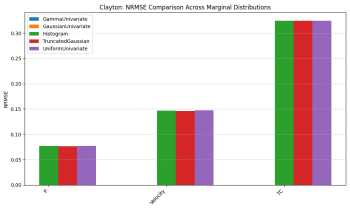


(b) Average PSNR

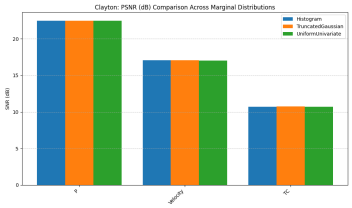


(c) Compression Ratio

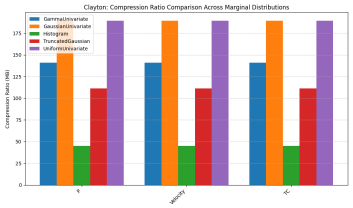
Figure 5: Plots across various scalar fields over different marginals for Gaussian Copula



(a) Average NRMSE

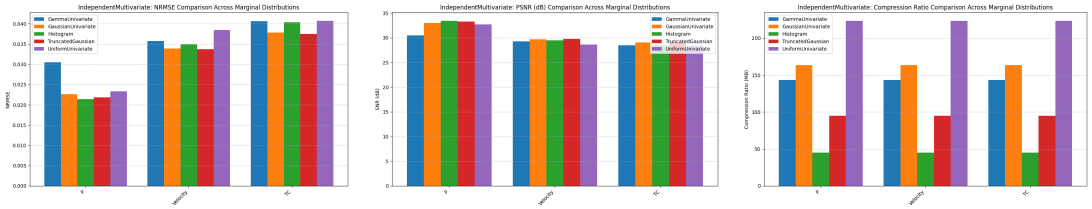


(b) Average PSNR



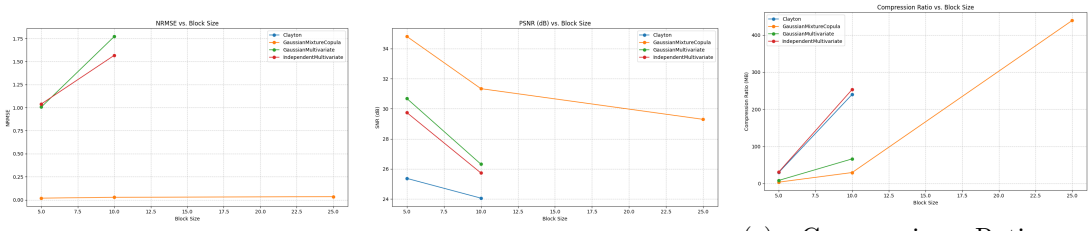
(c) Compression Ratio

Figure 6: Plots across various scalar fields over different marginals for Clayton Copula



(a) Average NRMSE (b) Average PSNR (c) Compression Ratio

Figure 7: Plots across various scalar fields over different marginals for Independent Copula



(a) NRMSE vs Block Sizes (b) PSNR vs Block Sizes (c) Compression Ratio vs Block Sizes

Figure 8: Plots of Average metrics vs the block sizes for different copulas

to deliver the highest PSNR and thus the highest accuracy in representing the data's nuances. However, their non-parametric nature, which involves maintaining detailed bin counts, results in a compression ratio that lags behind more parsimonious models. In contrast, both Gaussian and Truncated Gaussian marginals demonstrate an impressive balance between reconstruction fidelity and storage efficiency. Their parametric form, based on a limited set of parameters such as mean and variance, allows them to achieve performance levels nearly as high as those obtained using histograms, while significantly reducing the overhead in terms of parameter count.

Further, the intrinsic trade-offs underlying these modeling choices become evident when considering the complexity of the parameterization and the corresponding computational demands. The richness of the GMC copula and the detailed granularity of the Histogram approach naturally lead to more complex models that, while adept at capturing subtle dependencies and intricate data characteristics, also require more extensive storage resources and computational effort. On the other hand, Gaussian-based approaches impose structural assumptions—such as symmetry and specific tail behaviors—which, while less flexible than the histogram method, provide a practical and computationally

efficient means of achieving high-fidelity reconstructions.

It is also evident that Clayton Copula does not perform well compared to Gaussian Copula and Independent cases possibly due to some implementation error or because it only works with positive dependency ($\theta > 0$).

Ultimately, the optimal choice among these methods depends on the balance one wishes to strike between maximum accuracy and efficient compression. For applications where absolute reconstruction fidelity is paramount and storage limitations are less critical, the GMC copula and Histogram marginals emerge as superior choices. Conversely, when a more parsimonious model is desired—one that still preserves a high level of accuracy while offering a better compression ratio—the Gaussian and Truncated Gaussian marginals serve as appealing alternatives. This nuanced evaluation underscores the broader theme in data modeling of balancing complexity with efficiency, with the decision being guided by the specific demands of the intended application and the nature of the data at hand.

5 Conclusion

We presented an extended copula-based framework for efficient multivariate data summarization and visualization, building upon the CoDDA approach. By allowing alternative copula families (Gaussian, Clayton, GMC) and a rich set of univariate marginal models (Gaussians, Gamma, truncated Gaussians, histogram, etc.), our framework can adapt to a wide range of data distributions while decoupling the modeling of marginals and dependencies. Applied to the Hurricane Isabel simulation dataset, the method achieved high-fidelity representation of the data: crucial features and correlations were preserved with low error (NRMSE on the order of 1–2% of data range, corresponding to ¿35 dB PSNR in reconstructed fields). This is a significant improvement over traditional multivariate histograms or global GMMs, which either failed to compress or suffered high error. Moreover, our summaries enable advanced analysis using arbitrary resolution, essentially allowing scientists to ask complex multi-variable questions and get answers directly from

the stored summaries.

One key to our approach is its in situ viability – the computations required (fitting marginals, computing correlations) are fast and parallel-friendly, adding only modest overhead to simulation runtime. The payoff is a dramatic reduction in data that needs to be saved: for instance, instead of writing full 3D fields for many time steps, a simulation could write our summaries and still permit thorough post-hoc analysis. This can alleviate I/O bottlenecks and storage constraints in high-performance computing environments. The storage footprint of our method, while sometimes larger than raw for a single time, is still far smaller than naive approaches when considering multiple time steps or ensembles, and it is tunable by adjusting block size or model complexity.

Our copula-based modeling proves to be a powerful tool for multivariate data summarization. It respects the diversity of each variable's distribution and the intricacies of their inter-dependence by using the appropriate statistical building blocks (marginals and copulas). The result is a compact yet expressive summary that serves as a surrogate for the original data in many tasks. This approach merges concepts from statistics (copulas, parametric fitting) with the needs of large-scale data visualization (fast query, interactive exploration), exemplifying an interdisciplinary solution to data overload. We envision this method being integrated into simulation workflows, enabling scientists to effectively "take the data's distribution with them" instead of petabytes of raw data. The ability to visualize and analyze phenomena like Hurricane Isabel's evolution using only a few hundred megabytes of summary data is a testament to the efficacy of statistical data modeling in the era of exascale simulations.

6 Code Availability

The python code for the project is available at:

`https://github.com/kaushik3012/codda`

References

[1] Subhashis Hazarika, Soumya Dutta, Han-Wei Shen, and Jen-Ping 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, 2019.

[2] Thorsten Schmidt. Coping with copulas. *Copulas - From Theory to Application in Finance*, 01 2007.

[3] Abe Sklar. Fonctions de répartition à n dimensions et leurs marges. *Publ. Inst. Statist. Univ. Paris*, 8:229–231, 1959.

[4] Ashutosh Tewari. On the estimation of Gaussian mixture copula models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 34090–34104. PMLR, 23–29 Jul 2023.