Critical Point Extraction from Multivariate Functional Approximation

Guanqun Ma *
University of Utah

David Lenz †
Argonne National Laboratory

Tom Peterka ‡
Argonne National Laboratory

Hanqi Guo §
Ohio State University

Bei Wang ¶
University of Utah

ABSTRACT

Advances in high-performance computing require new ways to represent large-scale scientific data to support data storage, data transfers, and data analysis within scientific workflows. Multivariate functional approximation (MFA) has recently emerged as a new continuous meshless representation that approximates raw discrete data with a set of piecewise smooth functions. An MFA model of data thus offers a compact representation and supports high-order evaluation of values and derivatives anywhere in the domain. In this paper, we present CPE-MFA, the first critical point extraction framework designed for MFA models of large-scale, high-dimensional data. CPE-MFA extracts critical points directly from an MFA model without the need for discretization or resampling. This is the first step toward enabling continuous implicit models such as MFA to support topological data analysis at scale.

Index Terms: Multivariate functional approximation, critical points, topological data analysis, implicit models

1 INTRODUCTION

Advances in high-performance computing (HPC) require new ways to represent large-scale scientific data to support data storage, data transfers, and data analysis. To that end, there has been a growing interest in replacing discrete data with continuous, high-order, and differentiable representations, such as functional models and implicit neural networks [50, 51, 34, 37], to enhance scientific workflows.

Among the functional models, multivariate functional approximation(MFA) [43, 42] has recently emerged as a new continuous meshless representation of discrete data. It approximates the raw discrete data by a set of piecewise smooth polynomial functions, and supports high-order evaluation of values and derivatives anywhere in the domain. An MFA model offers a compact representation as it consumes less storage space than the original discrete data, and serves as a surrogate in supporting spatiotemporal analysis in the continuous domain.

MFA is a technology supported by the United States Department of Energy (DOE) under the SciDAC RAPIDS Institute [2] and has been applied to large-scale projects in high-energy physics [1] and climate science [3]. MFA was originally proposed as a tool for modeling structured scientific datasets [43], and has been expanded to handle complex unstructured point clouds [26, 27] and operates in distributed parallel environments [32, 56]. It can also be used as an intermediate representation for high-quality volume rendering [55].

However, an unexplored area of research is to utilize continuous implicit models such as MFA for topological data analysis (TDA) and visualization. A fundamental step to recovering topology from an MFA model is to extract its critical points. In Morse theory, critical points correspond to topological changes in scalar fields and they play a crucial role in TDA [15]. In this paper, we present CPE-MFA, the *first* critical point extraction framework designed for MFA models of large-scale, high-dimensional data. Our method represents the first approach to enable topological feature extraction from implicit continuous models (such as MFA). A standard MFA model is designed as a forward model to evaluate variables at query locations. In

contrast, our CPE-MFA framework facilitates inverse evaluations for such a forward model, broadening the applicability of MFA in TDA.

In MFA, piecewise functional approximations replace discrete data with linear combinations of *basis functions* (e.g., nonuniform rational B-spline functions or NURBS) and a set of reference points called *control points* [43]. The places where the piecewise functions meet are known as *knots*, and the interval between knots are *knot spans* (or just *spans*). The function value in a span depends on a set of control points. Based on the known control points, we can first bypass spans devoid of critical points. We then employ Newton's method to extract critical points in the remaining spans. Since each piecewise functional is a polynomial, we could use its closed-form derivative to obtain the critical points, greatly reducing the complexity of Newton's method. Finally, we introduce spatial hashing to remove duplicated copies of critical points. Our contributions include:

- We introduce a novel framework, called CPE-MFA, for critical point extraction from MFA models. Our framework extracts critical points directly from an MFA model without the need for discretization or resampling.
- We propose multiple strategies to enhance the efficiency during critical point extraction, including multithreading, bypassing specific spans devoid of critical points, and removing duplicates using spatial hashing.
- We demonstrate the efficacy of our framework across multiple scientific datasets.

Our work is the first step toward enabling continuous implicit models such as MFA to support topological data analysis at scale.

2 RELATED WORK

We review relevant work on MFA and critical point extraction.

Multivariate Functional Approximation, or MFA, is a method for representing a scientific dataset with a continuous B-spline function for the purposes of analysis and visualization [43, 42, 26]. MFA may be considered a form of scattered data approximation (SDA), a field of mathematics concerned with constructing continuous functions that approximate a spatial dataset [59]. Numerous SDA methods have been developed over the years, with popular functional approximations based on wavelets [23], radial basis functions [33], and splines [12].

MFAs are built on geometric basis functions, specifically, B-splines. B-splines and their generalization, nonuniform rational B-splines or NURBS are smooth, flexible curves widely used in modeling and visualization software [46, 29]. Before MFA, Martin and Cohen proposed a model using NURBS to represent data in 2D and 3D [36]. This work was extended by Martin et al. [35] to parameterize 2D triangular and 3D tetrahedral data with tensor product splines. MFA is an extension of this method to model data with any number of dimensions.

Some visualization algorithms have been built upon geometric functional representations in the past. In 1997, Park and Lee [41] utilized a high-dimensional trivariate NURBS representation to visualize fluid flow data. In 2001, Martin and Cohen [36] constructed isosurfaces and achieved ray tracing using NURBS. More recently, Sun et al. [55, 56] developed techniques for scalable, interactive volume visualization of MFA representations. Their techniques produced high quality renders with moderate time complexity, and introduced fewer visual artifacts than traditional local filtering techniques.

*e-mail: guanqun.ma@utah.edu

†e-mail: dlenz@anl.gov

‡e-mail: tpeterka@mcs.anl.gov

§e-mail: guo.2154@osu.edu

¶ e-mail: beiwang@sci.utah.edu

Critical point extraction. TDA utilizes topological descriptors and provides robust feature extraction techniques for large-scale scientific data; see [21, 60] for surveys. These techniques have been applied to diverse research fields such as chemistry [40, 7], astrophysics [54, 49], and biomedical imaging [5, 8]. Critical points may correspond to atoms in chemistry [7], galaxy clusters in astrophysics [54, 49], and serve as good seed points for neuron reconstruction in biomedical imaging [11]. For scalar field data, critical points are part of the foundational structures of topological descriptors, such as merge trees, contour trees [9, 53], Reeb graphs [45], and Morse-Smale complexes [14, 13]. Morse theory [38] plays an important role in the study of critical points. Forman [16] generalized the theory from the smooth setting to the discrete setting, referred to as the discrete Morse theory, making it practical for studying discrete data. The challenge of extracting critical points for discrete data arises due to the absence of proper differentiability. Banchoff [6] characterized piecewise-linear (PL) critical points for an input PL scalar field defined on PL manifolds, based on lower- and upper-links of vertices. In particular, a vertex is *regular* if both its lower- and upper-links are simply connected, and *critical* otherwise. While there are various notions of PL critical points in the literature, they are shown to be equivalent, and correspond to discrete Morse cell in discrete Morse theory [18].

Recently, Vidal et al. [58] introduced a progressive method to extract critical points from PL functions defined on triangulated meshes. Their method utilized the fast identification of topologically invariant vertices using a hierarchical data representation. We use this method in Sec. 5 as a reference to evaluate our framework.

There are numerous studies on the numerical extraction of critical points, especially from a vector field perspective. Helman and Hesselink [22] first located isolated non-degenerate critical points of vector fields based on the Jacobian. A number of methods focused on solving a system of linear equations in a PL vector field, for each cell in a triangulated domain [28, 25]. Skala and Smolik [52] approximated discrete vector fields with radial basis functions (RBFs). In particular, based on a vector field represented as an RBF approximation, they constructed a function related to the speed of a particle and extracted its critical points.

3 TECHNICAL BACKGROUND

3.1 Multivariate Functional Approximation

At their core, MFA models are tensor-product B-spline functions that serve as an approximation to a dataset. In this section, we provide an overview of the basic definitions and constructions necessary to describe B-spline models for scientific data. A thorough presentation on the fundamental theory of B-splines can be found in the books by de Boor [12] and Piegl and Tiller [44].

B-spline curves. First, we consider the 1-dimensional case to illustrate the main components of a B-spline. Consider a set of point locations $\{u_0, \ldots, u_{m-1}\} \subset [0, 1]$ with a value f_i associated to each point.[1] A best-fit B-spline curve is a function $F : \mathbb{R} \to \mathbb{R}$ such that $F(u_i) \approx f_i$ for all indices i, subject to certain conditions that we will define shortly.

A B-spline curve of *degree p* is a piecewise-polynomial function with $p - 1$ continuous derivatives, where the pieces are polynomials of degree p. The points where the curve transitions between polynomial pieces are called *knots*, and the interval between two subsequent knots is a *knot span*. The overall shape of the B-spline is determined by the location of geometric *control points* scattered throughout the domain. The B-spline curve smoothly follows the polyline given by the control points, but does not coincide with the control points. Intuitively, it can be seen that B-splines with more control points can twist and bend with greater flexibility than those with fewer control points. Fig. 1 (left) shows examples of a 1-dimensional B-spline curve (top) and a 2-dimensional B-spline surface.

---

[1] We assume the input are re-scaled to lie in $[0, 1]$.

Throughout this paper, we will denote the degree of a B-spline by p. The set of control points is $\{P_j\}_{j=0}^{n-1}$, and the set of knots is denoted $\{t_j\}_{j=0}^{n+p}$. We note that the number of knots and control points in a B-spline are closely linked: a degree-p spline with n control points must have $n + p + 1$ knots [12]. Mathematically, B-splines may be described as linear combinations of B-spline basis functions, where the coefficient on each basis function is a control point. Namely,

$$F(u) = \sum_{j=0}^{n-1} N_{j,p}(u) P_j, \tag{1}$$

where the functions $N_{j,p}$ are bump functions that are nonzero in the subinterval $[t_j, t_{j+p+1}]$ and zero elsewhere. For a full description of the basis functions we refer to the canonical text by de Boor [12].

The purpose of MFA is to construct the B-spline that minimizes the root mean squared (RMS) error between a set of values and the spline's approximation to those values. Given a set of point-value pairs $\{u_i, f_i\}$, a degree p, and a set of knot locations $\{t_j\}$, the best-fit B-spline is the B-spline function F that solves the minimization problem:

$$\min_F \left(\frac{1}{m} \sum_{i=0}^{m-1} |f_i - F(u_i)|^2 \right)^{1/2}. \tag{2}$$

MFA can also construct a B-spline model using an adaptive feedback loop. Fig. 1 (right) reproduces the pipeline to construct an MFA model from the raw input data with an adaptive refinement algorithm. MFA begins by rescaling input data points to the interval $[0, 1]$ and initializing a knot distribution with a small number of control points. The loop begins by computing the best-fit B-spline given the coarse distribution of knots, and checking if the desired error tolerance is achieved. If not, MFA repeatedly adds new knots, computes a new best-fit B-spline over the new knots, and checks the error again. The process ends when the resultant B-spline has the desired pointwise errors, or a maximum number of knots are added.
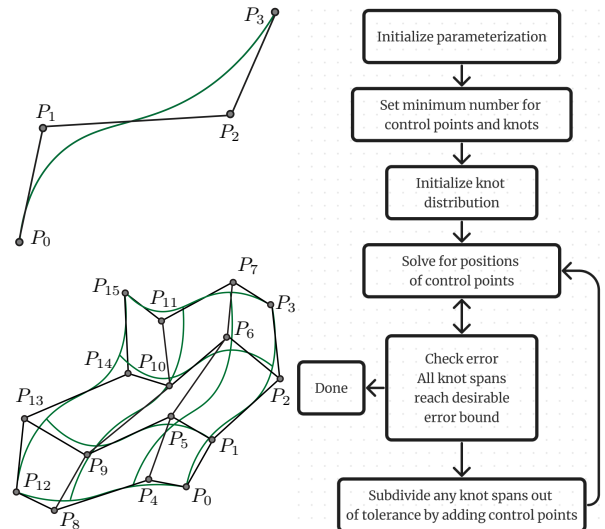

Figure 1: Left: a 1-dimensional B-spline curve (top), and a 2-dimensional B-spline surface (bottom). P_i are control points, control meshes are in black, approximated curve/surface are in green. Right: an overall MFA model pipeline reproduced from [42, Fig. 1.4].

B-Splines in higher dimensions. The preceding discussion can be extended to 2- and higher-dimensional data with tensor-product B-splines. In a tensor-product B-spline, the basis functions are still bump functions and are formed by multiplying 1-dimensional B-spline basis functions from each dimension together. For a 2-dimensional case, a B-spline of degree p has two sets of knots $\{t_{j_1}^{(1)}\}_{j_1=0}^{n_1+p}$ and $\{t_{j_2}^{(2)}\}_{j_2=0}^{n_2+p}$ for the x and y dimensions, respectively.

The 2-dimensional basis functions are of the form $N_{j_1,j_2}(u,v) = N_{j_1,p}^{(1)}(u)N_{j_2,p}^{(2)}(v)$. Consequently, the mathematical expression for a 2-dimensional tensor-product B-spline is as follows, visualized in Fig. 1 (left bottom),

$$F(u,v) = \sum_{j_1=0}^{n_1-1}\sum_{j_2=0}^{n_2-1} N_{j_1,p}^{(1)}(u)N_{j_2,p}^{(2)}(v)P_{j_1,j_2}. \qquad (3)$$

For the general d-dimensional case, the form of a tensor-product B-spline is

$$F(u_1,\cdots,u_d) = \\ \sum_{j_1}\cdots\sum_{j_d} N_{j_1,p}^{(1)}(u_1)\cdots N_{j_d,p}^{(d)}(u_d)P_{j_1,\cdots,j_d}. \qquad (4)$$

In one dimension, the term "knot span" refers to the subinterval $[t_j, t_{j+1}]$ between any two adjacent knots. In higher dimensions, a knot span refers to the tensor product of 1-dimensional knot spans. Thus a d-dimensional span would be a region like

$$[t_{j_1}^{(1)}, t_{j_1+1}^{(1)}] \times \cdots \times [t_{j_d}^{(d)}, t_{j_d+1}^{(d)}].$$

Strong convex hull property. Unlike other continuous models for scientific data, B-splines have a useful property that can be leveraged to dramatically accelerate the extraction of critical points. The idea behind the *strong convex hull property* of B-splines is that, within a given knot span, the values of the spline function are entirely contained within the convex hull of the neighboring control points. More specifically, we can say the following:

Theorem 1 (Strong Convex Hull Property). *If $(u_1,\cdots,u_d)$ is a point in the span $[t_{j_1}^{(1)}, t_{j_1+1}^{(1)}] \times \cdots \times [t_{j_d}^{(d)}, t_{j_d+1}^{(d)}]$, then $F(u_1,\cdots,u_d)$ lies in the convex hull defined by control points $P_{h_1,\cdots,h_d}$, where $j_l \le h_l \le j_l + p$, for $1 \le l \le d$.*

3.2 Critical Point Extraction

Newton's Method, also known as the Newton-Raphson method, is a widely used numerical algorithm to find the roots of a differentiable function.

Suppose $f : \mathbb{R} \to \mathbb{R}$ is a differentiable scalar function. Starting with an initial guess of a value x_0, the iterative process is repeated as

$$x_{n+1} = x_n - \nabla f(x_n)^{-1} f(x_n), \qquad (5)$$

where n is the iteration number, x_n is the current estimate of the root, and $\nabla f(x_n)$ is the gradient of f at x_n. The iteration stops after step n when $||\nabla f(x_n)||$ is smaller than a predefined threshold ε.

MFA is a continuous polynomial function, and its high-order derivatives can be calculated analytically. Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is a general differentiable function defined on a d-dimensional domain. Finding the critical points of MFA is equivalent to solving the equation $\nabla f(x) = 0$. Newton's method fits well in this scenario. Starting with an initial guess, a d-dimensional vector x_0, the iterative formula is

$$x_{n+1} = x_n - H(x_n)^{-1}\nabla f(x_n). \qquad (6)$$

Here, x_n is a d-dimensional vector at the nth iteration, $H(x_n)^{-1}$ is the inverse of the Hessian matrix of f at x_n, $\nabla f(x_n)$ is the gradient of the function f at x_n. Iteration terminates when $||\nabla f(x_n)|| \le \varepsilon$.

Critical points. Geometrically, the goal of studying critical points is to extract features surrounding them that help scientists understand and represent data for downstream analysis (e.g., [4]). Critical point detection is the identification of extrema and saddles that assists in understanding the topology of a scalar field.

Morse theory [38] is a mathematical tool to detect and classify critical points. Given a function $f : \mathbb{M} \to \mathbb{R}$ defined on a smooth manifold $\mathbb{M}$, the gradient ∇f vanishes at a critical point p. A critical point p is *non-degenerate* if its Hessian matrix $H(p)$ is not singular. A Morse function requires all its critical points to be non-degenerate and have distinct function values. The index $\lambda(p)$ of a critical point p is defined by the number of negative eigenvalues of its Hessian $H(p)$: local maxima have zero negative eigenvalues, local minima have all negative eigenvalues, and saddles have some (but not all) negative eigenvalues. Extraction critical points in a PL setting is described in the supplement.

4 METHOD

Our CPE-MFA framework consists of three main stages. First, we bypass spans devoid of critical points and work with the remaining valid ones. Second, we utilize Newton's method to extract critical points from each valid span. Since each piecewise functional is a polynomial, we use its closed-form derivative to obtain the critical points, greatly reducing the complexity of Newton's method. Finally, we remove duplicated critical points using spatial hashing.

CPE-MFA is designed for seamless integration with multithreading to enhance its efficiency in span filtration and critical point extraction from spans. In these two stages, each span can be processed independently, which naturally aligns with multithreading.

4.1 Span Filtration

By applying the strong convex hull property of B-splines (c.f. Sec. 3.1), we are able to quickly determine that some regions in the domain cannot possibly contain a critical point. This allows us to exclude these (potentially large) regions from our fine-grained search for critical points.

First, we note that the derivative of B-spline function is also a B-spline function [44]. The partial derivative of Eq. (3) with respect to the first dimension is:

$$\frac{\partial}{\partial u_1}F(u_1,\cdots,u_d) = \\ \sum_{j_1=0}^{n_1-2}\sum_{j_2=0}^{n_2-1}\cdots\sum_{j_d=0}^{n_d-1} N_{j_1,p-1}^{(1)}(u_1)N_{j_2,p}^{(2)}(u_2)\cdots N_{j_l,p}^{(d)}(u_d)\tilde{P}_{j_1,\cdots,j_d}, \qquad (7)$$

where

$$\tilde{P}_{j_1,\cdots,j_d} = \frac{p}{t_{j_1+p+1}^{(1)} - t_{j_1+1}^{(1)}}\left(P_{j_1+1,j_2,\cdots,j_d} - P_{j_1,\cdots,j_d}\right). \qquad (8)$$

Partial derivatives in the other directions follow by analogy. We abbreviate the derivative notation as $\partial_l F = \frac{\partial}{\partial u_l}F$.

To locate all critical points of the B-spline defined in Eq. (4), we will identify points where every first partial derivative is zero. But since the partial derivatives of a B-spline are also B-splines, the problem of finding vanishing partial derivatives simply reduces to the problem of finding roots of the associated "derivative B-spline."

For each dimension, we compute the derivative spline using Eq. (7). Next, we scan the control points of the derivative spline to find regions in the domain that cannot contain zero values; if the derivative spline is not zero in this region, then there cannot be a critical point in this region. To achieve this, we utilize the strong convex hull property.

Consider an individual knot span and a direction l. From Theorem 1, we know that throughout this entire span, the value of the derivative spline $\partial_l F$ is within the convex hull of its neighboring control points. Should this convex hull fail to encompass zero, it must follow that $\partial_l F$ never vanishes within this knot span. Thus, this knot span cannot contain a critical point. Therefore, by iterating through each span and computing the convex hull of the associated control points, we can find spans guaranteed to not harbor a zero derivative in at least one dimension. By excluding all of these spans, we can focus our search only on spans that may contain critical points.

This span filtration stage significantly streamlines the computational effort by discarding spans that do not contain critical points. As a result, this focused strategy not only saves computational resources but also time, enhancing the efficiency of extracting critical points.

4.2 Critical Point Extraction in a Single Span

For a point $(u_1, \cdots, u_d)$ in the span $[u_{i_1}, u_{i_1+1}] \times \cdots \times [u_{i_d}, u_{i_d+1}]$, the polynomial function $f := F(u_1, \cdots, u_d)$ is decided by control points $P_{h_1, \cdots, h_d}$. The derivative of this polynomial function f, described in Eq. (7), is crucial for extracting critical points in the span.

Specifically, the critical points are identified by locating all points in the span whose first derivatives are zero in every dimension. MFA provides accurate high-order derivatives. Given the availability of the first and second derivatives of f, Newton's method becomes a natural choice for finding zeros of the first derivative of MFA.

Critical points within each span are computed using Newton's method, detailed in Algorithm 1, with a set of initial points X uniformly sampled in the span. We sample $(p+1)^d$ initial points in each span. Let i_{max} be an upper bound on the number of iterations using the Newton's method. We set $i_{max} = 20$ to ensure convergence, although the average number of iterations is often much lower as shown in Tab. 3. During an iteration of Newton's method, if a point moves beyond a certain distance ξ from the center of the span, then it is unlikely to return (as a critical point), prompting the termination of the iteration to enhance the efficiency of the algorithm. In our experiments, setting ξ to be five times the span's diagonal length appears to be adequate. We focus on extracting non-degenerate, isolated critical points with non-zero ($\geq \delta$) determinant of the Hessian matrix. We choose $\delta = 10^{-13}$ to be a sufficiently small positive number that strikes a balance between precision and efficiency.

Throughout the process, when we identify a critical point within a span, we compare it against the set of already identified critical points Z within the span. If the distance from newly identified critical point x to the existing points in Z is less than τ, we consider this point to be a duplicate and ignore it. Otherwise, if x is within the span, we include it into the list of critical points Z. ε is the threshold to decide if the gradient is small enough to stop the iteration. τ and ε are user-defined parameters discussed in Sec. 5.

Algorithm 1 Critical Point Extraction in a Single Span

Require: An input span s, and a function f defined on the span.
Ensure: A set of critical points Z in the span s.
1: $Z = \emptyset$.
2: Uniformly sample a set of initial points X in the span s.
3: Compute the center c of the span s.
4: **for all** initial point $x \in X$ **do**
5:     $i = 0, x_i = x$
6:     **while** $i < i_{max}$ **do**
7:        **if** $\det(H(x_i)) < \delta$ **then**
8:           **break**
9:        $x_{i+1} = x_i - H^{-1}(x_i)\nabla f(x_i)$, H is the Hessian matrix.
10:       **if** $||x_{i+1} - c|| > \xi$ **then**
11:          **break**
12:       **if** $||\nabla f(x_{i+1})|| < \varepsilon$ **then**
13:          **if** $||x_{i+1} - y|| \geq \tau, \forall y \in Z$ and $x_{i+1} \in s$ **then**
14:             $Z = Z \bigcup x_{i+1}$
15:          **break**

4.3 Duplication Removal

After removing duplicated critical points within each span in Algorithm 1, there may still be duplicates near the junctions of spans. We address this issue by introducing spatial hashing. Assuming that critical points within a distance less than τ are considered to be duplicates, our duplication removal algorithm ensures that such points are assigned to the same hash buckets.

For a point $x = (x_1, \cdots, x_d)$, we would like to "snap" it to the nearest integer grid point and apply spatial hashing. To do so, let $k = \frac{x}{20\tau} = (k_1, \cdots, k_d)$. We consider a hash index associated with x, $I = (I_1, \cdots, I_d)$. Depending on the approximation, each I_i can have two distinct values $\lfloor k_i \rfloor$ and $\lceil k_i \rceil$. We choose $k = \frac{x}{20\tau}$, and two rounding methods (floor and ceiling) so that x shares at least one hash index with all points within a distance τ.

By choosing floor or ceiling for each dimension I_i, we obtain a set of hash indices, $\mathcal{I} = \{I\}$, where there are 2^d distinct hash indices in $\mathcal{I}$. For each $I \in \mathcal{I}$, we employ boost::hash_combine() from the *Boost C++ Libraries* as the hash function to compute the hash values V for I. Finally, we compare x to all the points y already in the hash buckets determined by V. If x and y are less than τ apart, then x is flagged as a duplicate. The pseudocode for spatial hashing is available in the supplement.

4.4 Time Complexity

We give a summary of complexity here; see the supplement for details. In span filtration, $p(p+1)^{d-1}$ control points are utilized in every span of the first derivative. The cost of span filtration is $\mathcal{O}(dp^d n)$, where n is the number of spans. Each iteration of Newton's method involves computing the Hessian and gradient at $\mathcal{O}(d^2 p^d)$ and solving linear system at $\mathcal{O}(d^3)$. Since $(p+1)^d$ initial points are used in each span, the cost of finding critical points in all spans is $\mathcal{O}(i_{max} d^2 p^{2d} n)$, whereas the spatial hashing takes $\mathcal{O}((2p)^d dn)$. When $p \geq 2$, the overall time complexity is $\mathcal{O}(i_{max} d^2 p^{2d} n)$. In practice, $p, d \ll n$ (and can be treated as constants). In all the experiments conducted, more than 95% of the time was spent finding critical points using Newton's method, corroborating the analysis of time complexity.

5 EXPERIMENTAL RESULTS

We perform a sanity check for the validity of our framework with a synthetic dataset (Sec. 5.1) followed by a number of scientific datasets in 2D (Sec. 5.2 and Sec. 5.3) and 3D (Sec. 5.4 and Sec. 5.5) with complex topological features. We employ adaptive fitting MFA [43] and use degree-3 and degree-2 polynomials for the synthetic and scientific datasets, respectively. After fitting an MFA model into a discrete dataset, the MFA model serves as the input and basis for all further experiments.

Implementation. We conduct experiments on a desktop equipped with an Intel 3.5 GHz Core i9 CPU featuring 8 hardware cores and 8 threads, along with 32 GB of DDR4 RAM. Our code is compiled using g++ version 11.4.0 with -O3 optimization. We use threading building blocks (TBB) for parallelization. TBB balances the workload in every thread dynamically. The number of threads is set to be the number of hardware cores.

TTK. The Topology ToolKit (TTK) [57] is a toolbox for topological data analysis. TTK assumes that the input data is a piecewise linear (PL) scalar field f defined on a PL manifold M of dimension 2 or 3 (e.g., a mesh interpolating a discrete set of points). f has value at the vertices of M, and is linearly interpolated on the higher-order mesh elements. In the PL setting, a vertex is *regular* if both its lower link and upper link are simply connected; otherwise, it is a *critical point* of f. TTK extracts critical points from M that comply with a discrete gradient [57]. We call these critical points the *PL critical points*, to differentiate them from those extracted from MFA using our approach. From an implementation perspective, we call the ttkScalarFieldCriticalPoints() function from TTK [58] to extract PL critical points from a PL function defined on a mesh.

Evaluation. Our initial objective is to evaluate whether our CPE-MFA framework could extract all critical points from each scientific dataset. However, since the "ground truth" critical points from each dataset are unknown, we evaluate the output of CPE-MFA against those obtained by applying TTK to an MFA model. We aim to demonstrate that critical points extracted by our approach align well with the expectation. That is, the critical points extracted from an

MFA model using our approach are similar to those extracted by a different critical point extraction method (e.g., TTK) applied to the same MFA model. The results of TTK are used as a reference, not as the "ground truth".

First, we apply MFA to obtain a continuous implicit model. Second, we extract critical points from the MFA model using our approach based on Newton's method; this process is called CPE-MFA. Third, since TTK is not natively equipped to extract critical points from continuous implicit models, we construct a PL dataset by sampling a set of discrete points from the MFA model, and apply TTK to compute its PL critical points. We assume that a higher resolution PL dataset aligns more with a continuous model and captures the critical points more accurately. This process is called TTK-MFA. In summary, CPE-MFA processes critical points as elements of a continuous model, whereas TTK-MFA operates on a PL dataset sampled from an MFA model.

Despite the fundamental difference between these two approaches, the TTK-MFA output provides a valuable reference for assessing the effectiveness of CPE-MFA. We demonstrate across all scientific datasets in this paper that despite minor differences, CPE-MFA critical points are similar to those from TTK-MFA.

Technical details. For TTK-MFA, we sample a PL dataset (with a grid size of 1) from the MFA model that is identical in size to the original input data, to maintain consistency in our comparative analysis. Thus, in Secs. 5.2 to 5.5, we set a threshold $\tau = 0.999$ for duplication removal in CPE-MFA, based on the grid size 1 to sample the PL dataset for TTK-MFA. TTK-MFA may identify some points on the boundary of the domain to be critical points (referred to as *boundary critical points*) due to partial neighborhood information. However, it cannot be ascertained that these points are genuine critical points with zero gradient. Therefore, we exclude boundary critical points identified by TTK-MFA in our result presentation.

To quantitatively evaluate the similarity, we utilize the Jaccard index, defined as $\frac{|A \cap B|}{|A| + |B| - |A \cap B|}$, where $|A|$ and $|B|$ are the number of critical points from CPE-MFA and TTK-MFA, respectively, and $|A \cap B|$ denotes the number of critical point alignments. A critical point from CPE-MFA is aligned with one from TTK-MFA if they are less than a grid size apart from each other (and thus considered to be co-located).

Datasets and blocks. For each scientific dataset, visualized in Fig. 2, we extract three blocks of data to highlight the extracted critical points from both CPE-MFA and TTK-MFA. The sizes and locations of these blocks are described in the supplement.

Dataset	Original #Spans	Actually Evaluated #Spans (%)	Skipped #Spans (%)
CESM	276740	43039 (15.55%)	233701 (84.45%)
S3D	44377	9908 (22.33%)	34469 (77.67%)
QMC	18432	8229 (44.65%)	10203 (55.35%)
RTI	1741932	1707319 (98.01%)	34613 (1.99%)

Table 1: Results of span filtration, reporting the number of original spans, the number of spans (and percentage) actually being evaluated, and the number of spans (and percentage) being skipped during the optimization.

Span optimization. We report the number of spans identified in each dataset before and after the span filtration process, as shown in Tab. 1. Our optimization process ignores a substantial number of spans (up to 84%). Even for the most complex RTI dataset (Sec. 5.5), a number of spans are excluded from the final phase of critical point extraction. These exclusions highlight the effectiveness of our span filtration process in reducing irrelevant or redundant spans, thus substantially accelerating the overall critical point extraction algorithm.

Running time. Finally, we report the running time of CPE-MFA and TTK-MFA for all scientific datasets in Tab. 2.
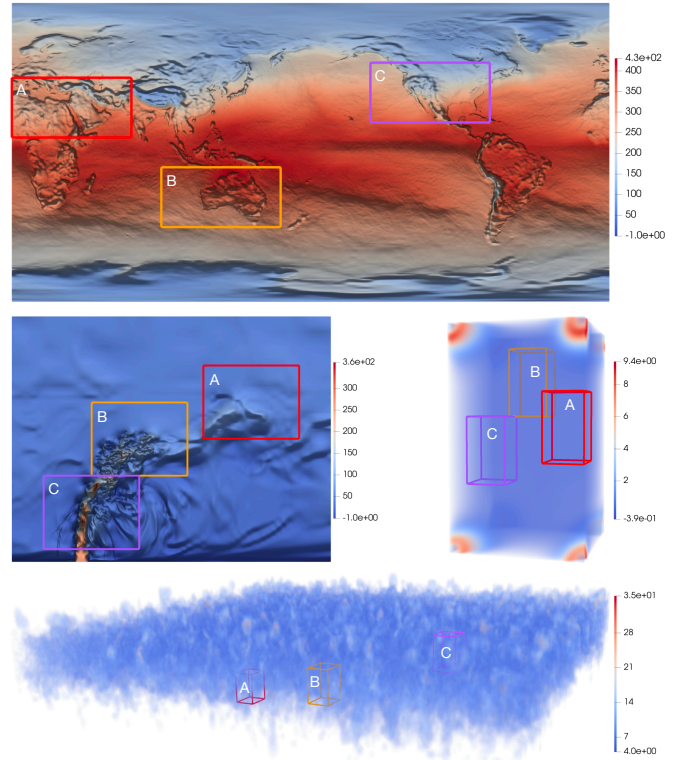

Figure 2: Scientific datasets: CESM (top), S3D (middle left), QMC (middle right), and RTI (bottom).

Block	CPE-MFA original	TTK-MFA original	CPE-MFA upsample	TTK-MFA upsample
CESM				
Entire domain	3.30	1.66	3.43	-
A	0.314	0.0759	0.391	7.305
B	0.187	0.0724	0.189	7.720
C	0.122	0.0823	0.122	7.306
S3D				
Entire domain	0.554	0.101	0.560	-
A	0.017	0.0256	0.017	0.756
B	0.169	0.0228	0.169	0.772
C	0.169	0.0289	0.167	0.773
QMC				
Entire domain	3.310	0.517	3.375	-
A	0.180	0.0316	0.222	15.247
B	0.149	0.0304	0.151	15.646
C	0.159	0.0316	0.161	16.002
RTI				
Entire domain	1003.1	15.50	985.1	-
A	0.887	0.0186	0.876	1.823
B	1.079	0.0165	1.124	1.840
C	1.209	0.0189	1.165	1.973

Table 2: Running time (in seconds) across all scientific datasets. The upsampling ratio is at 10^2 for the 2D and 10^3 for the 3D datasets.

5.1 Schwefel Dataset

A (scaled) Schwefel function is a non-convex function that can be generated in any dimension [47], that is, for $\mathbf{x} = [x_1, x_2, \ldots, x_d]$,

$$f(\mathbf{x}) = \frac{1}{2}\left(418.9829d - \sum_{i=1}^{d} x_i \sin(\sqrt{|x_i|})\right), \quad (9)$$

where d is the dimension. It serves as a good synthetic dataset as it is complex and contains many local minima. There are $(2k-2)^d$ non-degenerate critical points in the domain $[-((k+\frac{1}{2})\pi)^2, ((k+\frac{1}{2})\pi)^2]^d$, where k is a natural number. We scale the function by $1/2$ for a better visualization. Among all the critical points, there are $(k-1)^d$ local maxima, $(k-1)^d$ local minima, and $2(k-1)^d$ saddles.
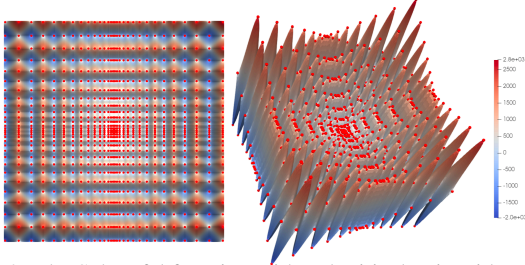

Figure 3: The Schwefel function with red critical points identified by CPE-MFA: (left) top view (right) side view.

To evaluate the efficacy of our method, we generate a *Schwefel dataset* by sampling from a 2D Schwefel function uniformly, selecting 200^2 points within the domain $[-2400, 2400]^2$. Theoretical analysis predicts the existence of 900 critical points in this domain. We apply MFA to this dataset with a polynomial degree of 3 and 100^2 control points. Using a span filtration, 2809 spans are identified for critical point extraction out of all 9409 spans. As shown in Fig. 3, CPE-MFA accurately identifies 900 critical points, aligning perfectly with the theoretical expectation. Meanwhile, CPE-MFA successfully classifies them into 225 local minima, 225 local maxima, and 450 saddles. We set τ to be 10^{-4} times the minimum width of each dimension in the domain.

Utilizing this dataset, in Tab. 3, we illustrate the impact of ε on both the number of iterations (required by Newton's method) and the accuracy of the extracted critical points, accessed by the magnitude of the gradient. The theoretical average gradient magnitude of critical points is zero. Setting $\varepsilon = 10^{-7}$ achieves an optimal balance between the number of iterations and the accuracy of the extracted critical points. This ε is tested across several synthetic datasets and consistently performs well. Therefore, we use $\varepsilon = 10^{-7}$ for all the experiments.

ε	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-11}
Avg #itr	3.16	3.51	3.78	3.91	19.34
Avg grad mag	$9.00e^{-5}$	$7.91e^{-7}$	$2.20e^{-9}$	$1.36e^{-10}$	-

Table 3: Average number of iterations and average gradient magnitude of the extracted critical points with different ε. When $\varepsilon = 10^{-11}$, Newton's method cannot find all critical points within $i_{max} = 20$ iterations.

5.2 CESM Dataset

The Community Earth System Model (CESM) offers comprehensive global climate data, spanning the Earth's land, oceans, atmosphere and sea ice. Our analysis focuses on the FLDSC variable, which represents the clear-sky downwelling long-wave flux on the surface, within the Community Atmosphere Model (CAM) developed at the National Center for Atmospheric Research (NCAR) [39]. The CESM dataset we use comprises a 3600×1800 2D domain, with each grid point representing a singular value of the FLDSC variable.

For qualitative and quantitative analysis, Fig. 4 visualizes the critical points identified by both methods, whereas Tab. 4 reports their numbers, position and type alignments. In Figs. 4 to 7, critical points where CPE-MFA and TTK-MFA align are depicted in yellow. The critical points exclusive to TTK-MFA are colored in purple, and those exclusive to CPE-MFA are shown in pink. In Fig. 4 A, B, and C, the extracted critical points from CPE-MFA and TTK-MFA mostly align

with each other (observing all yellow points), with small discrepancies between the two methods across all three blocks. The Jaccard indices between these two sets of critical points are shown in Tab. 4 (original), valued at 0.76 for the entire dataset, and 0.76, 0.74, 0.75 for blocks A, B, and C, respectively.

There are minor discrepancies (misalignments) in certain regions where critical points are detected by one but not the other method (see independent purple or pink points within the boxes 1, 2, and 3, with zoomed-in views). We hypothesize that these discrepancies are due to the sampling process that generates a PL dataset for TTK, which may not capture all the features from the continuous MFA model and may contain spurious critical points due to the PL approximation. We hypothesize further that upsampling the PL dataset will improve the alignment between the two methods.

To validate this hypothesize, we increase the sampling resolution for TTK to better preserve the features of MFA. By applying an upsampling ratio of 10^2, we expand the size of the PL dataset in every dimension by a factor of 10 relative to the original sampling resolution. Critical points extracted using TTK from the upsampled PL dataset demonstrate improved alignment between the two methods, see Fig. 4 A*, B*, and C*, as well as zoomed-in boxes: 1 vs 1*, 2 vs 2*, and 3 vs 3*. As shown in Tab. 4 (upsample), the number of perfectly aligned critical points increases with upsampling, and the Jaccard indices (upsample) also increase from around 0.75 to 0.86 across all three blocks. For points that do not align, we observe closely situated points in different colors.

For the remaining misaligned critical points, an increased upsampling ratio of 10^4 gives rise to perfectly aligned critical points between the two methods in these regions (results not shown).

In summary, the improved alignment with upsampling validates our hypothesis regarding the initial discrepancies.

Details on critical point alignment. Tab. 4 reports the number of critical points identified by CPE-MFA and TTK-MFA, respectively. The column TTK-MFA original lists the number of critical points identified by TTK-MFA in the original resolution ($u = 1$). To ensure comparability between the results of CPE-MFA from a continuous model and those of TTK-MFA from the corresponding PL dataset, we align τ with the grid size. Thus, the column CPE-MFA original displays the number of critical points detected by CPE-MFA at a duplicate removal threshold $\tau = 0.999$. The column TTK-MFA upsample reveals the number of critical points identified by TTK-MFA after upsampling at a ratio $u = 10^2$. Correspondingly, the column CPE-MFA upsample uses a threshold of $\tau = 0.0999$ under the same upsampling condition.

Highlighted results. We would like to emphasize that we do not expect the critical points to align perfectly between the two methods, in both locations and quantities, since they are fundamentally different extraction methods. Using TTK-MFA extracted critical points as a reference, we demonstrate a strong alignment between the two sets of critical points (in terms of location and type), indicating the validity of our CPE-MFA framework. Furthermore, as we increase the upsampling ratio for TTK, the alignment between the two sets of critical points improves, validating our hypothesis that the discrepancies are due to the sampling resolution (thus approximation quality) for TTK.

5.3 Turbulent Combustion Dataset

The dataset is from a turbulent combustion S3D simulation [10] that models the combustion of a fuel jet influenced by an external cross flow [17, 20, 42]. The simulation uses a 3D domain of size $704 \times 540 \times 550$. The variable of interest represents the magnitude of the 3D velocity within the domain. For our experiment, we use a 2D cross-section of the dataset of size 704×540.

The critical points extracted by CPE-MFA and TTK-MFA are visualized in Fig. 5 A, B, and C across all three blocks. Mirroring similar observations from the CESM dataset, we observe reasonable critical point alignments between the two methods, despite minor dis-
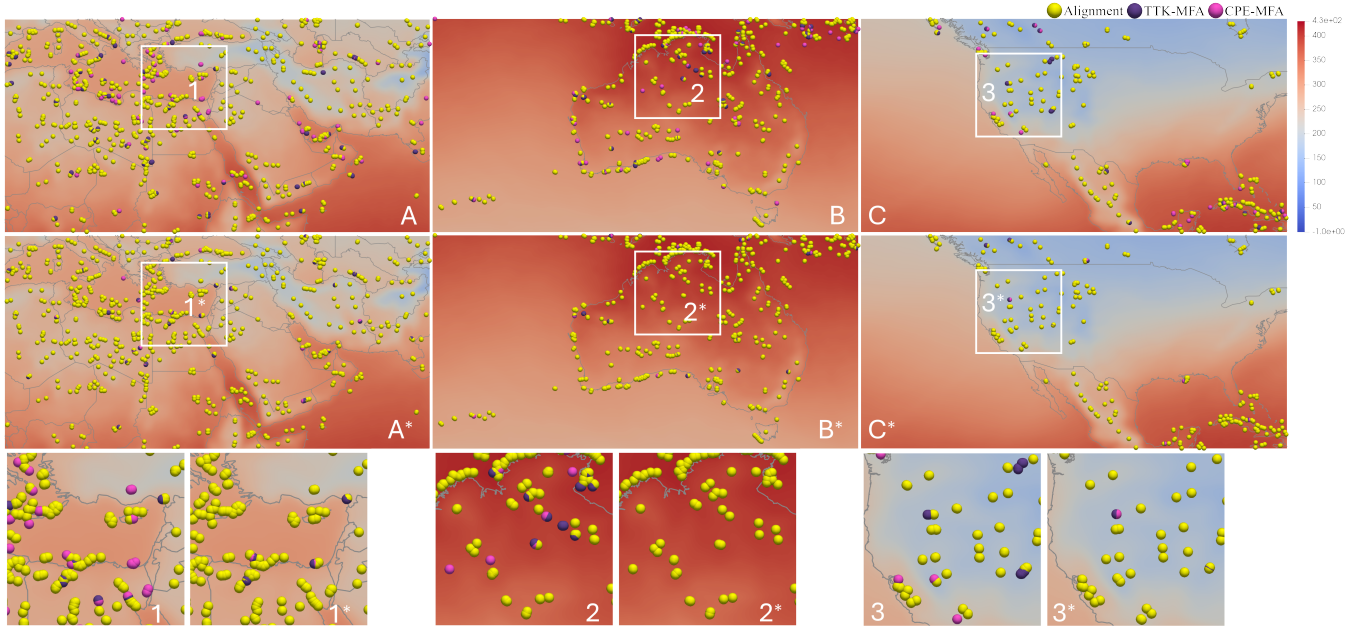
Figure 4: CESM dataset with critical points identified by CPE-MFA and TTK-MFA. Top: critical points from blocks A, B, and C, respectively. Middle: critical points from upsampled blocks A, B, and C (labeled as A*, B*, and C*), respectively, with a ratio of 10^2. Bottom: zoomed-in views of regions in the domain with and without upsampling (at a ratio of 10^2).
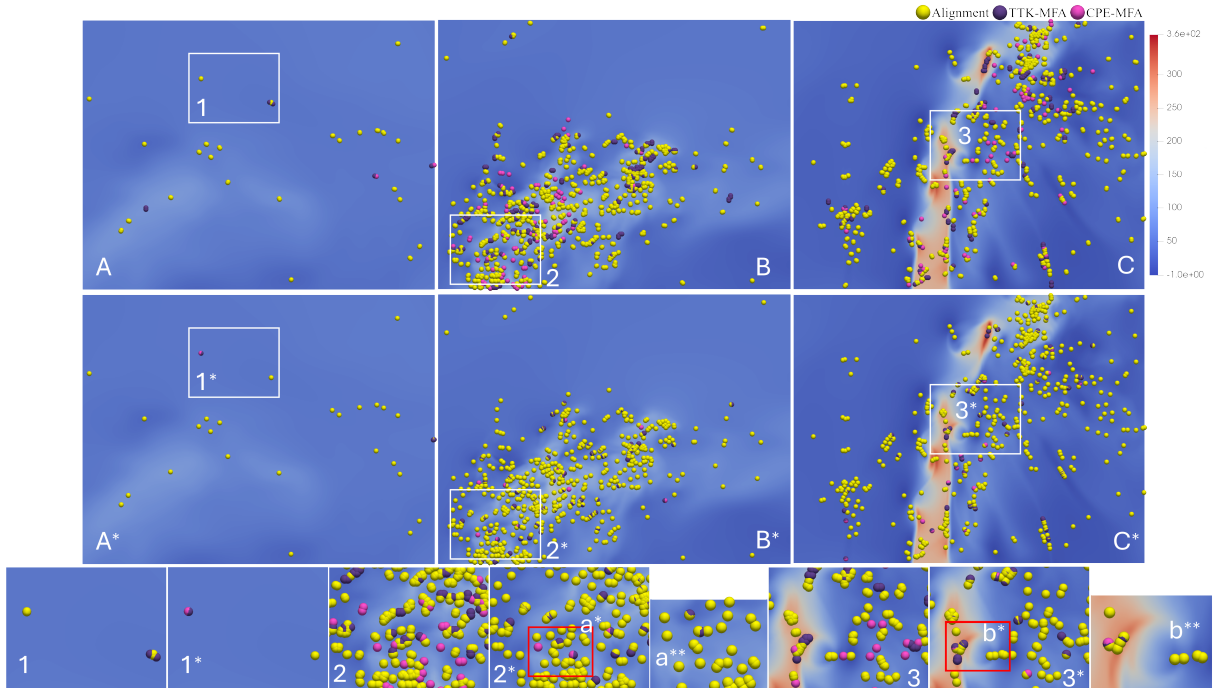

Figure 5: S3D dataset with critical points identified by CPE-MFA and TTK-MFA. Top: critical points from blocks A, B, and C, respectively. Middle: critical points from upsampled blocks A, B, and C (labeled as A*, B*, and C*), respectively, with a ratio of 10^2. Bottom: $(1, 1^*, 2, 2^*, 3, 3^*)$: zoomed-in views of regions in the domain with and without upsampling at a ratio of 10^2; (a^{**}, b^{**}): better critical point alignments (in regions a^*, b^*, respectively) after upsampling at a ratio of 10^4.

crepancies; see the zoomed-in views of boxes 1, 2, and 3 with blocks A, B, and C, respectively. The number of critical points and their similarities (reported as Jaccard indices) are shown in Tab. 4.

Again, these discrepancies can be mitigated following an upsampling for TTK at a ratio $u = 10^2$. As shown in the zoomed-in boxes 2 vs. 2* and 3 vs. 3* in Fig. 5, the critical points from both methods become more aligned in these regions with upsampling. In box 1*, even though each method has an unaligned point, the points nearly overlap with each other. The distance between these two points is

smaller than the distance between the corresponding points in box 1. Thus, points in box 1* is more aligned than in box 1. The Jaccard indices among the two sets of critical points also increase drastically with upsampling, as shown in Tab. 4.

A few discrepancies remain (at an upsampling resolution 10^2), they are highlighted in regions enclosed by small red boxes labeled as a^* and b^*. Critical points in these regions become more aligned with further upsampling at a ratio of 10^4; see the zoomed-in views of a^{**}, b^{**} in Fig. 5. For all unaligned points, nearly overlapping

Block	CPE-MFA original #cp	TTK-MFA original #cp	Aligned #cp / type	Jaccard index original	CPE-MFA upsample #cp	TTK-MFA upsample #cp	Aligned #cp / type	Jaccard index upsample
CESM (2D)								
Entire domain	7940	8759	7204 / 7149	0.76	8549	-	-	-
A	714	765	641 / 636	0.76	783	865	767 / 766	0.87
B	421	461	374 / 373	0.74	473	527	461 / 461	0.86
C	208	218	183 / 182	0.75	228	254	223 / 222	0.86
D	623	676	553 / 551	0.74	688	772	664 / 661	0.83
E	636	703	591 / 588	0.79	684	801	674 / 672	0.83
S3D (2D)								
Entire domain	1317	1732	1150 / 1116	0.61	1569	-	-	-
A	27	31	25 / 24	0.76	27	36	26 / 26	0.70
B	485	646	415 / 405	0.58	608	813	596 / 595	0.72
C	450	552	379 / 363	0.61	562	717	541 / 536	0.73
D	72	98	70 / 68	0.70	74	100	73 / 73	0.72
E	44	62	43 / 43	0.68	45	63	44 / 44	0.69
QMC (3D)								
Entire domain	210	262	167 / 161	0.55	244	-	-	-
A	64	61	46 / 45	0.58	75	91	73 / 73	0.78
B	14	20	13 / 11	0.62	14	52	14 / 14	0.27
C	23	20	17 / 17	0.65	26	29	25 / 25	0.83
D	9	17	9 / 9	0.53	9	13	8 / 8	0.57
E	9	10	7 / 5	0.58	10	10	10 / 10	1.00
RTI (3D)								
Entire domain	711504	670156	425906 / 292248	0.45	1258055	-	-	-
A	118	81	61 / 50	0.44	181	253	161 / 156	0.59
B	438	292	256 / 172	0.54	827	1352	735 / 710	0.51
C	415	344	280 / 185	0.58	807	1198	733 / 712	0.58
D	503	315	289 / 186	0.55	952	1482	858 / 825	0.54
E	467	346	300 / 211	0.58	852	1223	771 / 753	0.59

Table 4: Number of critical points (#cp) extracted by CPE-MFA and TTK-MFA within each of the four scientific datasets as well as their corresponding selected blocks: original #cp shows the number of critical points at the original resolution; upsample #cp shows the number of critical points at an upsampling ratio of 10^2 for the 2D and 10^3 for the 3D datasets; aligned #cp shows the number of critical points perfectly aligned between the two methods; aligned type shows the number of aligned critical points that share the same critical point type; Jaccard index reports the similarity between the two sets of critical points extracted from CPE-MFA and TTK-MFA respectively. Critical points from TTK-MFA in the upsampling setting are not reported as it runs out of memory when processing the large data. We add two additional randomly selected blocks (block D and block E) per dataset to collect more statistics on the distributions of critical points.

points in different colors can be found. As the threshold for alignment decreases with upsampling, the criteria for alignment become increasingly stringent. For critical points in a** and b**, even if they do not align, they remain very close to each other.

Highlighted results. Recall that TTK-MFA extracts critical points from PL dataset sampled from an MFA model, whereas CPE-MFA extracts critical points directly from the same continuous MFA model. The output from TTK-MFA depends on the sampling resolution, whereas the output of CPE-MFA does not depend on discretization or sampling. However, we do observe an increase in the number of critical points from CPE-MFA as we increase the sampling resolution. This is because we decrease the threshold used to remove duplicates to be aligned with the upsampling resolution. The key observation is that as the sampling resolution increases, the result of TTK-MFA becomes more aligned with that of CPE-MFA. Since CPE-MFA operates directly on a continuous model, its output is not severely impacted by discretization or sampling, as does TTK-MFA.

5.4 Quantum Monte Carlo Dataset

The QMC (or QMCPACK) dataset [61] comes from an open-source Quantum Monte Carlo program designed for high-level *ab initio* calculations of the electronic structure in atoms, molecules, solids, and 2D nanomaterials [24]. The dataset we use comes from a $69 \times 69 \times 115 \times 288$ Einspline dataset in QMC. We choose the 10th orbit among all the 288 orbitals as the 3D scaler field of interest.

The results of this 3D dataset are presented in Fig. 6 and Tab. 4. As

shown in Tab. 4, we observe that at the original resolution, there are some reasonable alignments between the two methods. Specifically, in the entire domain, 80% of CPE-MFA critical points align with 64% of TTK-MFA critical points, giving a Jaccard index of 0.55 between the two sets of critical points.

To facilitate clearer observation, we display results only within specific value ranges. For block A in Fig. 6, points with value in $[-6.4e^{-6}, 1e^{-3}]$ are shown. For block B, the range is $[-2.5e^{-5}, 2e^{-3}]$, and for block C, it is $[-6.3e^{-6}, 6.3e^{-5}]$. As shown in Fig. 6 A*, B*, and C* respectively, upsampling with ratio 10^3 greatly reduces the discrepancies.

For an in-depth investigation, for the discrepancies observed with an upsampling ratio at 10^3, we could apply a higher upsampling ratio at 10^6 for TTK-MFA to obtain further improved critical point alignment. For instance, within the red cube 1 from Fig. 6 A* at an upsampling ratio of 10^3, there are unaligned points from both TTK-MFA and CPE-MFA. Upon increasing the sampling ratio to 10^6, all points from CPE-MFA align, the unaligned TTK-MFA points are very close to an aligned CPE-MFA point, as shown in Fig. 6 1*. Similar results appear in 2 vs. 2*. In cube 3*, corresponding to the red cube 3, the unaligned point from CPE-MFA and TTK-MFA are almost aligned.

Highlighted results. The above results suggest that upsampling effectively bridges the gap between the sampled PL-dataset used by TTK-MFA and the continuous model used by CPE-MFA. This experiment confirms the efficacy of CPE-MFA in identifying critical points from MFA.
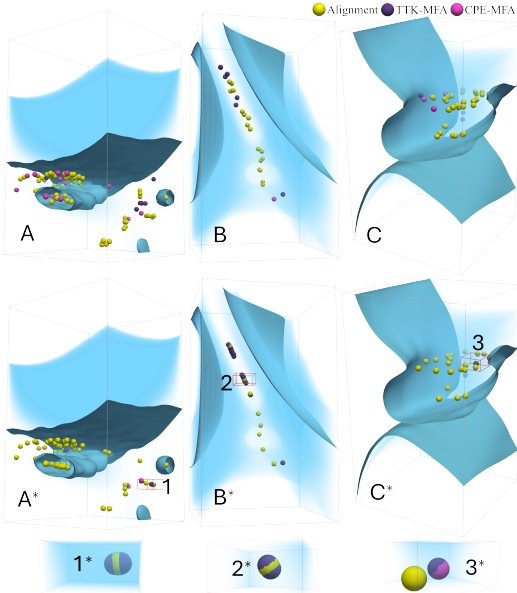
Figure 6: QMC dataset with critical points identified by CPE-MFA and TTK-MFA. Top: critical points from blocks A, B, and C, respectively. Middle: critical points from upsampled blocks A, B, and C (labeled as A*, B*, and C*), respectively, with a ratio of 10^3. Bottom: better critical point alignments 1*, 2*, 3* (in regions 1, 2, 3, respectively) after upsampling at a ratio of 10^6.

5.5 Rayleigh-Taylor Instability Dataset

When two fluids of different densities interact with each other, and the lighter fluid pushes the heavier fluid under constant acceleration, the phenomenon called the Rayleigh-Taylor instability [31] emerges. We use a 3D RTI dataset generated by simulating this instability with a CFDNS [30] Navier-Stokes solver. We use the velocity vector magnitude as the scalar field of interest. We sample a single time step within a domain of size $144 \times 256 \times 256$.

Fig. 7 and Tab. 4 display the results of CPE-MFA and TTK-MFA across all three blocks. This is a particularly challenging dataset due to the complexity of the flow, which gives rise to densely distributed critical points. We set the display range for point value in blocks A, B, and C to $[8, 16]$, $[2, 5]$, and $[8, 13]$, respectively. We observe some critical point alignment, see Fig. 7 A, B, and C. Upsampling with a ratio of 10^3 has proven effective in reducing the discrepancies between the two methods; see Fig. 7 A*, B*, and C*.

Upsampling further at a ratio of 10^6 aligns the critical points better in highlighted regions (red cubes 1, 2, and 3). All unaligned critical points in cube 1 become perfectly aligned after upsampling, shown in 1*. Compared with cube 2, the purple and pink points in 2* are almost overlapping. Although they do not align under the stringent threshold for the upsampled data, their positions are very close to one another. Similar results are observed in cubes 3 and 3*.

Highlighted results. The RTI dataset is topologically the most complex one in our experiments. We observe a reasonable alignment between the two sets of critical points extracted by CPE-MFA and TTK-MFA both qualitatively in Fig. 7 and qualitatively in Tab. 4, indicated by Jaccard indices between 0.44 and 0.58 in the original resolution. Upsampling for the TTK-MFA further improves the alignment, again, showcasing the efficacy of our method.

6 CONCLUSION AND DISCUSSION

We introduce CPE-MFA, the first framework in extracting critical points from multivariate functional approximation (MFA) models of large-scale datasets. Our framework processes the continuous MFA models directly without resampling or discretization, utilizing multithreading. The framework's ability to bypass spans devoid of critical
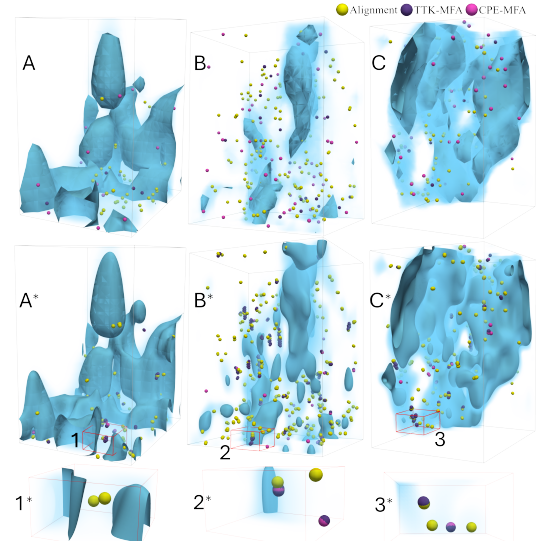

Figure 7: RTI dataset with critical points identified by CPE-MFA and TTK-MFA. Top: critical points from blocks A, B, and C, respectively. Middle: critical points from upsampled blocks A, B, and C (labeled as A*, B*, and C*), respectively, with a ratio of 10^3. Bottom: better critical point alignments 1*, 2*, 3* (in regions 1, 2, 3, respectively) after upsampling at a ratio of 10^6.

points further enhances its efficiency. Although our experiments focus on 2D and 3D datasets, our framework is dimension independent and generalizes easily to high-dimensional data. We demonstrate the effectiveness of CPE-MFA across various scientific datasets. This is the first step toward enabling continuous implicit models such as MFA to support topological data analysis at scale. In the future, we are interested in leveraging our current findings to extract topological descriptors from MFA models, such as contour trees and Morse complexes. We will also explore how critical points vary with different MFA approximation levels.

Our framework has its limitations. Since TTK is a widely-used tool for critical point extraction, we compare our CPE-MFA results against those obtained by TTK based on a PL interpolation of points sampled from an MFA. Alternatively, we could also compare against trilinear interpolation (e.g., following Globus et al [19]). Note that PL and trilinear interpolation work with points sampled from an MFA model, thus they do not perfectly approximate the continuous domain. They serve as references, not as ground truth for critical point extraction. For certain dataset, even though the Jaccard index is low between CPE-MFA and TTK-MFA, (almost) all critical points from CPE-MFA are aligned with TTK-MFA. However, TTK-MFA tends to extract more (spurious) critical points than CPE-MFA (c.f., Tab. 4) due to the noisy PL reconstruction of an MFA model (e.g., critical points lie in adjacent zigzags). Applying persistence simplification to the PL approximation might remove spurious critical points of TTK-MFA, thus increasing the Jaccard index. This is left for future work.

Recall a critical point x is *isolated* if there is a neighborhood U around x and x is the only critical point in U; otherwise, it is *non-isolated*. Our framework only handles isolated critical points, but not non-isolated ones (e.g., a manifold of critical points). Additionally, we omit boundary critical points at the moment, which might not have zero derivatives. Furthermore, the choice of initial points for Newton's method may affect the results. For some MFA models, we may need more initial points in a span to extract all critical points. Finally, Bézier clipping [48] (also known as interval Newton's method) offers an alternative approach for root finding with guaranteed convergence. It utilizes the convex hull properties of Bézier curves and iteratively subdivides the domain to narrow down the root-finding region. Replacing Newton's method with Bézier clipping in our framework will be an interesting future direction.

REFERENCES

[1] HepOnHPC. https://computing.fnal.gov/hep-on-hpc/. Accessed: 2024-06-28. 1

[2] RAPIDS2. https://rapids.lbl.gov/. Accessed: 2024-06-28. 1

[3] Seahorce. https://seahorce-scidac.github.io/. Accessed: 2024-06-28. 1

[4] M. Allili, D. Corriveau, S. Derivière, M. Ethier, and T. Kaczynski. Detecting critical regions in multidimensional data sets. *Computers and Mathematics with Applications*, 61(2):499–512, 2011. doi: 10.1016/j.camwa.2010.11.029 3

[5] K. L. Anderson, J. S. Anderson, S. Palande, and B. Wang. Topological data analysis of functional MRI connectivity in time and space domains. In *Connectomics in NeuroImaging: Second International Workshop, CNI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings*, pp. 67–77. Springer-Verlag, Berlin, Heidelberg, 2018. doi: 10.1007/978-3-030-00755-3_8 2

[6] T. F. Banchoff. Critical points and curvature for embedded polyhedral surfaces. *The American Mathematical Monthly*, 77(5):475–485, 1970. doi: 10.1080/00029890.1970.11992523 2, 12

[7] H. Bhatia, A. G. Gyulassy, V. Lordi, J. E. Pask, V. Pascucci, and P.-T. Bremer. TopoMS: Comprehensive topological exploration for molecular and condensed-matter systems. *Journal of computational chemistry*, 39(16):936–952, 2018. doi: 10.1002/jcc.25181 2

[8] A. Bock, H. Doraiswamy, A. Summers, and C. Silva. TopoAngler: Interactive topology-based extraction of fishes. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):812–821, 2018. doi: 10.1109/TVCG.2017.2743980 2

[9] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75–94, Feb. 2003. doi: 10.1016/S0925-7721(02)00093-7 2

[10] J. H. Chen, A. Choudhary, B. De Supinski, M. DeVries, E. R. Hawkes, S. Klasky, W.-K. Liao, K.-L. Ma, J. Mellor-Crummey, N. Podhorszki, et al. Terascale direct numerical simulations of turbulent combustion using S3D. *Computational Science & Discovery*, 2(1):015001, 2009. doi: 10.1088/1749-4699/2/1/015001 6

[11] W. Chen, M. Liu, Q. Zhan, Y. Tan, E. Meijering, M. Radojević, and Y. Wang. Spherical-patches extraction for deep-learning-based critical points detection in 3D neuron microscopy images. *IEEE Transactions on Medical Imaging*, 40(2):527–538, 2021. doi: 10.1109/TMI.2020.3031289 2

[12] C. De Boor. *A practical guide to splines*, vol. 27 of *Applied Mathematical Sciences*. Springer-Verlag New York, revised ed., 2001. 1, 2

[13] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Morse-Smale complexes for piecewise linear 3-manifolds. In *Proc. 19th Annual Symposium on Computational Geometry*, pp. 361–370. ACM, 2003. doi: 10.1145/777792.777846 2

[14] H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse complexes for piecewise linear 2-manifolds. *Discrete & Computational Geometry*, 30(1):87–107, 2003. doi: 10.1007/s00454-003-2926-5 2

[15] H. Edelsbrunner and J. L. Harer. *Computational topology: an introduction*. American Mathematical Society, 2010. 1, 12

[16] R. Forman. A user's guide to discrete Morse theory. *Séminaire Lotharingien de Combinatoire*, 48:B48c, 35, 2002. 2

[17] T. Fric and A. Roshko. Vortical structure in the wake of a transverse jet. *Journal of Fluid Mechanics*, 279:1–47, 1994. doi: 10.1017/S0022112094003800 6

[18] U. Fugacci, C. Landi, and H. Varlı. Critical sets of PL and discrete Morse theory: A correspondence. *Computers & Graphics*, 90:43–50, 2020. doi: 10.1016/j.cag.2020.05.020 2

[19] A. Globus, C. Levit, and T. Lasinski. A tool for visualizing the topology of three-dimensional vector fields. In *Proceeding Visualization '91*, pp. 33–40, 1991. doi: 10.1109/VISUAL.1991.175773 9

[20] R. Grout, A. Gruber, C. S. Yoo, and J. Chen. Direct numerical simulation of flame stabilization downstream of a transverse fuel jet in crossflow. *Proc. Combustion Institute*, 33(1):1629–1637, 2011. doi: 10.1016/j.proci.2010.06.013 6

[21] C. Heine, H. Leitte, M. Hlawitschka, F. Iuricich, L. De Floriani, G. Scheuermann, H. Hagen, and C. Garth. A survey of topology-based methods in visualization. *Computer Graphics Forum (CGF)*, 35(3):643–667, 2016. doi: 10.1111/cgf.12933 2

[22] J. Helman and L. Hesselink. Automated analysis of fluid flow topology. In S. S. Fisher and W. E. Robbins, eds., *Three-Dimensional Visualization and Display Technologies*, vol. 1083, pp. 144 – 152. International Society for Optics and Photonics, SPIE, 1989. doi: 10.1117/12.952882 2

[23] M. Jansen and P. Oonincx. *Second generation wavelets and applications*. Springer, London, 2005. doi: 10.1007/1-84628-140-7 1

[24] J. Kim, A. D. Baczewski, T. D. Beaudet, A. Benali, M. C. Bennett, M. A. Berrill, N. S. Blunt, E. J. L. Borda, M. Casula, D. M. Ceperley, S. Chiesa, B. K. Clark, R. C. Clay, K. T. Delaney, M. Dewing, K. P. Esler, H. Hao, O. Heinonen, P. R. C. Kent, J. T. Krogel, I. Kylänpää, Y. W. Li, M. G. Lopez, Y. Luo, F. D. Malone, R. M. Martin, A. Mathuriya, J. McMinis, C. A. Melton, L. Mitas, M. A. Morales, E. Neuscamman, W. D. Parker, S. D. P. Flores, N. A. Romero, B. M. Rubenstein, J. A. R. Shea, H. Shin, L. Shulenburger, A. F. Tillack, J. P. Townsend, N. M. Tubman, B. V. D. Goetz, J. E. Vincent, D. C. Yang, Y. Yang, S. Zhang, and L. Zhao. QMCPACK: an open source ab initio quantum Monte Carlo package for the electronic structure of atoms, molecules and solids. *Journal of Physics: Condensed Matter*, 30(19):195901, 2018. doi: 10.1088/1361-648X/aab9c3 8

[25] Y. Lavin, R. Batra, and L. Hesselink. Feature comparisons of vector fields using earth mover's distance. In *Proceedings Visualization '98*, pp. 103–109, 1998. doi: 10.1109/VISUAL.1998.745291 2

[26] D. Lenz, R. Yeh, V. Mahadevan, I. Grindeanu, and T. Peterka. Customizable adaptive regularization techniques for B-spline modeling. *Journal of Computational Science*, 71:102037, 2023. doi: 10.1016/j.jocs.2023.102037 1

[27] D. Lenz, R. Yeh, V. Mahadeven, I. Grindeanu, and T. Peterka. Adaptive regularization of B-spline models for scientific data. In D. Groen, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot, eds., *Computational Science – ICCS 2022*, pp. 150–163. Springer International Publishing, Cham, 2022. doi: 10.1007/978-3-031-08751-6_11 1

[28] W.-c. Li, B. Vallet, N. Ray, and B. Levy. Representing higher-order singularities in vector fields on piecewise linear surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1315–1322, 2006. doi: 10.1109/TVCG.2006.173 2

[29] H. Lin, T. Maekawa, and C. Deng. Survey on geometric iterative methods and their applications. *Computer-Aided Design*, 95:40 – 51, 2018. doi: 10.1016/j.cad.2017.10.002 1

[30] D. Livescu, J. Mohd-Yusof, M. Petersen, and J. Grove. CFDNS: a computer code for direct numerical simulation of turbulent flows. *Los Alamos National Laboratory Technical Report No. LA-CC-09-100*, 2009. 9

[31] D. Livescu, J. R. Ristorcelli, M. R. Petersen, and R. A. Gore. New phenomena in variable-density Rayleigh–Taylor turbulence. *Physica Scripta*, 2010(T142):014015, Dec. 2010. doi: 10.1088/0031-8949/2010/T142/014015 9

[32] V. Mahadevan, D. Lenz, I. Grindeanu, and T. Peterka. Accelerating multivariate functional approximation computation with domain decomposition techniques. In J. Mikyška, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. Sloot, eds., *Computational Science – ICCS 2023*, pp. 89–103. Springer Nature Switzerland, Cham, 2023. doi: 10.1007/978-3-031-35995-8_7 1

[33] Z. Majdisova and V. Skala. Radial basis function approximations: comparison and applications. *Applied Mathematical Modelling*, 51:728–743, Nov. 2017. doi: 10.1016/j.apm.2017.07.033 1

[34] J. N. P. Martel, D. B. Lindell, C. Z. Lin, E. R. Chan, M. Monteiro, and G. Wetzstein. ACORN: adaptive coordinate networks for neural scene representation. *ACM Transactions on Graphics*, 40(4):58:1–58:13, 2021. 1

[35] T. Martin, E. Cohen, and R. Kirby. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. *Computer Aided*

Geometric Design, 26(6):648–664, Aug. 2009. Solid and Physical Modeling 2008. doi: 10.1016/j.cagd.2008.09.008 1

[36] W. Martin and E. Cohen. Representation and extraction of volumetric attributes using trivariate splines: a mathematical framework. In *Proc. the Sixth ACM Symposium on Solid Modeling and Applications*, SMA '01, pp. 234–240. Association for Computing Machinery, New York, NY, USA, 2001. doi: 10.1145/376957.376984 1

[37] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proc. European Conference on Computer Vision*, pp. 405–421, 2020. 1

[38] J. Milnor. *Morse Theory*, vol. 51. Princeton University Press, Princeton, 1963. doi: 10.1515/9781400881802 2, 3

[39] R. B. Neale, C.-C. Chen, A. Gettelman, P. H. Lauritzen, S. Park, D. L. Williamson, A. J. Conley, R. Garcia, D. Kinnison, J.-F. Lamarque, et al. Description of the NCAR community atmosphere model (cam 5.0). *NCAR Technical Note NCAR/TN-486+STR, National Center for Atmospheric Research, Boulder, Colorado, USA*, 2010. 6

[40] M. Olejniczak, A. Severo Pereira Gomes, and J. Tierny. A topological data analysis perspective on noncovalent interactions in relativistic calculations. *International Journal of Quantum Chemistry*, 120(8):e26133, Dec. 2019. doi: 10.1002/qua.26133 2

[41] S. Park and K. Lee. High-dimensional trivariate NURBS representation for analyzing and visualizing fluid flow data. *Computers & Graphics*, 21(4):473–482, 1997. doi: 10.1016/S0097-8493(97)00023-X 1

[42] T. Peterka, Y. Nashed, I. Grindeanu, V. Mahadevan, R. Yeh, and D. Lenz. Multivariate functional approximation of scientific data. In *In Situ Visualization for Computational Science*, pp. 375–397. Springer, Cham, 2022. doi: 10.1007/978-3-030-81627-8_17 1, 2, 6

[43] T. Peterka, Y. S. Nashed, I. Grindeanu, V. S. Mahadevan, R. Yeh, and X. Tricoche. Foundations of multivariate functional approximation for scientific data. In *2018 IEEE 8th symposium on large data analysis and visualization (LDAV)*, pp. 61–71, 2018. doi: 10.1109/LDAV.2018.8739195 1, 4

[44] L. Piegl and W. Tiller. *The NURBS book*. Springer-Verlag, 2 ed., 1997. doi: 10.1007/978-3-642-59223-2 2, 3

[45] G. Reeb. Sur les points singuliers d'une forme de pfaff completement integrable ou d'une fonction numerique [on the singular points of a completely integrable pfaff form or of a numerical function]. *Comptes Rendus Acad. Sciences Paris*, 222:847–849, 1946. 2

[46] D. F. Rogers. *An introduction to NURBS: with historical perspective*. Morgan Kaufmann, 2001. doi: 10.1016/B978-1-55860-669-2.X5000-3 1

[47] H.-P. Schwefel. *Numerical optimization of computer models*. John Wiley & Sons, Inc., USA, 1981. 5

[48] T. Sederberg and T. Nishita. Curve intersection using Bézier clipping. *Computer-Aided Design*, 22(9):538–549, 1990. doi: 10.1016/0010-4485(90)90039-F 9

[49] N. Shivashankar, P. Pranav, V. Natarajan, R. van de Weygaert, E. P. Bos, and S. Rieder. Felix: A topology based framework for visual exploration of cosmic filaments. *IEEE Transactions on Visualization and Computer Graphics*, 22(6):1745–1759, June 2015. doi: 10.1109/TVCG.2015.2452919 2

[50] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. Advances in Neural Information Processing Systems*, 2020. 1

[51] V. Sitzmann, M. Zollhöfer, and G. Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *Proc. Advances in Neural Information Processing Systems*, pp. 1119–1130, 2019. 1

[52] V. Skala and M. Smolik. A new approach to vector field interpolation, classification and robust critical points detection using radial basis functions. In R. Silhavy, ed., *Cybernetics and Algorithms in Intelligent Systems*, pp. 109–115. Springer, Cham, 2019. doi: 10.1007/978-3-319-91192-2_12 2

[53] D. Smirnov and D. Morozov. Triplet merge trees. In *Topological Methods in Data Analysis and Visualization V*, TopoInVis 2017, pp. 19–36. Springer, 2020. doi: 10.1007/978-3-030-43036-8_2 2

[54] T. Sousbie. The persistent cosmic web and its filamentary structure – I. Theory and implementation. *Monthly Notices of the Royal Astronomi-*

cal Society, 414(1):350–383, 06 2011. doi: 10.1111/j.1365-2966.2011.18394.x 2

[55] J. Sun, D. Lenz, H. Yu, and T. Peterka. Scalable volume visualization for big scientific data modeled by functional approximation. In *2023 IEEE International Conference on Big Data*, BigData, pp. 905–914, 2023. doi: 10.1109/BigData59044.2023.10386434 1

[56] J. Sun, D. Lenz, H. Yu, and T. Peterka. MFA-DVR: direct volume rendering of MFA models. *Journal of Visualization*, 27(1):109–126, Oct. 2024. doi: 10.1007/s12650-023-00946-y 1

[57] J. Tierny, G. Favelier, J. A. Levine, C. Gueunet, and M. Michaux. The topology toolkit. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):832–842, Jan. 2018. doi: 10.1109/TVCG.2017.2743938 4

[58] J. Vidal, P. Guillou, and J. Tierny. A progressive approach to scalar field topology. *IEEE Transactions on Visualization and Computer Graphics*, 27(6):2833–2850, Feb. 2021. doi: 10.1109/TVCG.2021.3060500 2, 4

[59] H. Wendland. *Scattered Data Approximation*, vol. 17 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, 2004. doi: 10.1017/CBO9780511617539 1

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

[61] K. Zhao, S. Di, X. Lian, S. Li, D. Tao, J. Bessac, Z. Chen, and F. Cappello. SDRBench: scientific data reduction benchmark for lossy compressors. In *2020 IEEE International Conference on Big Data*, BigData, pp. 2716–2724, 2020. doi: 10.1109/BigData50022.2020.9378449 8

A DETAILS ON SCIENTIFIC DATASETS

For each scientific dataset, we extract three blocks of data to highlight the extracted critical points from both CPE-MFA and TTK-MFA. The sizes and locations of these blocks are detailed in Tab. 5.

Dataset	Size	Block A
CESM	3600×1800	$[0, 720] \times [989, 1349]$
S3D	704×540	$[422, 633] \times [270, 431]$
QMC	$69 \times 69 \times 115$	$[48, 68] \times [48, 68] \times [40, 74]$
RTI	$144 \times 256 \times 256$	$[74, 90] \times [66, 78] \times [88, 98]$

Dataset	Block B	Block C
CESM	$[900, 1620] \times [450, 810]$	$[2159, 2879] \times [1079, 1439]$
S3D	$[176, 387] \times [189, 350]$	$[70, 281] \times [28, 189]$
QMC	$[24, 44] \times [24, 44] \times [68, 102]$	$[41, 61] \times [7, 27] \times [34, 68]$
RTI	$[57, 73] \times [23, 35] \times [156, 166]$	$[44, 60] \times [51, 63] \times [204, 214]$

Table 5: The sizes of four scientific datasets and the locations of their corresponding blocks.

B CRITICAL POINT EXTRACTION IN THE PL SETTING

In the PL setting, Banchoff [6] proposed a method to detect critical points, and Edelsbrunner and Harer [15] gave a detailed description as follows. Assuming the input data is modeled as a scalar field f defined on a PL manifold $\mathbb{M}$, the sublevel set of the data is $f^{-1}(-\infty, t] = \{x \in \mathbb{M} \mid f(x) \leq t\}$ for some $t \in \mathbb{R}$. A *proper face* τ of a simplex $\sigma \in \mathbb{M}$ is the convex hull of a non-empty, strict subset of vertices in σ. We call σ a *proper coface* of τ and denote it by $\tau < \sigma$. The *star* $St(v)$ of a vertex $v \in \mathbb{M}$ is a set of its cofaces $St(v) = \{\sigma \in \mathbb{M} \mid v < \sigma\}$. The smallest complex that contains the star is the *closed star* $\overline{St}(v)$. The *link* $Lk(v)$ of v is the set of simplices in the closed star but not in the star. The link can be thought as the boundary of a small neighborhood around v. The *lower link* $Lk^-(v)$ is a set of simplices in the link whose vertices all have values smaller than v:

$$Lk^-(v) = \{\sigma \in Lk(v) \mid \forall u \in \sigma, f(u) < f(v)\}. \quad (10)$$

Correspondingly, the *upper link* $Lk^+(v)$ is a set of simplices in the link whose vertices have values larger than v. Vertex v is a *PL regular point* if both $Lk^-(v)$ and $Lk^+(v)$ are simply connected. Otherwise, it is a *PL critical point*.

In a PL 2-manifold, there are three types of critical points: local minimum ($Lk^-(v) = \emptyset$), local maximum ($Lk^+(v) = \emptyset$), and saddles (when $Lk^-(v)$ or $Lk^+(v)$ is not simply connected). We show examples of lower links in Fig. 8. For a PL 3-manifold, there are four types of critical points: local minimum, local maximum, and two different types of saddles.
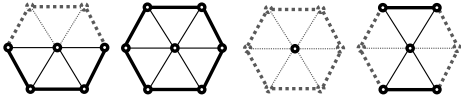

Figure 8: From left to right: the lower link of a regular point, a local maximum, a local minimum, and a saddle.

C PSEUDOCODE FOR SPATIAL HASHING

We provide the pseudocode for spatial hashing used to remove duplicated critical points in Sec. 4.3.

Algorithm 2 Removing Duplicated Critical Points

Require: A set of critical points X and a threshold τ.
Ensure: A set of critical points Q after duplication removal.
1: $Y = \emptyset$, a duplicated set of critical points to be removed from X.
2: **for all** $x \in X$ **do**
3:     $k = \frac{x}{20\tau}$
4:     duplicate_label = false
5:     **for** $i \in [1, d]$ **do**
6:        $I_{i,1} = \lfloor k_i \rfloor$, $I_{i,2} = \lceil k_i \rceil$
7:     We obtain all hash indices in a set $\mathcal{I} = \{I = (I_1, \cdots, I_d)\}$ where each dimension I_i is chosen from $I_{i,1}$ or $I_{i,2}$.
8:     **for all** hash index $I \in \mathcal{I}$ **do**
9:        V = boost::hash_combine(I)
10:        **for all** point y in the hash buckets corresponding to V **do**
11:           **if** $||x - y|| < \tau$ **then**
12:             Identify x as a duplicated critical point.
13:             duplicate_label = true
14:             $Y = Y \bigcup x$.
15:             Goto line 16, exit the two innermost for loops.
16:     **if** duplicate_label is false **then**
17:        Register x to all the hash buckets corresponding to $\mathcal{I}$.
18: **return** $Q = X \setminus Y$

D DETAILED COMPLEXITY ANALYSIS

In span filtration, for every dimension's first derivative, $p(p+1)^{d-1}$ control points are utilized in every span. Let n denote the number of spans. In the experiments, we consistently maintain $p \geq 2$. Within each span, it is necessary to evaluate all the utilized control points. The cost of span filtration is $\mathcal{O}(dp^d n)$. In Newton's method, the computation cost for every element in the Hessian matrix is $\mathcal{O}(p^d + dp^3)$. Each iteration of Newton's method involves computing the Hessian and gradient at $\mathcal{O}(d^2(p^d + dp^3))$ and solving linear system at $\mathcal{O}(d^3)$. Since $(p+1)^d$ initial points are used in each span, the cost of finding critical points in all spans is $\mathcal{O}(i_{max} d^2(p^{2d} + dp^{3+d})n)$. During spatial hashing, registration and comparison of at most $(p+1)^d$ points occur 2^d times per span. The spatial hashing takes $\mathcal{O}((2p)^d dn)$. In practical applications, p and d are always 2 or 3 which are much smaller than n (and can be treated as constants). The complexity of span filtration and spatial hashing are smaller than finding all critical points via Newton's method. The overall time complexity of CPE-MFA is $\mathcal{O}(i_{max} d^2(p^{2d} + dp^{3+d})n)$, which matches the time complexity of finding critical points in all spans. In all the experiments conducted, more than 95% of the time was spent finding critical points using Newton's method, confirming the analysis that this step dominates the overall time complexity.