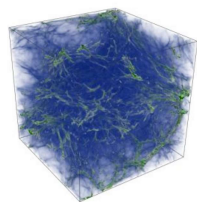


Principal Geodesic Analysis Of Merge Trees (and Persistence Diagrams)

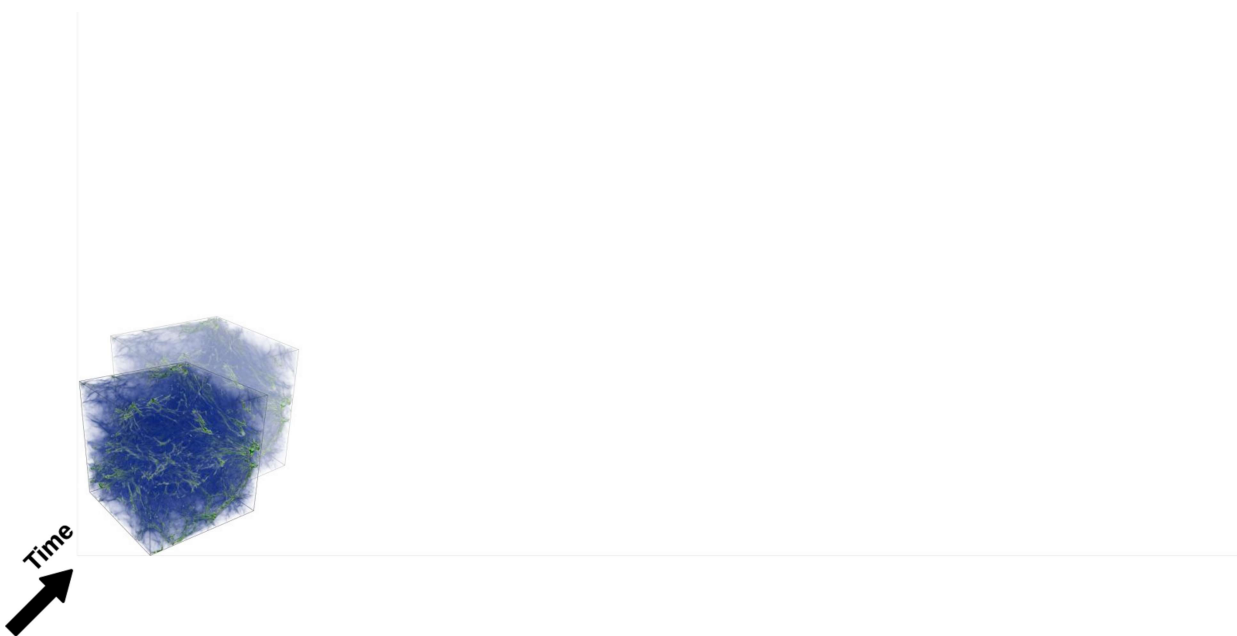
Mathieu Pont, Jules Vidal, Julien Tierny



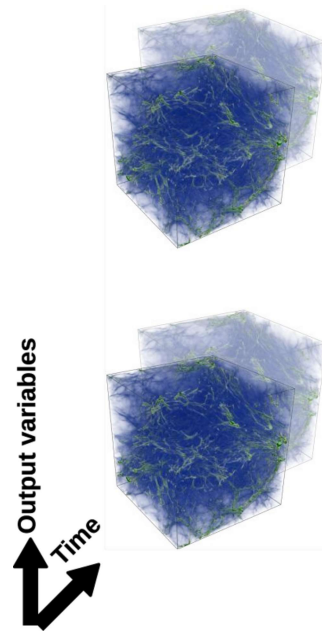
Topological data reduction



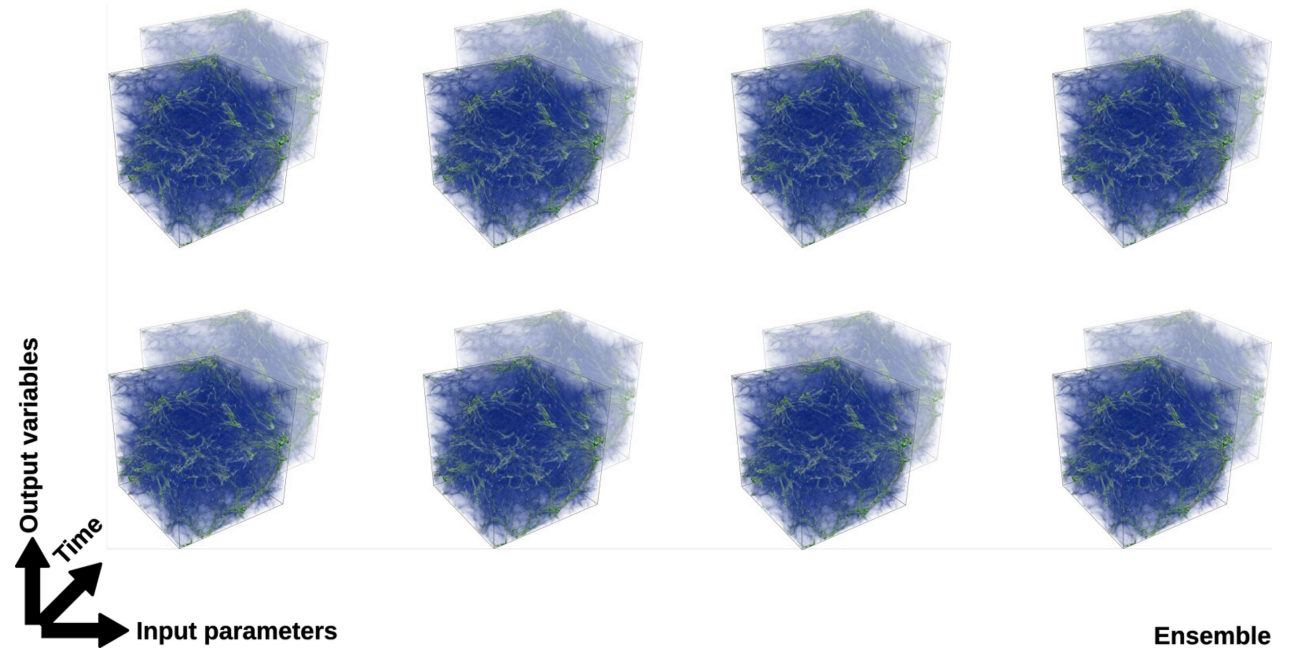
Topological data reduction



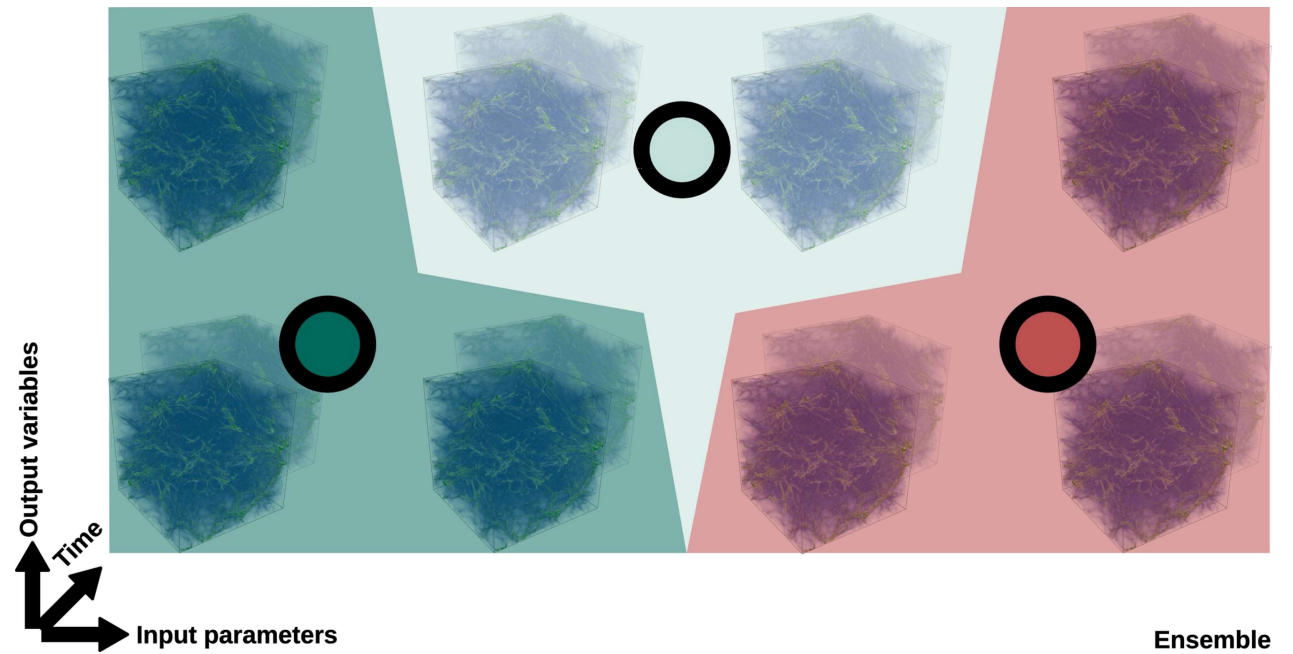
Topological data reduction



Topological data reduction



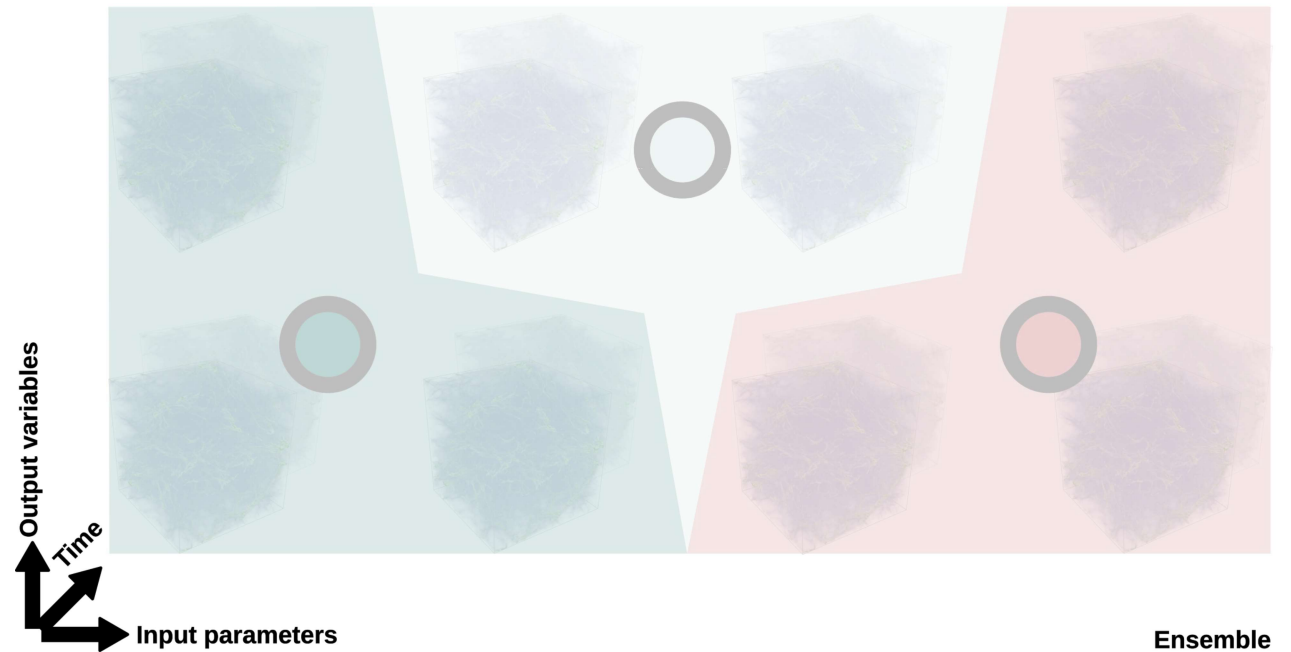
Topological data reduction



Topological data reduction

- **Challenges**

- Significance
- Data size



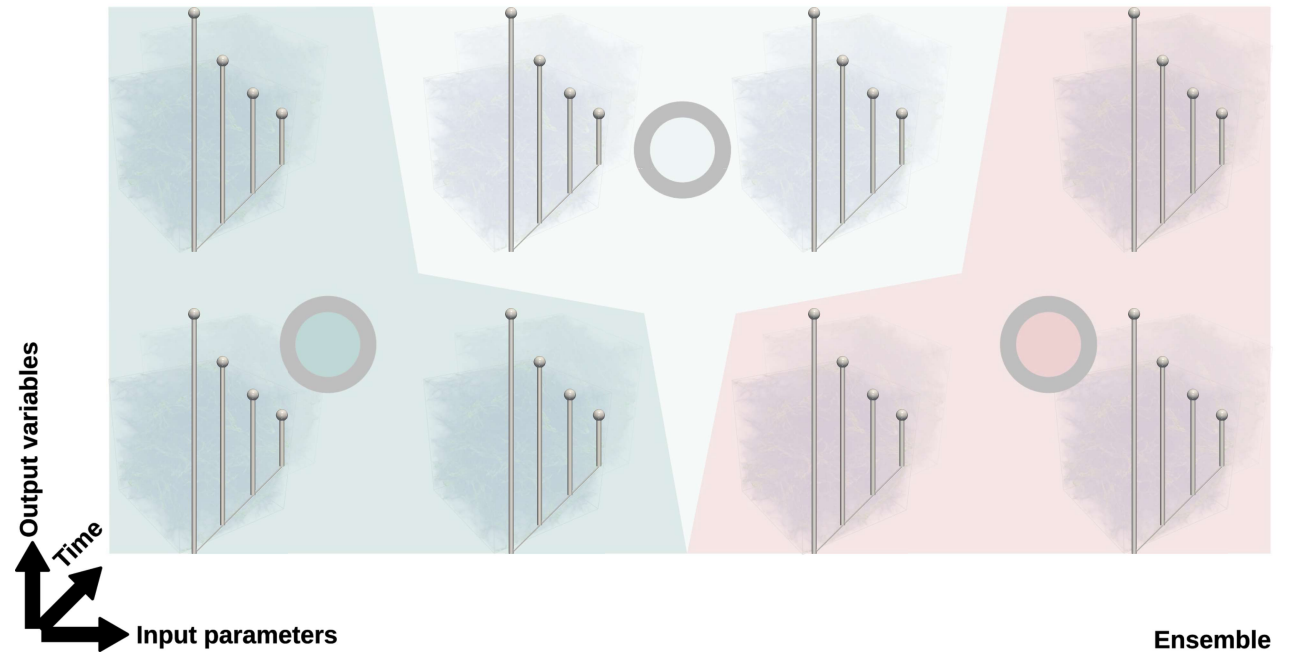
Topological data reduction

- **Challenges**

- Significance
- Data size

- **Strategy**

- Topological reduction



Topological data reduction

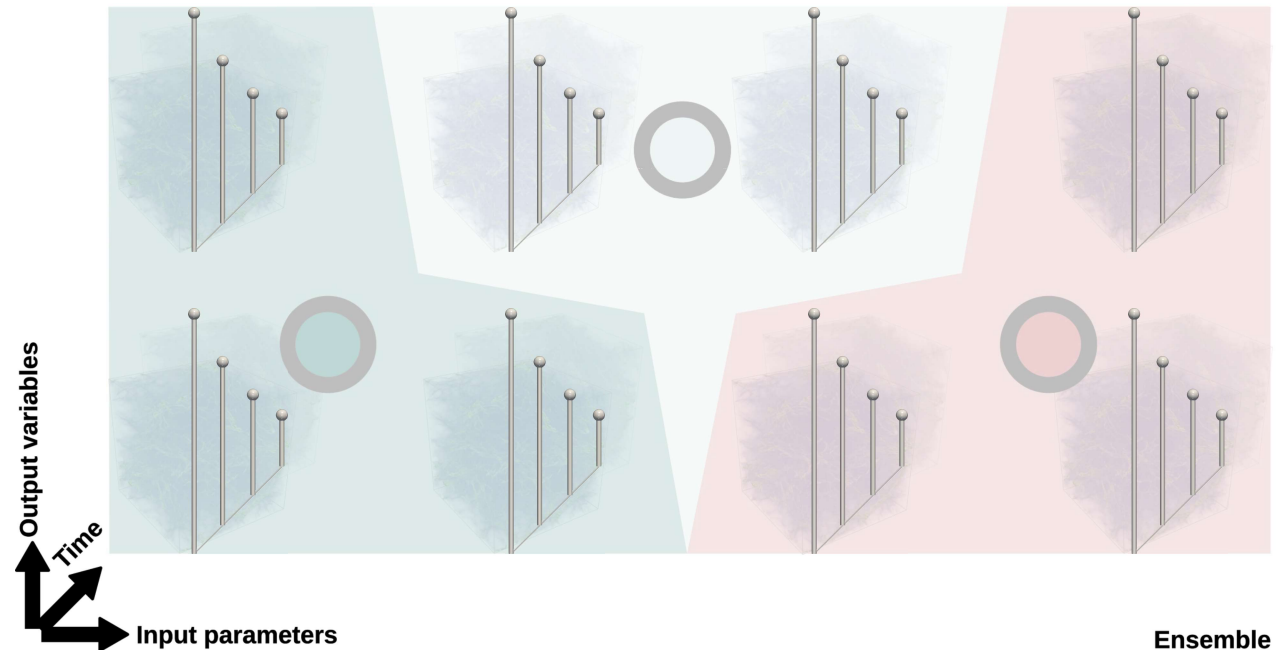
- **Challenges**

- Significance
- Data size

- **Strategy**

- Topological reduction

 <http://erc-tori.github.io/>



Metric spaces for Persistence diagrams

- Vast literature!

Stability of Persistence Diagrams

David Cohen-Steiner
Dartmouth College
North Carolina, USA
dcohen@dartmouth.edu

Herbert Edelsbrunner
Dartmouth College
Research Center, R2P
North Carolina, USA
edels@cs.dartmouth.edu

John Harer
Dartmouth College
Dartmouth College
North Carolina, USA
harer@math.dartmouth.edu

ABSTRACT
The persistence diagram of a real-valued function on a topological space is a multiset of points in the extended plane. We prove that small changes in the function only lead to small changes in the diagram. We apply this result to extending the homology of sets in a metric space and to comparing nested clustering persistence diagrams.

Categories and Subject Descriptors
F.2.2 Analysis of Algorithms and Problem Complexity; Non-numerical Algorithms and Problems; Geometric problems and computations; Computation on graphs; Graphs; G.2 Discrete Mathematics; Combinatorics—Counting problems

General Terms
Algorithms, Theory

Keywords
Combinatorial topology; combinatorial topology; homology groups; persistence; stability

1. INTRODUCTION
In this paper we consider multivalued functions on topological spaces and the concept of persistence in such situations and quantitative homology. More specifically, we track the topological invariance of a function in a metric space by persistence diagrams and study the stability of this invariant.

Mathematics Subject Classification
54B20, 54C20, 55M10, 55N10, 55N20, 55N30, 55N40, 55N65, 55N99, 57M10, 57M20, 57M30, 57M40, 57M50, 57M60, 57M70, 57M80, 57M90, 57N10, 57N20, 57N30, 57N40, 57N50, 57N60, 57N70, 57N80, 57N90, 58A10, 58A20, 58A30, 58A40, 58A50, 58A60, 58A70, 58A80, 58A90, 58B10, 58B20, 58B30, 58B40, 58B50, 58B60, 58B70, 58B80, 58B90, 58C10, 58C20, 58C30, 58C40, 58C50, 58C60, 58C70, 58C80, 58C90, 58D10, 58D20, 58D30, 58D40, 58D50, 58D60, 58D70, 58D80, 58D90, 58E10, 58E20, 58E30, 58E40, 58E50, 58E60, 58E70, 58E80, 58E90, 58F10, 58F20, 58F30, 58F40, 58F50, 58F60, 58F70, 58F80, 58F90, 59A10, 59A20, 59A30, 59A40, 59A50, 59A60, 59A70, 59A80, 59A90, 59B10, 59B20, 59B30, 59B40, 59B50, 59B60, 59B70, 59B80, 59B90, 59C10, 59C20, 59C30, 59C40, 59C50, 59C60, 59C70, 59C80, 59C90, 59D10, 59D20, 59D30, 59D40, 59D50, 59D60, 59D70, 59D80, 59D90, 59E10, 59E20, 59E30, 59E40, 59E50, 59E60, 59E70, 59E80, 59E90, 59F10, 59F20, 59F30, 59F40, 59F50, 59F60, 59F70, 59F80, 59F90, 59G10, 59G20, 59G30, 59G40, 59G50, 59G60, 59G70, 59G80, 59G90, 59H10, 59H20, 59H30, 59H40, 59H50, 59H60, 59H70, 59H80, 59H90, 59J10, 59J20, 59J30, 59J40, 59J50, 59J60, 59J70, 59J80, 59J90, 59K10, 59K20, 59K30, 59K40, 59K50, 59K60, 59K70, 59K80, 59K90, 59L10, 59L20, 59L30, 59L40, 59L50, 59L60, 59L70, 59L80, 59L90, 59M10, 59M20, 59M30, 59M40, 59M50, 59M60, 59M70, 59M80, 59M90, 59N10, 59N20, 59N30, 59N40, 59N50, 59N60, 59N70, 59N80, 59N90, 59O10, 59O20, 59O30, 59O40, 59O50, 59O60, 59O70, 59O80, 59O90, 59P10, 59P20, 59P30, 59P40, 59P50, 59P60, 59P70, 59P80, 59P90, 59Q10, 59Q20, 59Q30, 59Q40, 59Q50, 59Q60, 59Q70, 59Q80, 59Q90, 59R10, 59R20, 59R30, 59R40, 59R50, 59R60, 59R70, 59R80, 59R90, 59S10, 59S20, 59S30, 59S40, 59S50, 59S60, 59S70, 59S80, 59S90, 59T10, 59T20, 59T30, 59T40, 59T50, 59T60, 59T70, 59T80, 59T90, 59U10, 59U20, 59U30, 59U40, 59U50, 59U60, 59U70, 59U80, 59U90, 59V10, 59V20, 59V30, 59V40, 59V50, 59V60, 59V70, 59V80, 59V90, 59W10, 59W20, 59W30, 59W40, 59W50, 59W60, 59W70, 59W80, 59W90, 59X10, 59X20, 59X30, 59X40, 59X50, 59X60, 59X70, 59X80, 59X90, 59Y10, 59Y20, 59Y30, 59Y40, 59Y50, 59Y60, 59Y70, 59Y80, 59Y90, 59Z10, 59Z20, 59Z30, 59Z40, 59Z50, 59Z60, 59Z70, 59Z80, 59Z90, 60A10, 60A20, 60A30, 60A40, 60A50, 60A60, 60A70, 60A80, 60A90, 60B10, 60B20, 60B30, 60B40, 60B50, 60B60, 60B70, 60B80, 60B90, 60C10, 60C20, 60C30, 60C40, 60C50, 60C60, 60C70, 60C80, 60C90, 60D10, 60D20, 60D30, 60D40, 60D50, 60D60, 60D70, 60D80, 60D90, 60E10, 60E20, 60E30, 60E40, 60E50, 60E60, 60E70, 60E80, 60E90, 60F10, 60F20, 60F30, 60F40, 60F50, 60F60, 60F70, 60F80, 60F90, 60G10, 60G20, 60G30, 60G40, 60G50, 60G60, 60G70, 60G80, 60G90, 60H10, 60H20, 60H30, 60H40, 60H50, 60H60, 60H70, 60H80, 60H90, 60I10, 60I20, 60I30, 60I40, 60I50, 60I60, 60I70, 60I80, 60I90, 60J10, 60J20, 60J30, 60J40, 60J50, 60J60, 60J70, 60J80, 60J90, 60K10, 60K20, 60K30, 60K40, 60K50, 60K60, 60K70, 60K80, 60K90, 60L10, 60L20, 60L30, 60L40, 60L50, 60L60, 60L70, 60L80, 60L90, 60M10, 60M20, 60M30, 60M40, 60M50, 60M60, 60M70, 60M80, 60M90, 60N10, 60N20, 60N30, 60N40, 60N50, 60N60, 60N70, 60N80, 60N90, 60O10, 60O20, 60O30, 60O40, 60O50, 60O60, 60O70, 60O80, 60O90, 60P10, 60P20, 60P30, 60P40, 60P50, 60P60, 60P70, 60P80, 60P90, 60Q10, 60Q20, 60Q30, 60Q40, 60Q50, 60Q60, 60Q70, 60Q80, 60Q90, 60R10, 60R20, 60R30, 60R40, 60R50, 60R60, 60R70, 60R80, 60R90, 60S10, 60S20, 60S30, 60S40, 60S50, 60S60, 60S70, 60S80, 60S90, 60T10, 60T20, 60T30, 60T40, 60T50, 60T60, 60T70, 60T80, 60T90, 60U10, 60U20, 60U30, 60U40, 60U50, 60U60, 60U70, 60U80, 60U90, 60V10, 60V20, 60V30, 60V40, 60V50, 60V60, 60V70, 60V80, 60V90, 60W10, 60W20, 60W30, 60W40, 60W50, 60W60, 60W70, 60W80, 60W90, 60X10, 60X20, 60X30, 60X40, 60X50, 60X60, 60X70, 60X80, 60X90, 60Y10, 60Y20, 60Y30, 60Y40, 60Y50, 60Y60, 60Y70, 60Y80, 60Y90, 60Z10, 60Z20, 60Z30, 60Z40, 60Z50, 60Z60, 60Z70, 60Z80, 60Z90

David Cohen-Steiner
Herbert Edelsbrunner
John Harer

Frochet Means for Distributions of Persistence Diagrams

Katharin Tupper · York Miyabe · Saurabh Mukherjee · John Harer

Revised: 12 June 2017; Revised: 10 March 2018; Accepted: 5 June 2018
Published online: 12 July 2018
© Springer Science+Business Media, New York 2018

ABSTRACT
Given a distribution μ on persistence diagrams and observations $X_1, \dots, X_n \in \mathbb{R}^2$, we introduce an algorithm in this paper that estimates a Frochet mean from the set of diagrams X_1, \dots, X_n . If the underlying measure μ is a combination of Dirac masses $\mu = \sum_{i=1}^k \alpha_i \delta_{X_i}$, then we prove the algorithm converges to a local minimum and a law of large numbers result for the Frochet mean computed by the algorithm given observation drawn iid from μ . We illustrate the convergence of an empirical mean computed by the algorithm to a population mean by simulations from Gaussian random fields.

Keywords: Persistence diagram · Frochet mean · Topological data analysis · Alexander space · Persistent homology

1 Introduction
There has been a recent effort in topological data analysis (TDA) to incorporate ideas from stochastic modeling. Much of this work involves the study of random abstract simplicial complexes generated from stochastic processes [10, 11, 14, 21, 23].

K. Tupper
Department of Mathematics, University of Chicago, Chicago, IL, USA

Y. Miyabe
Department of Mathematics, University of Hawaii at Manoa, Honolulu, HI, USA

S. Mukherjee
Department of Statistical Science, Computer Science, and Mathematics, Institute for Genome Sciences and Policy, Duke University, Durham, NC, USA

J. Harer
Department of Mathematics, Computer Science and Statistical and Computing Engineering, Center for Systems Biology, Dana-Farber Cancer Institute, Boston, MA, USA

Large Scale computation of Means and Clusters for Persistence Diagrams using Optimal Transport

This Lacombe
Marius Corti
Sven Oudot

Revised: 13 Nov 2018

ABSTRACT
Persistence diagrams (PDs) are now routinely used to summarize the underlying topology of complex data. Despite recent algorithmic improvements, PDs is a limiting operation on the data that involves the computation of an Earth Mover Distance. This was recently recognized in a series of papers which show that the worst case of computing a PD can be computationally prohibitive. In this paper, we propose a new algorithm for computing PDs based on optimal transport. This algorithm finds the a transportation PD between optimal transport (OT) problems. Owing to, we can exploit recent computational advances in OT problem to a great extent, which together with entropy regularization can be used to solve the problem using the Sinkhorn algorithm and its variants. This results in a highly efficient algorithm that can compute PDs. We demonstrate the efficiency of our approach by comparing our findings with diagram metrics on several thousands of PDs, a task most hard before in the literature.

1 Introduction
Topological data analysis (TDA) has been used successfully in a wide array of applications, for instance in medical Oncology [6, 20, 11] or network Theory [4, 20]. Since its inception, computer vision [2, 4, 20] or to classify NBA players [14] [15]. The goal of TDA is to capture and analyze the complex topology of complex data sets, such as point cloud data [17].

This Lacombe
Marius Corti
Sven Oudot

Progressive Wasserstein Barycenters of Persistence Diagrams

Jake Vitek, Joseph Burke, and John Terry

Revised: 13 Oct 2019

ABSTRACT
This paper presents an efficient algorithm for the progressive approximation of Wasserstein barycenters of persistence diagrams with applications to the rapid analysis of growing data. Given a set of persistence diagrams, the proposed algorithm iteratively computes barycenters of the diagrams by successively adding new diagrams. The algorithm is designed to be efficient and scalable, and is able to handle large numbers of diagrams. The algorithm is implemented in the software package *Wasserstein Barycenters* [1].

1 Introduction
Topological Data Analysis (TDA) has been used successfully in a wide array of applications, for instance in medical Oncology [6, 20, 11] or network Theory [4, 20]. Since its inception, computer vision [2, 4, 20] or to classify NBA players [14] [15]. The goal of TDA is to capture and analyze the complex topology of complex data sets, such as point cloud data [17].

[Cohen-Steiner '05]

[Turner '14]

[Lacombe '18]

[Vidal '19]

Metric spaces for Persistence diagrams

• Vast literature!

Stability of Persistence Diagrams

David Cohen-Steiner
Dartmouth College
Lebanon, NH
cohen@cispy.dartmouth.edu

Herbert Edelsbrunner
Dartmouth College
Lebanon, NH
edels@cispy.dartmouth.edu

John Harer
Dartmouth College
Lebanon, NH
harer@math.dartmouth.edu

ABSTRACT
The persistence diagram of a real-valued function on a topological space is a collection of points in the extended plane. We prove that small changes in the function imply only small changes in the diagram. We apply this result to extending the homology of sets to a metric space and to comparing clustering persistence diagrams.

Categories and Subject Descriptors
F.2.2 Analysis of Algorithms and Problem Complexity: Non-numerical Algorithms and Problems: Geometric problems and computations; Computation on discrete structures; G.1 Discrete Mathematics: Combinatorics—Counting problems

General Terms
Algorithms, Theory

Keywords
Combinatorial topology; combinatorial topology; homology groups; persistence; stability

1. INTRODUCTION
In this paper we consider robust functions on topological spaces and the concept of persistence to study their qualitative and quantitative behavior. More precisely, we track the topological components of a function as we vary its persistence diagram and study the stability of this tracking.

Mathematical background. Topological spaces are often an intermediate step to the full classification of the space's structure and computing and/or comparing topological invariants of the space.

The first author was partially supported by NSF grant grant 02-054642 and by DARPA award number HR0010-02-2-0022. The third author was partially supported by NSF grant DMS-05-00270 and by DARPA award number HR0010-02-2-0022.

Permission to make digital or physical copies of this work for personal or classroom use is granted by copyright owner for users registered with ACM. For all other uses, permission should be obtained from ACM. Copyright 2005 ACM 0-91131-010-0/05/0005...

[Cohen-Steiner '05]

Discrete Comput. Geom. (2014), 31:44–70
DOI 10.1007/s00454-014-9592-1

Friechet Means for Distributions of Persistence Diagrams

Katharine Turner · York-Mingyeh Suen · Makarand Karjane · John Harer

Revised: 1 June 2017 / Revised: 10 March 2018 / Accepted: 5 August 2018 / Published online: 12 July 2018
© Springer Science+Business Media, Inc. 2018

ABSTRACT
Given a distribution μ on persistence diagrams and observations $X_1, \dots, X_n \in \mathbb{R}^2$, we introduce an algorithm in this paper that estimates a Fréchet mean from the set of diagrams X_1, \dots, X_n . If the underlying metric μ is a combination of three metrics $\mu = \mu_1 \oplus \mu_2 \oplus \mu_3$, then we prove the algorithm converges to a local minimum and a few of larger numbers result for a Fréchet mean computed by the algorithm given observation drawn from μ . We illustrate the convergence of an empirical mean computed by the algorithm to a population mean by simulations from Gaussian random fields.

Keyword: Persistence diagram · Fréchet mean · Topological data analysis · Alexander space · Persistent homology

1. Introduction
There has been a recent effort in topological data analysis (TDA) to incorporate ideas from stochastic modeling. Much of this work involved the study of random abstract simplicial complexes generated from stochastic processes [10, 11, 14, 22, 23].

K. Turner
Department of Mathematics, University of Chicago, Chicago, IL, USA
kturner@math.uchicago.edu

M. Karjane
Department of Mathematics, University of Hawaii at Manoa, Honolulu, HI, USA
makarand@math.hawaii.edu

Y.-M. Suen
Department of Statistics Science, Computer Science, and Mathematics, Department for Science, Science and Policy, Duke University, Durham, NC, USA
ysuen@duke.edu

J. Harer
Department of Mathematics, Computer Science and Statistical and Complex Engineering, Center for Systems Biology, Duke University, Durham, NC, USA
harer@math.duke.edu

Springer

[Turner '14]

Large Scale computation of Means and Clusters for Persistence Diagrams using Optimal Transport

This Lacombe Harer
John Harer
Sven Steiner
Sven Steiner, Paris, France

Mario Corti
CHER, ENSAE
mario.corti@ensae.fr

Sven Oudot
John Harer
sven.oudot@duke.edu

ABSTRACT
Persistence diagrams (PDs) are now routinely used to summarize the underlying topology of complex data. Despite several appealing representations, computing PDs is a limiting operation on the data because they are not generally given as an Euler characteristic. Indeed, this was recently mentioned in a series of papers which show that the worst case of computing a PD can be computationally prohibitive. We propose in this article a stable framework to carry out standard tasks on PDs. In a limiting operation on the data because they are not generally given as an Euler characteristic. Indeed, this was recently mentioned in a series of papers which show that the worst case of computing a PD can be computationally prohibitive. We propose in this article a stable framework to carry out standard tasks on PDs. In a limiting operation on the data because they are not generally given as an Euler characteristic. Indeed, this was recently mentioned in a series of papers which show that the worst case of computing a PD can be computationally prohibitive. We propose in this article a stable framework to carry out standard tasks on PDs.

1. Introduction
Topological data analysis (TDA) has been used successfully in a wide array of applications, for instance in medical oncology [2, 20, 21] or material science [16, 20]. More recently, computer vision [1, 2, 24] or to classify NBA players [18, 21]. The goal of TDA is to capture and analyze the complex topological structure of high-dimensional data sets.

Mario Corti
CHER, ENSAE
mario.corti@ensae.fr

Sven Oudot
John Harer
sven.oudot@duke.edu

arXiv:1605.06331v2 [math.ML] 13 Nov 2018

[Lacombe '18]

Progressive Wasserstein Barycenters of Persistence Diagrams

Jules Vidal, Joseph Burke, and John Tierney

ABSTRACT
Persistence diagrams (PDs) are now routinely used to summarize the underlying topology of complex data. Despite several appealing representations, computing PDs is a limiting operation on the data because they are not generally given as an Euler characteristic. Indeed, this was recently mentioned in a series of papers which show that the worst case of computing a PD can be computationally prohibitive. We propose in this article a stable framework to carry out standard tasks on PDs. In a limiting operation on the data because they are not generally given as an Euler characteristic. Indeed, this was recently mentioned in a series of papers which show that the worst case of computing a PD can be computationally prohibitive. We propose in this article a stable framework to carry out standard tasks on PDs.

1. Introduction
Topological data analysis (TDA) has been used successfully in a wide array of applications, for instance in medical oncology [2, 20, 21] or material science [16, 20]. More recently, computer vision [1, 2, 24] or to classify NBA players [18, 21]. The goal of TDA is to capture and analyze the complex topological structure of high-dimensional data sets.

arXiv:1807.04652v2 [cs.GR] 9 Oct 2019

[Vidal '19]

Principal Geodesic Analysis for Probability Measures under the Optimal Transport Metric

Vivian Seguy
Grazhac School of Information
Kwan University
vivian.seguy@ipr.ist.ugr.es, vivian.seguy@ku.ac.jp

Mario Corti
Grazhac School of Information
Kwan University
mario.corti@ensae.jp

ABSTRACT
Given a family of probability measures in \mathbb{R}^2 , the space of probability measures on \mathbb{R}^2 is a metric space. We prove in this paper that the principal geodesic analysis (PGA) can be used to study the principal components of the space of probability measures. We prove that the principal geodesic analysis (PGA) can be used to study the principal components of the space of probability measures. We prove that the principal geodesic analysis (PGA) can be used to study the principal components of the space of probability measures.

1. Introduction
Optimal transport distances (Villani, 2006), also known as Wasserstein or earth mover's distance, offer a general framework to compare probability measures supported on a metric space. The Wasserstein space $\mathcal{P}_p(\mathbb{R}^2)$ is a metric space of probability measures on \mathbb{R}^2 endowed with the Wasserstein distance. It is a natural space which has natural analogies to the space of probability measures. The principal geodesic analysis (PGA) can be used to study the principal components of the space of probability measures. We prove that the principal geodesic analysis (PGA) can be used to study the principal components of the space of probability measures.

Principal Geodesic Analysis for Probability Measures under the Optimal Transport Metric
Vivian Seguy
Grazhac School of Information
Kwan University
vivian.seguy@ipr.ist.ugr.es, vivian.seguy@ku.ac.jp

Mario Corti
Grazhac School of Information
Kwan University
mario.corti@ensae.jp

arXiv:1506.07944v2 [math.ML] 22 Nov 2015

[Seguy '15]

Contributions

- **Principal Geodesic Analysis (PGA) of Merge Trees (and Persistence diagrams)**

Contributions

- **Principal Geodesic Analysis (PGA) of Merge Trees (and Persistence diagrams)**
 - Approach
 - Algorithm
 - Implementation
 - Applications

Background

Input data

- Ensemble of scalar fields

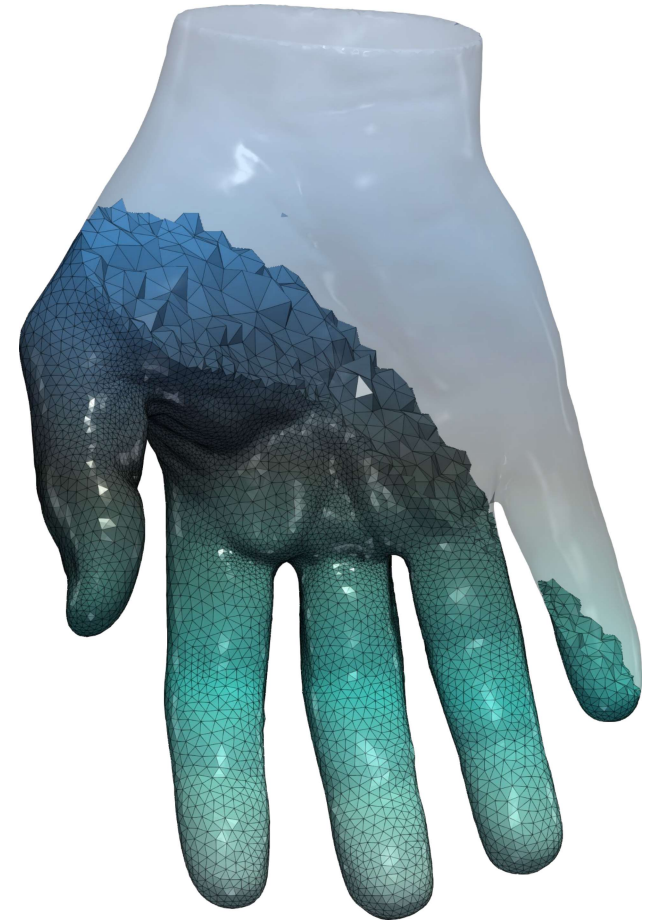
- $f_i: \mathcal{M} \rightarrow \mathbb{R}$
- $i \in \{1, \dots, N\}$



Input data

- Ensemble of scalar fields

- $f_i: \mathcal{M} \rightarrow \mathbb{R}$
- $i \in \{1, \dots, N\}$
- \mathcal{M} : simplicial complex



Persistence diagrams (PDs)

- **Sublevel set filtration**

- Topology changes at critical points
- Elder rule [Edelsbrunner&Harer '10]



Persistence diagrams (PDs)

- **Sublevel set filtration**

- Topology changes at critical points
- Elder rule [Edelsbrunner&Harer '10]



Persistence diagrams (PDs)

- **Sublevel set filtration**

- Topology changes at critical points
- Elder rule [Edelsbrunner&Harer '10]



Persistence diagrams (PDs)

- **Sublevel set filtration**
 - Topology changes at critical points
 - Elder rule [Edelsbrunner&Harer '10]



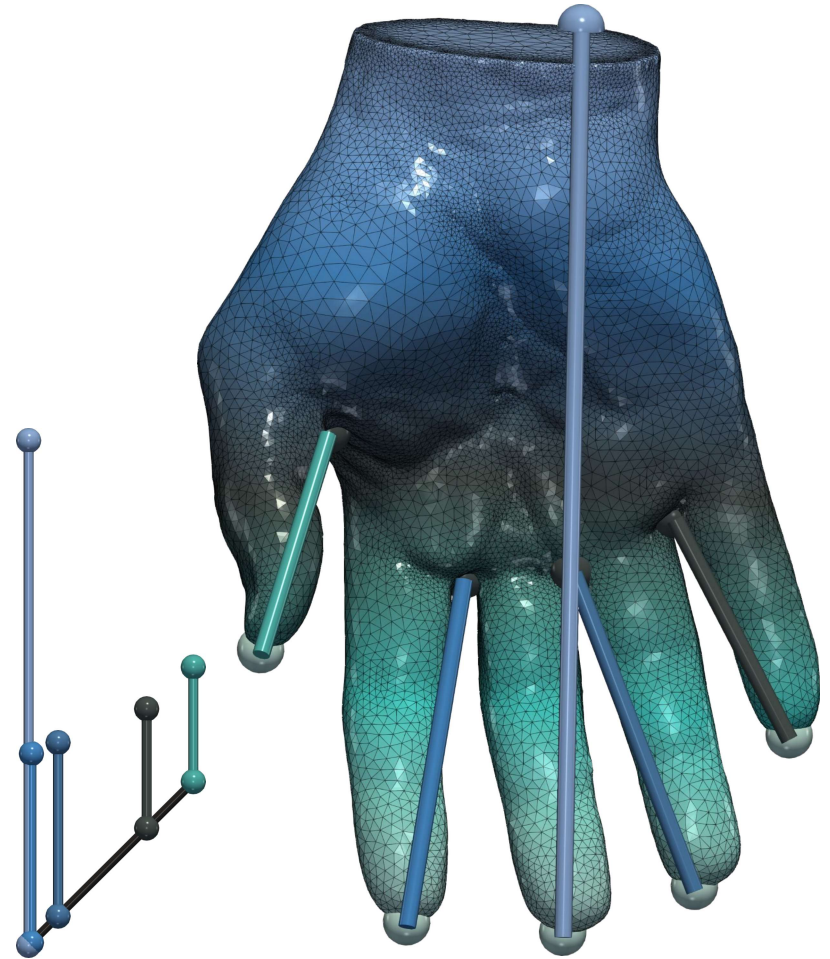
Persistence diagrams (PDs)

- **Sublevel set filtration**

- Topology changes at critical points
- Elder rule [Edelsbrunner&Harer '10]

- **Persistence diagram**

- $\mathcal{D}(f_i)$
- Bar code in birth/death space



Merge trees

- **1D simplicial complex**

- $\mathcal{I}^-(f_i) = \mathcal{M} / \sim$
- $p_1 \sim p_2$



Merge trees

- **1D simplicial complex**

- $\mathcal{T}^-(f_i) = \mathcal{M} / \sim$

- $p_1 \sim p_2$

- $f_i(p_1) = f_i(p_2)$

- Same connected component of sub-level set



Merge trees

- **1D simplicial complex**

- $\mathcal{T}^-(f_i) = \mathcal{M} / \sim$

- $p_1 \sim p_2$

- $f_i(p_1) = f_i(p_2)$

- Same connected component of sub-level set



Merge trees

- **1D simplicial complex**

- $\mathcal{T}^-(f_i) = \mathcal{M} / \sim$

- $p_1 \sim p_2$

- $f_i(p_1) = f_i(p_2)$

- Same connected component of sub-level set



Merge trees

- **1D simplicial complex**

- $\mathcal{T}^-(f_i) = \mathcal{M} / \sim$

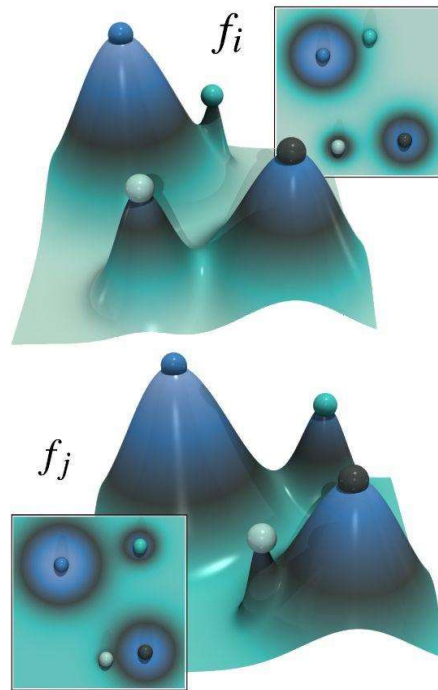
- $p_1 \sim p_2$

- $f_i(p_1) = f_i(p_2)$

- Same connected component of sub-level set

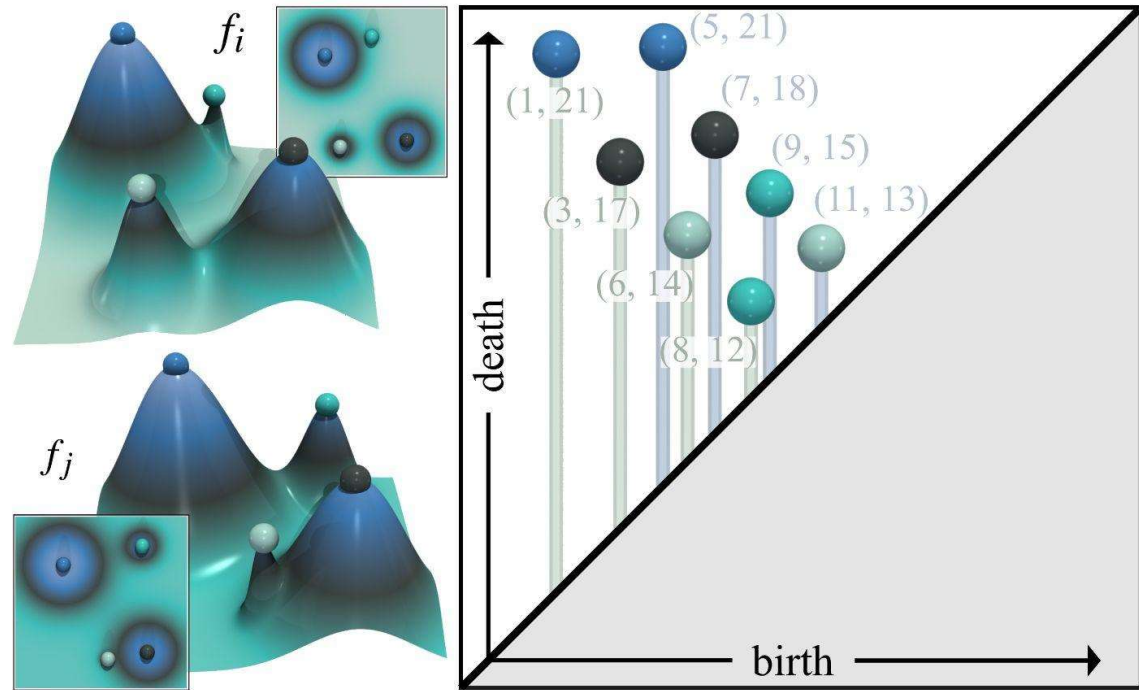


Wasserstein distance



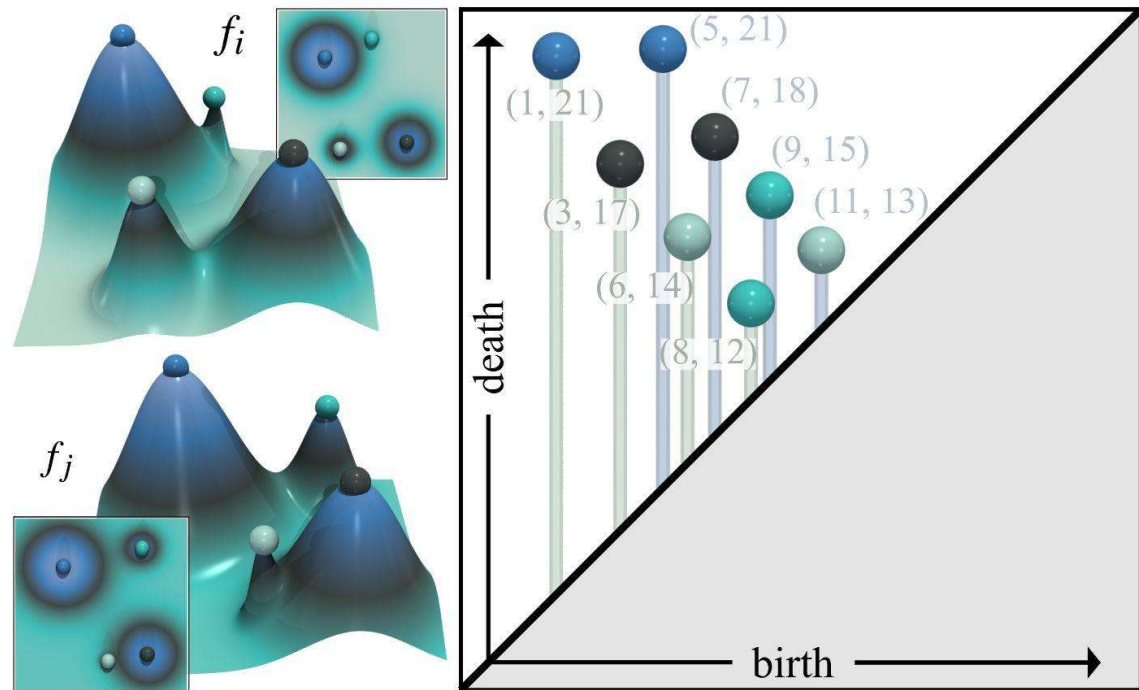
- Persistence diagrams

Wasserstein distance



- Persistence diagrams

Wasserstein distance

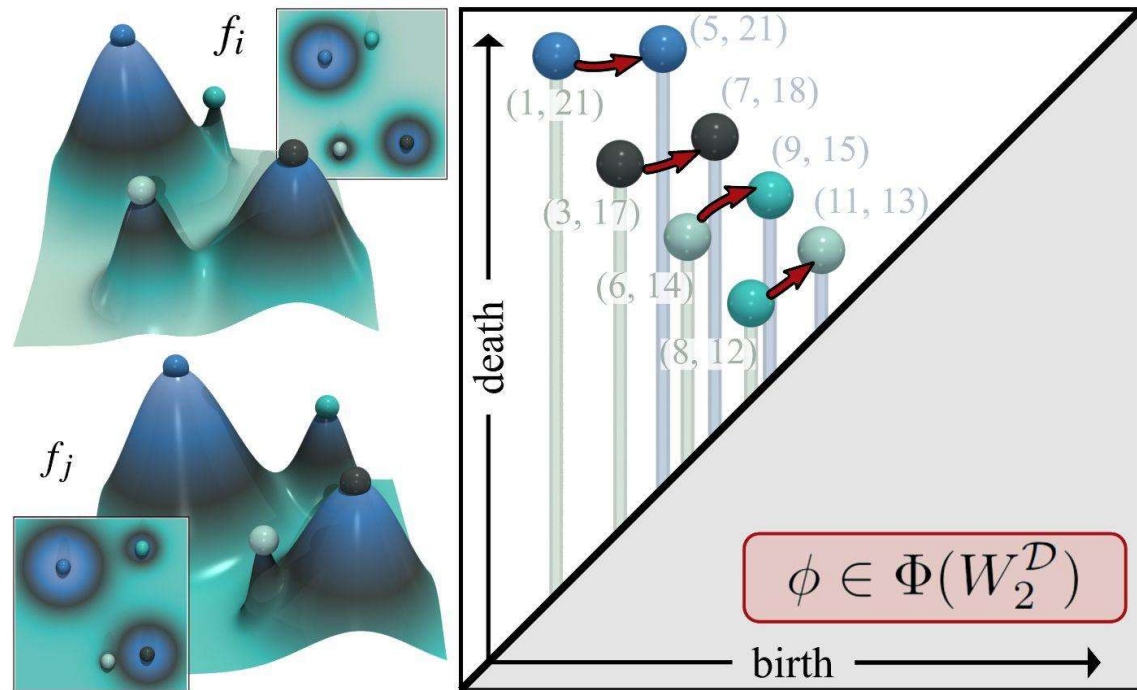


- **Persistence diagrams**

- Bipartite assignment

$$W_2^{\mathcal{D}}(\mathcal{D}(f_i), \mathcal{D}(f_j)) = \min_{\phi \in \Phi} \left(\sum_{p_i \in \mathcal{D}(f_i)} d_2(p_i, \phi(p_i))^2 \right)^{1/2}$$

Wasserstein distance

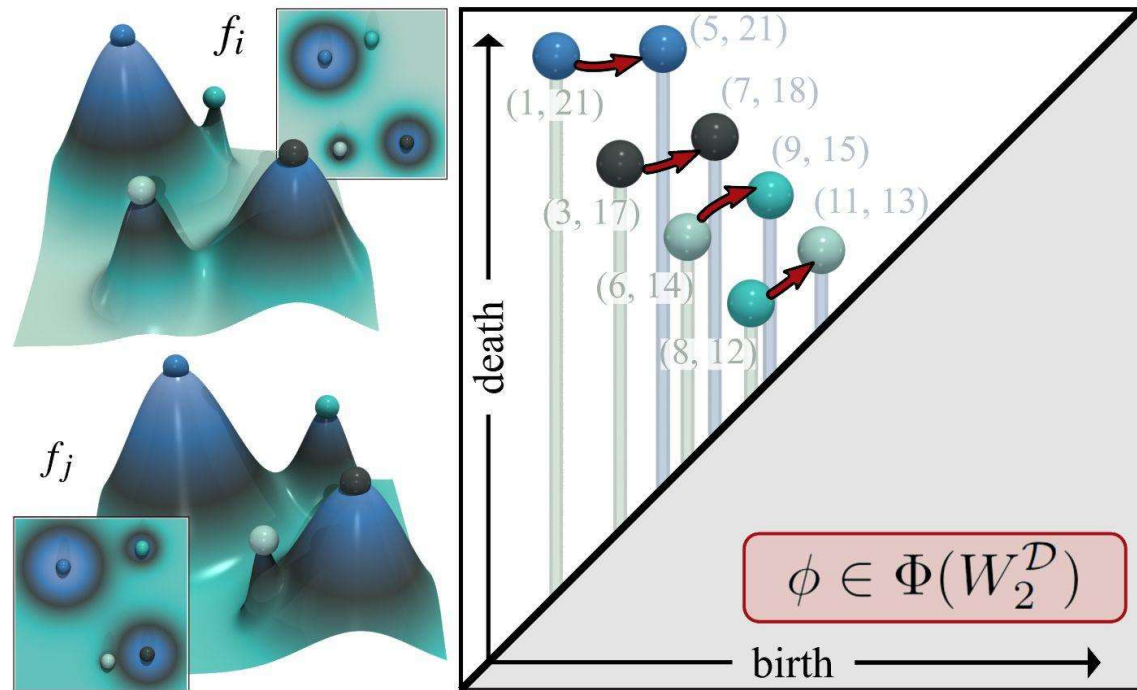


- **Persistence diagrams**

- Bipartite assignment

$$W_2^{\mathcal{D}}(\mathcal{D}(f_i), \mathcal{D}(f_j)) = \min_{\phi \in \Phi} \left(\sum_{p_i \in \mathcal{D}(f_i)} d_2(p_i, \phi(p_i))^2 \right)^{1/2}$$

Wasserstein distance

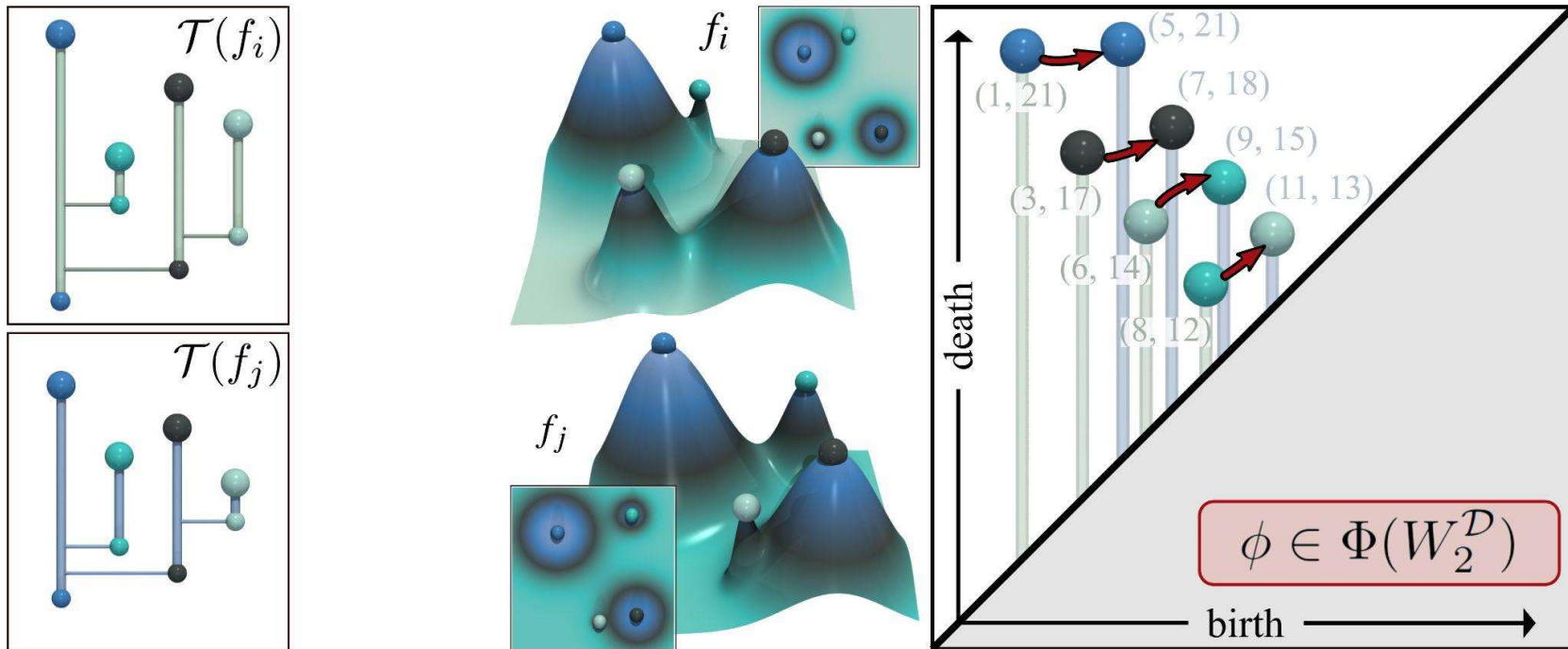


- **Persistence diagrams**

- Bipartite assignment
- Features treated independently

$$W_2^{\mathcal{D}}(\mathcal{D}(f_i), \mathcal{D}(f_j)) = \min_{\phi \in \Phi} \left(\sum_{p_i \in \mathcal{D}(f_i)} d_2(p_i, \phi(p_i))^2 \right)^{1/2}$$

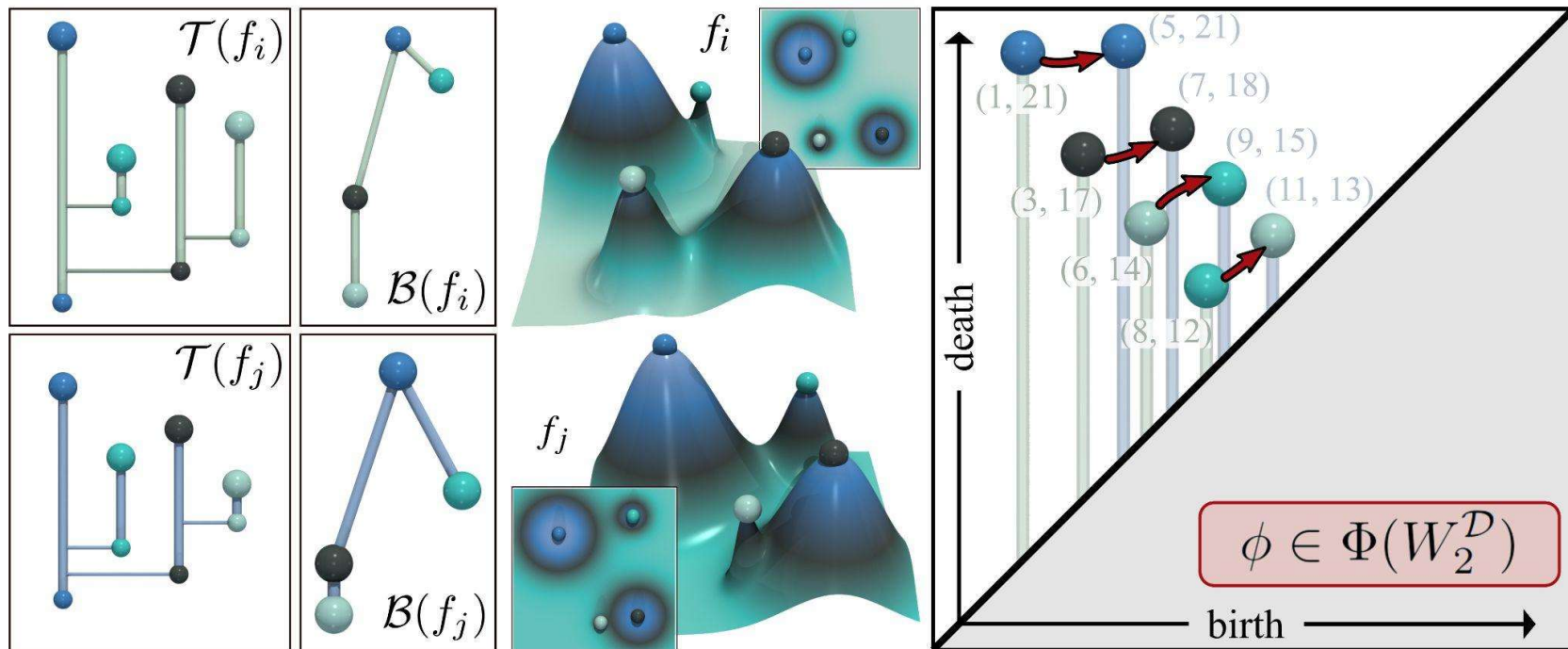
Wasserstein distance



- Merge trees [Pont '21]

$$W_2^{\mathcal{D}}(\mathcal{D}(f_i), \mathcal{D}(f_j)) = \min_{\phi \in \Phi} \left(\sum_{p_i \in \mathcal{D}(f_i)} d_2(p_i, \phi(p_i))^2 \right)^{1/2}$$

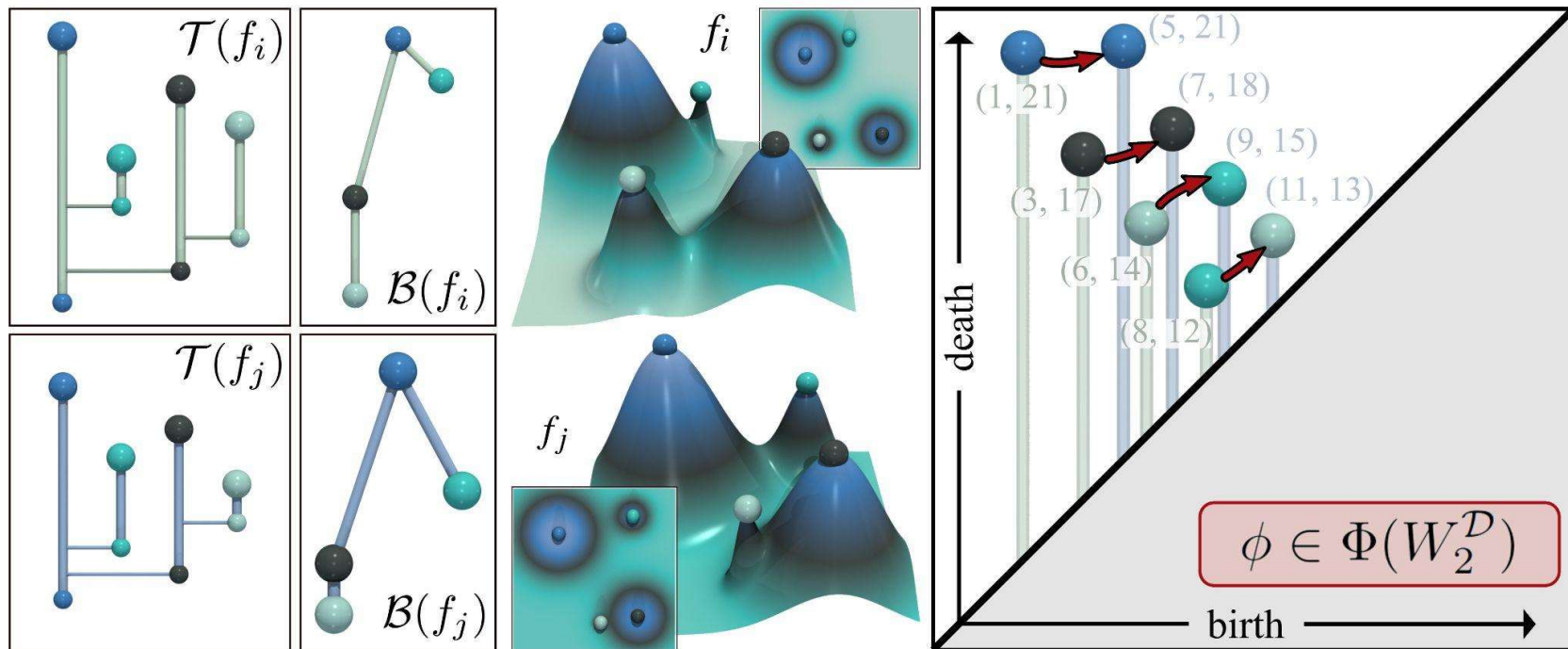
Wasserstein distance



- Merge trees [Pont '21]

$$W_2^{\mathcal{D}}(\mathcal{D}(f_i), \mathcal{D}(f_j)) = \min_{\phi \in \Phi} \left(\sum_{p_i \in \mathcal{D}(f_i)} d_2(p_i, \phi(p_i))^2 \right)^{1/2}$$

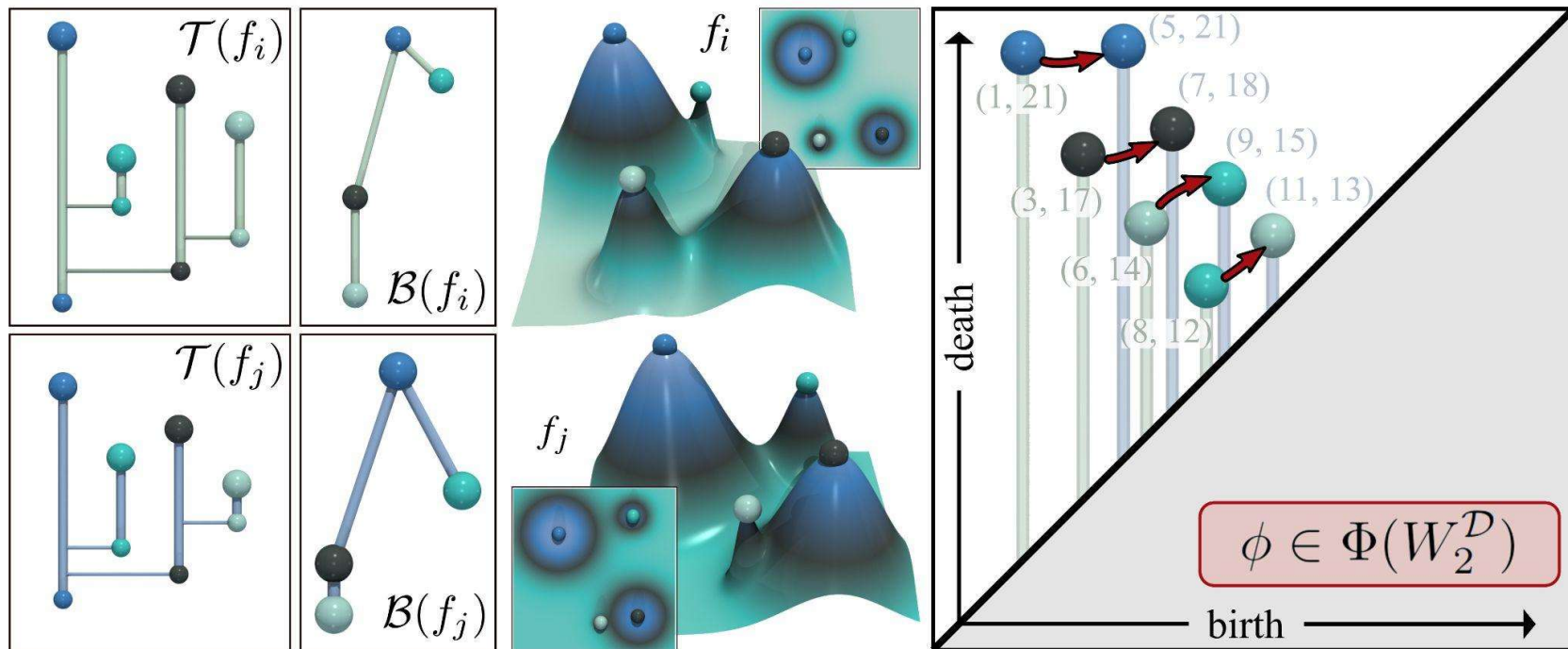
Wasserstein distance



- **Merge trees** [Pont '21]
 - Partial (rooted) isomorphisms

$$W_2^D(\mathcal{D}(f_i), \mathcal{D}(f_j)) = \min_{\phi \in \Phi} \left(\sum_{p_i \in \mathcal{D}(f_i)} d_2(p_i, \phi(p_i))^2 \right)^{1/2}$$

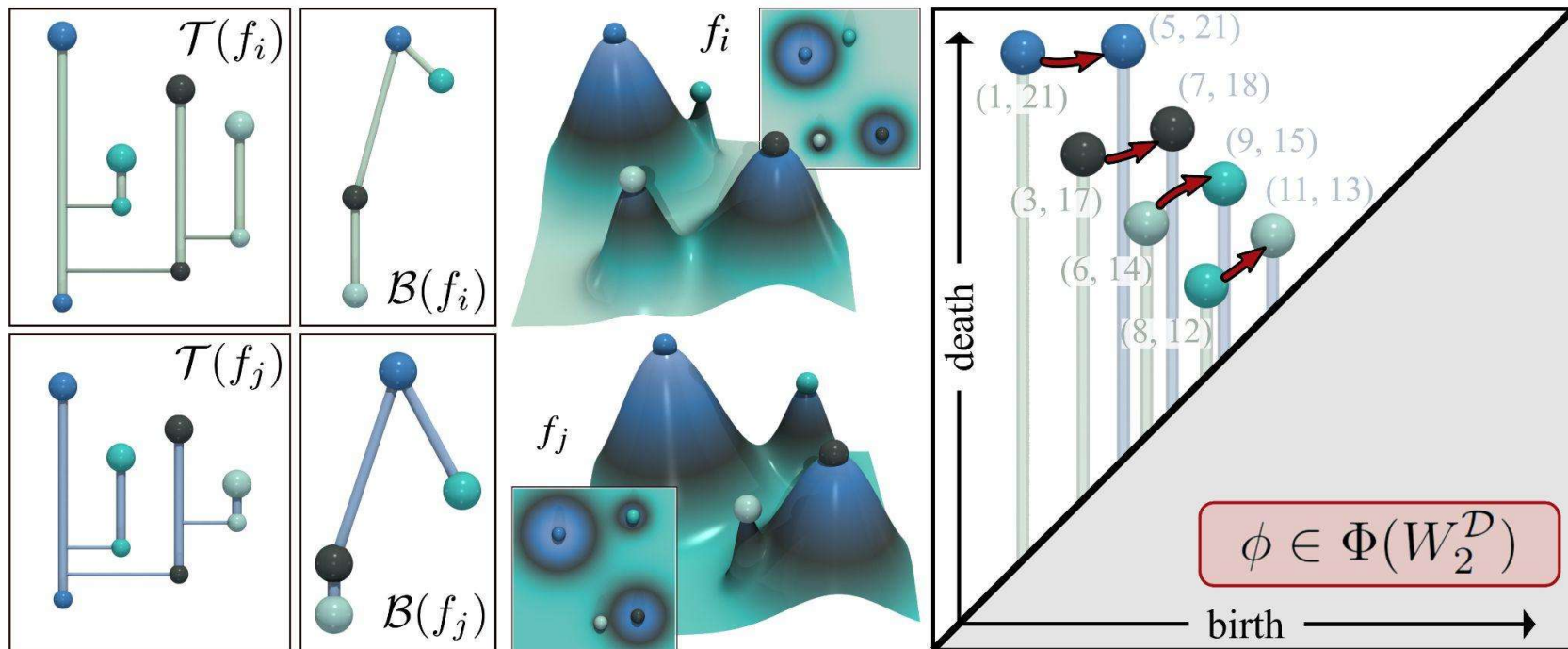
Wasserstein distance



- **Merge trees** [Pont '21]
 - Partial (rooted) isomorphisms

$$W_2^{\mathcal{T}}(\mathcal{B}(f_i), \mathcal{B}(f_j)) = \min_{\phi' \in \Phi'} \left(\sum_{p_i \in \mathcal{B}(f_i)} d_2(p_i, \phi'(p_i))^2 \right)^{1/2}$$

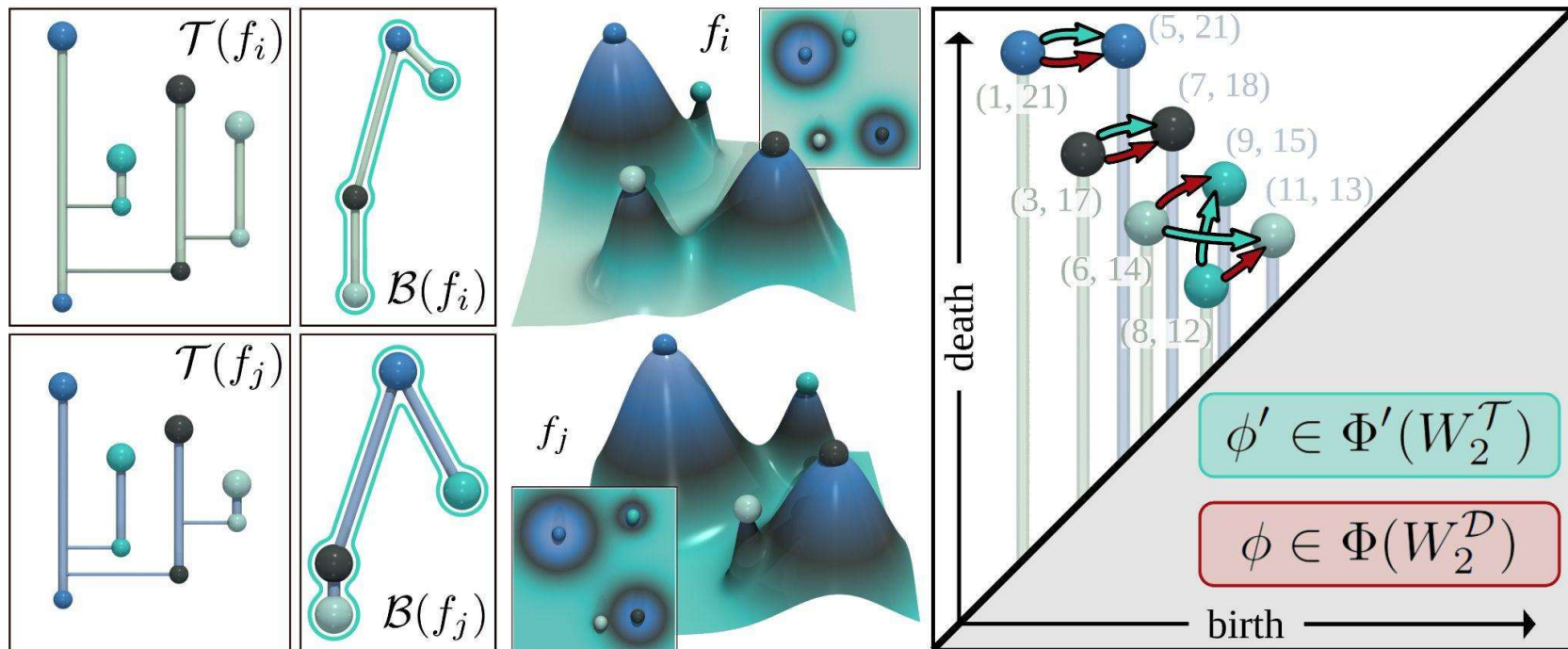
Wasserstein distance



- **Merge trees** [Pont '21]
 - Partial (rooted) isomorphisms

$$W_2^{\mathcal{T}}(\mathcal{B}(f_i), \mathcal{B}(f_j)) = \min_{\phi' \in \Phi'} \left(\sum_{p_i \in \mathcal{B}(f_i)} d_2(p_i, \phi'(p_i))^2 \right)^{1/2}$$

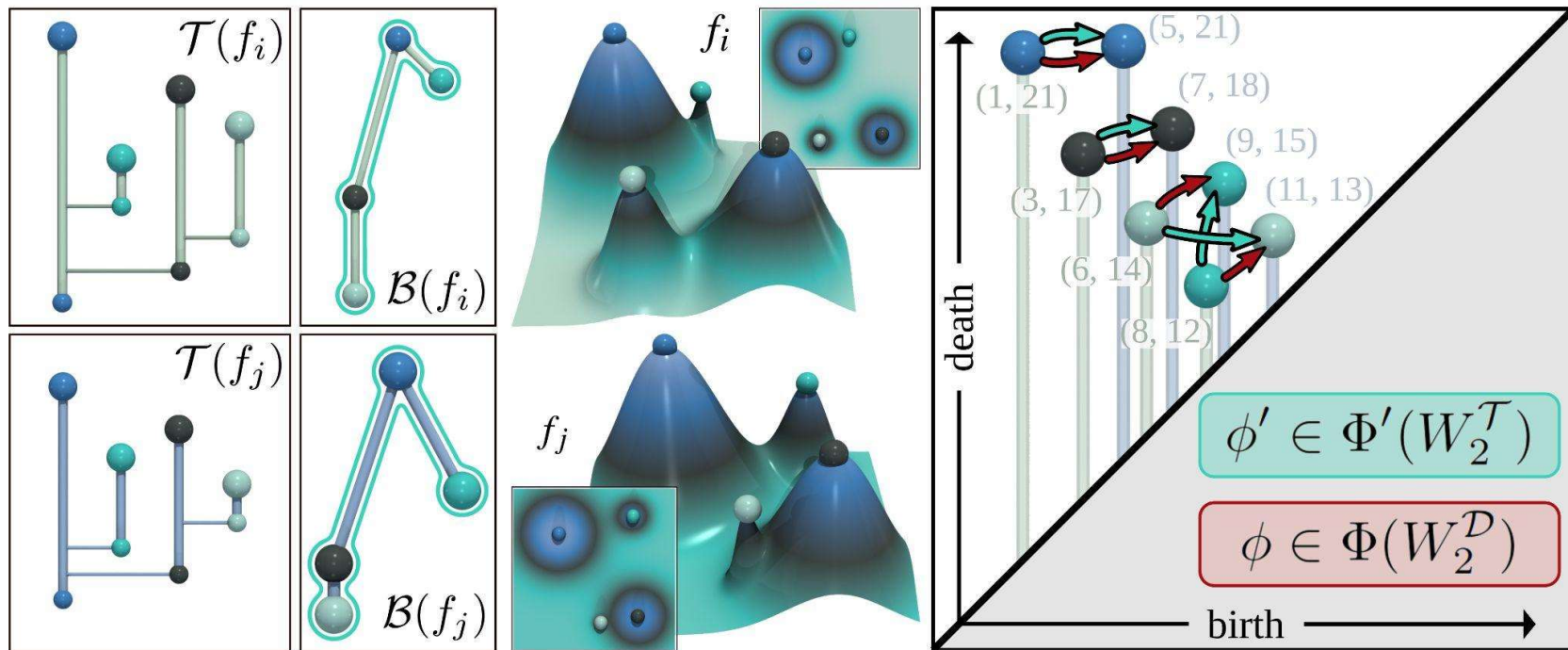
Wasserstein distance



- **Merge trees** [Pont '21]
 - Partial (rooted) isomorphisms

$$W_2^{\mathcal{T}}(\mathcal{B}(f_i), \mathcal{B}(f_j)) = \min_{\phi' \in \Phi'} \left(\sum_{p_i \in \mathcal{B}(f_i)} d_2(p_i, \phi'(p_i))^2 \right)^{1/2}$$

Wasserstein distance

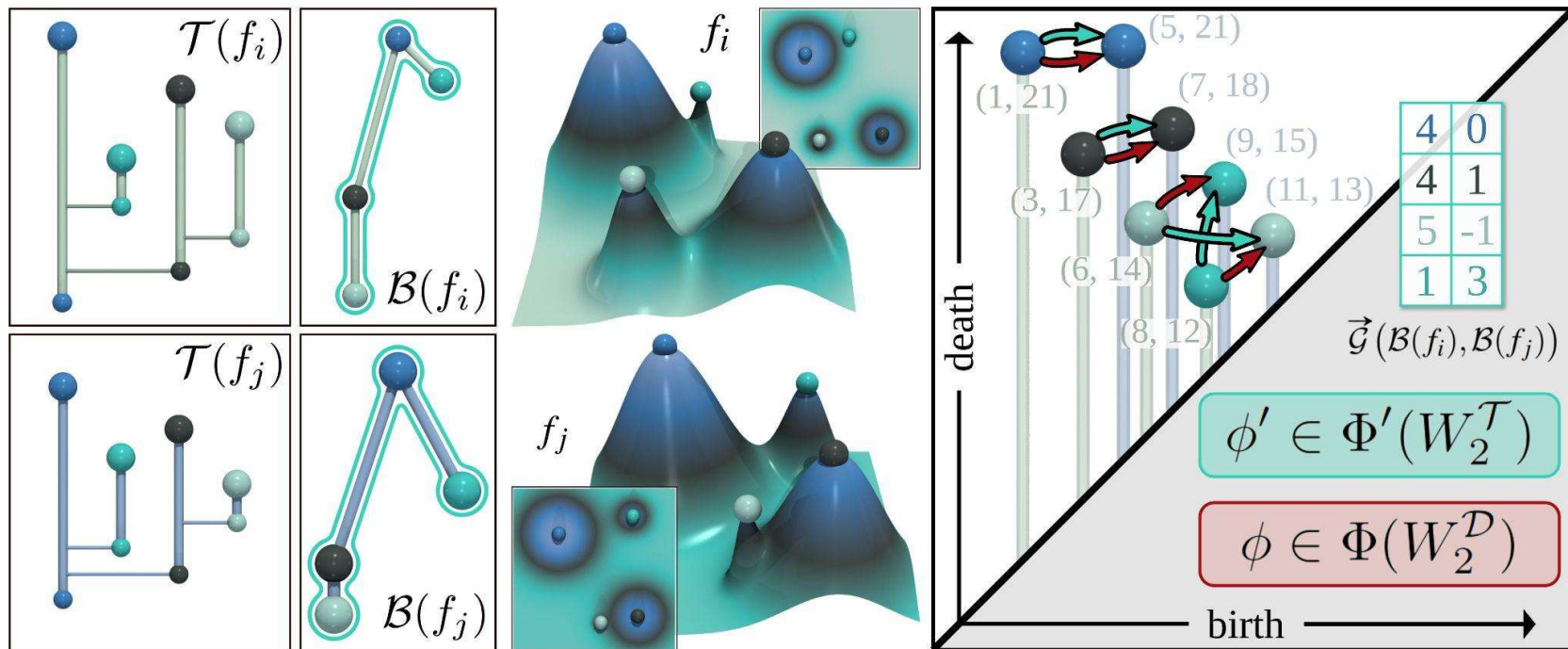


- **Merge trees** [Pont '21]

- Partial (rooted) isomorphisms
- Capture global structure

$$W_2^{\mathcal{T}}(\mathcal{B}(f_i), \mathcal{B}(f_j)) = \min_{\phi' \in \Phi'} \left(\sum_{p_i \in \mathcal{B}(f_i)} d_2(p_i, \phi'(p_i))^2 \right)^{1/2}$$

Wasserstein distance

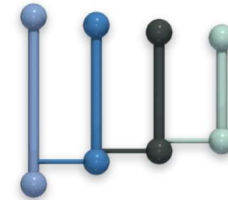
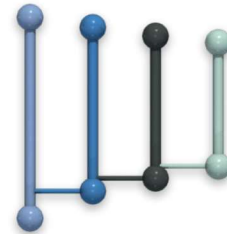
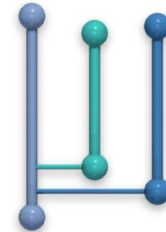
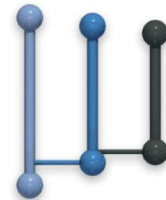
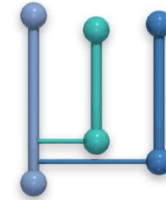
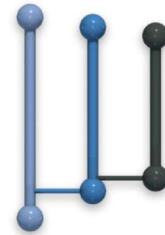


- **Merge trees** [Pont '21]

- Partial (rooted) isomorphisms
- Capture global structure
- **Geodesic**: vector in $\mathbb{R}^{2 \times |\mathcal{B}(f_i)|}$

$$W_2^{\mathcal{T}}(\mathcal{B}(f_i), \mathcal{B}(f_j)) = \min_{\phi' \in \Phi'} \left(\sum_{p_i \in \mathcal{B}(f_i)} d_2(p_i, \phi'(p_i))^2 \right)^{1/2}$$

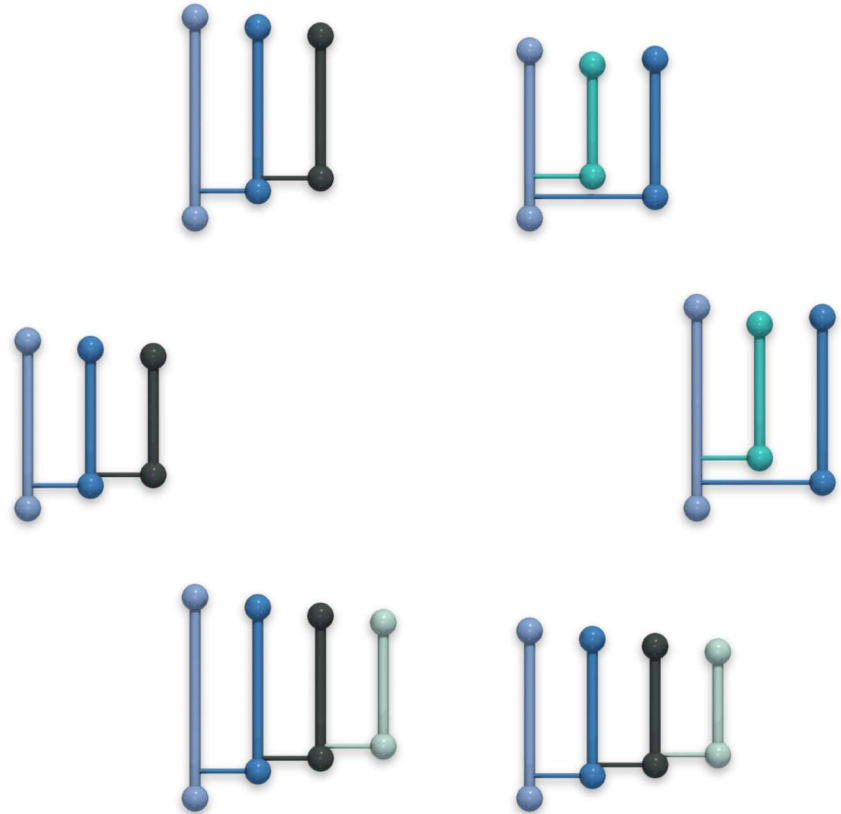
Wasserstein barycenters



Wasserstein barycenters

- Fréchet energy

$$E_F(\mathcal{B}) = \sum_{i=1}^N W_2^T(\mathcal{B}, \mathcal{B}(f_i))^2$$



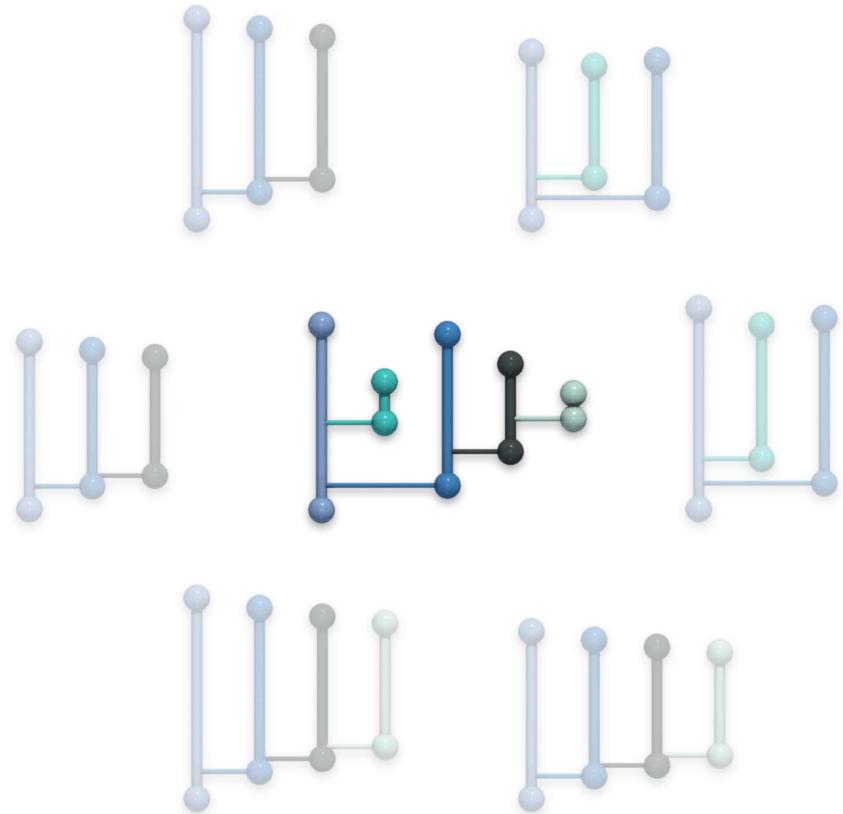
Wasserstein barycenters

- Fréchet energy

$$E_F(\mathcal{B}) = \sum_{i=1}^N W_2^T(\mathcal{B}, \mathcal{B}(f_i))^2$$

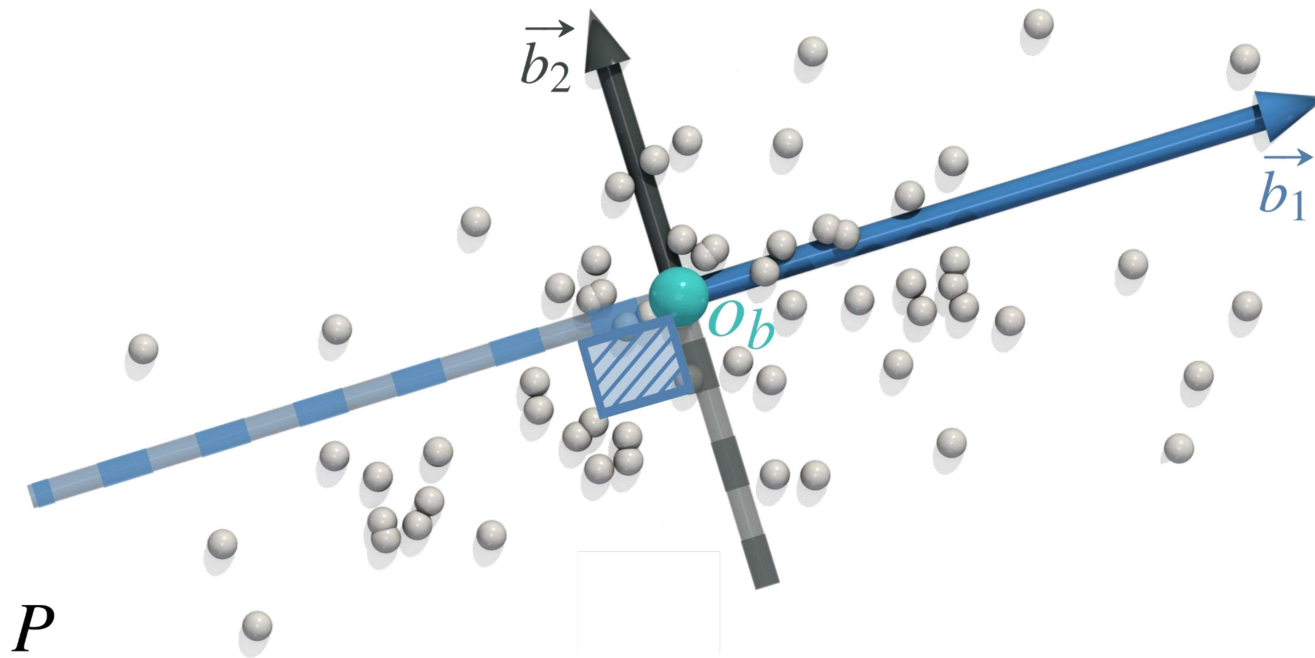
- **Algorithm** [Pont' 21]

- Iterative Assignment/Update



Approach

Geometric interpretation of PCA



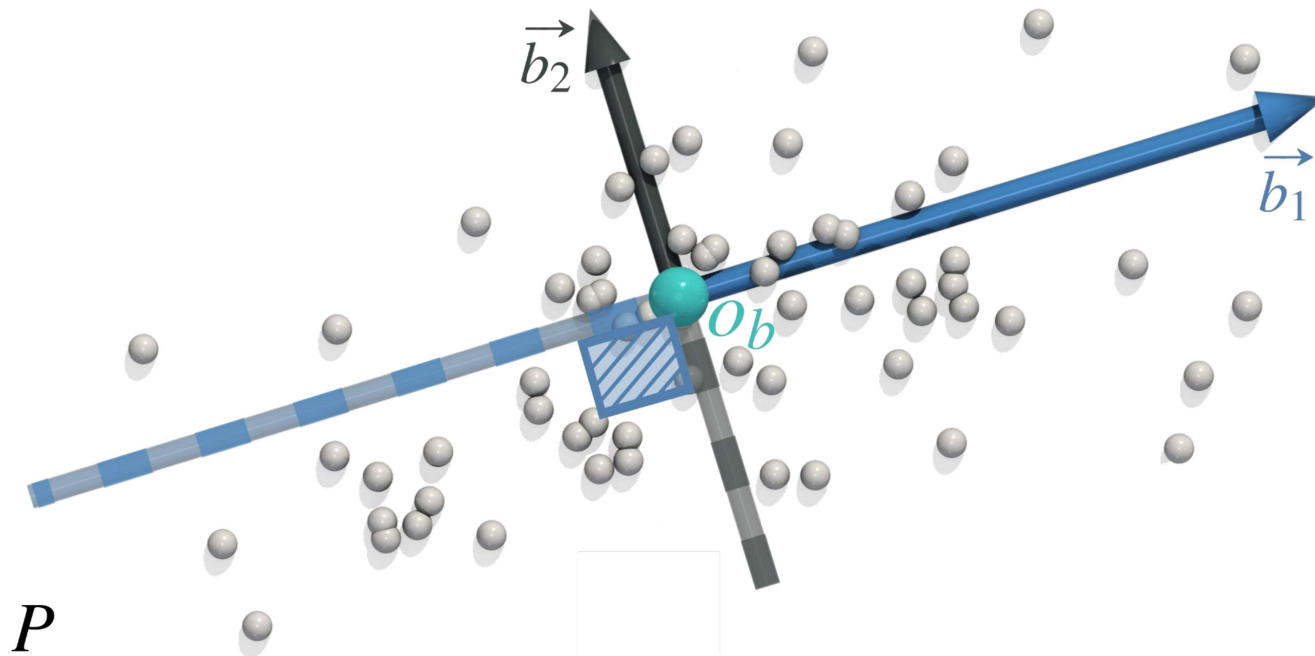
- Input

$$P = \{p_1, p_2, \dots, p_N\} \text{ in } \mathbb{R}^d$$

- PCA

$$B_{\mathbb{R}^d} = \{\vec{b}_1, \vec{b}_2, \dots, \vec{b}_d\}$$

Geometric interpretation of PCA



- Input

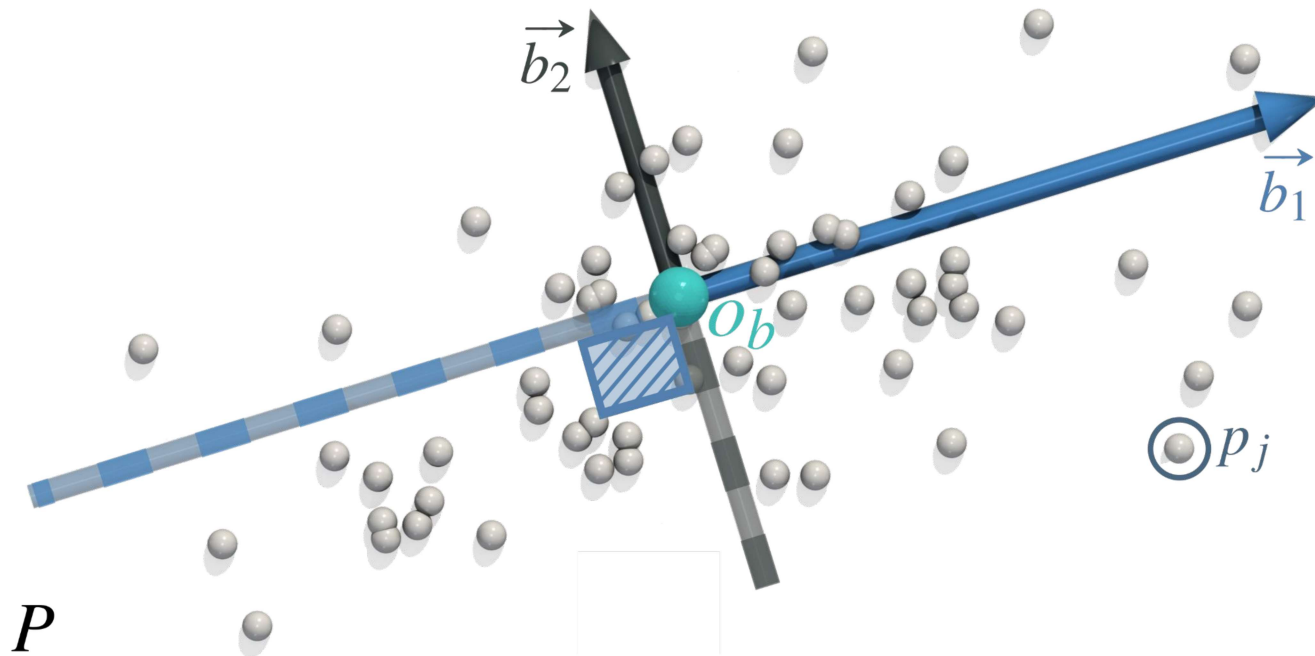
$$P = \{p_1, p_2, \dots, p_N\} \text{ in } \mathbb{R}^d$$

- PCA

$$B_{\mathbb{R}^d} = \{\vec{b}_1, \vec{b}_2, \dots, \vec{b}_d\}$$

$$E_{L_2}(B_{\mathbb{R}^d}) = \sum_{j=1}^N \|p_j - (o_b + \sum_{i=1}^{d'} \alpha_i^j \vec{b}_i)\|_2^2$$

Geometric interpretation of PCA



- Input

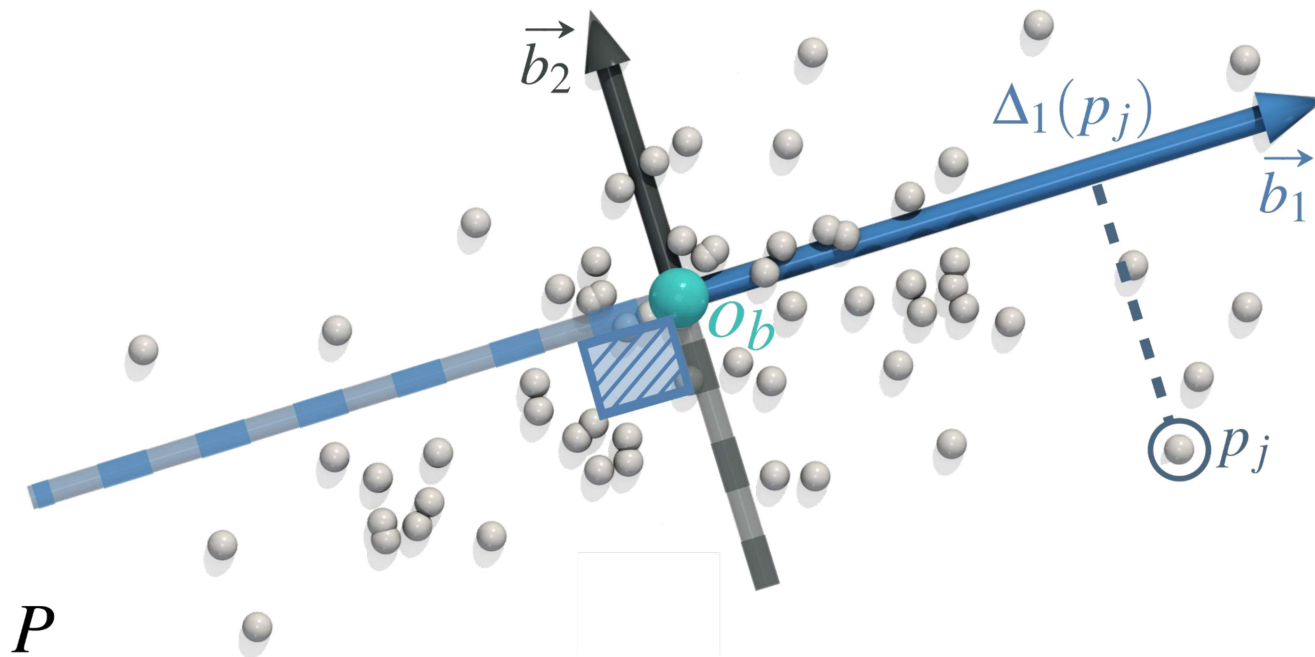
$$P = \{p_1, p_2, \dots, p_N\} \text{ in } \mathbb{R}^d$$

- PCA

$$B_{\mathbb{R}^d} = \{\vec{b}_1, \vec{b}_2, \dots, \vec{b}_d\}$$

$$E_{L_2}(B_{\mathbb{R}^d}) = \sum_{j=1}^N \|p_j - (o_b + \sum_{i=1}^{d'} \alpha_i^j \vec{b}_i)\|_2^2$$

Geometric interpretation of PCA



- Input

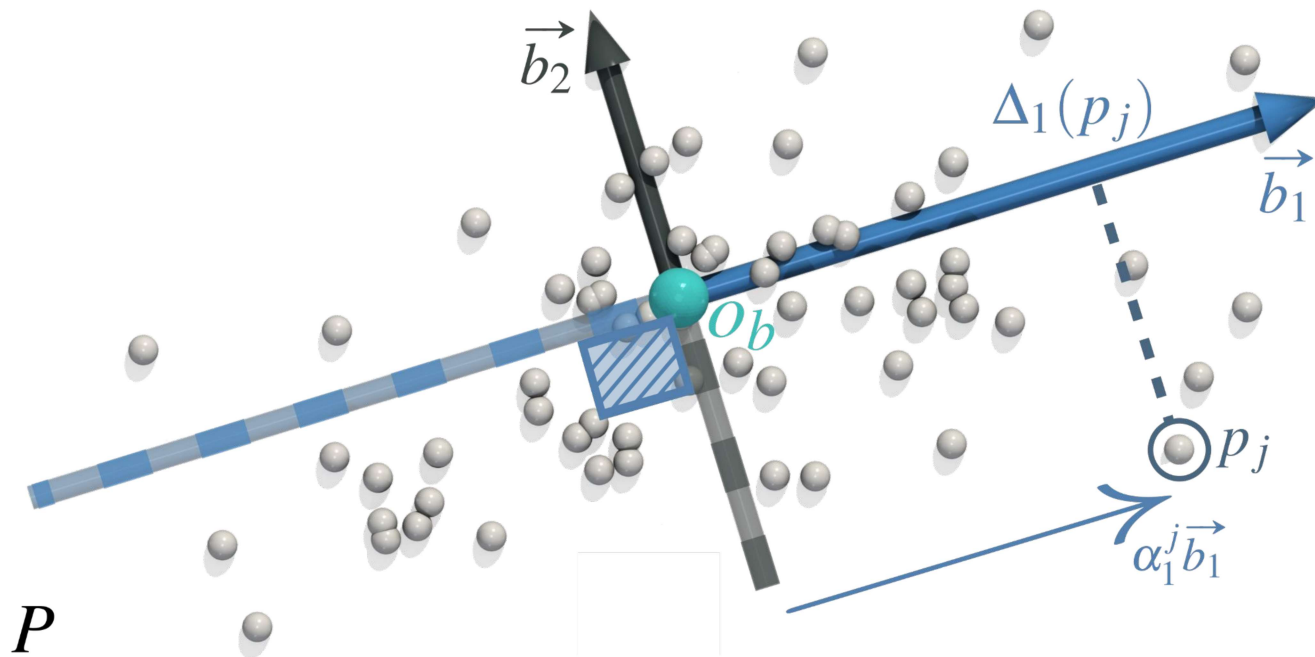
$$P = \{p_1, p_2, \dots, p_N\} \text{ in } \mathbb{R}^d$$

- PCA

$$B_{\mathbb{R}^d} = \{\vec{b}_1, \vec{b}_2, \dots, \vec{b}_d\}$$

$$E_{L_2}(B_{\mathbb{R}^d}) = \sum_{j=1}^N \|p_j - (o_b + \sum_{i=1}^{d'} \alpha_i^j \vec{b}_i)\|_2^2$$

Geometric interpretation of PCA



- Input

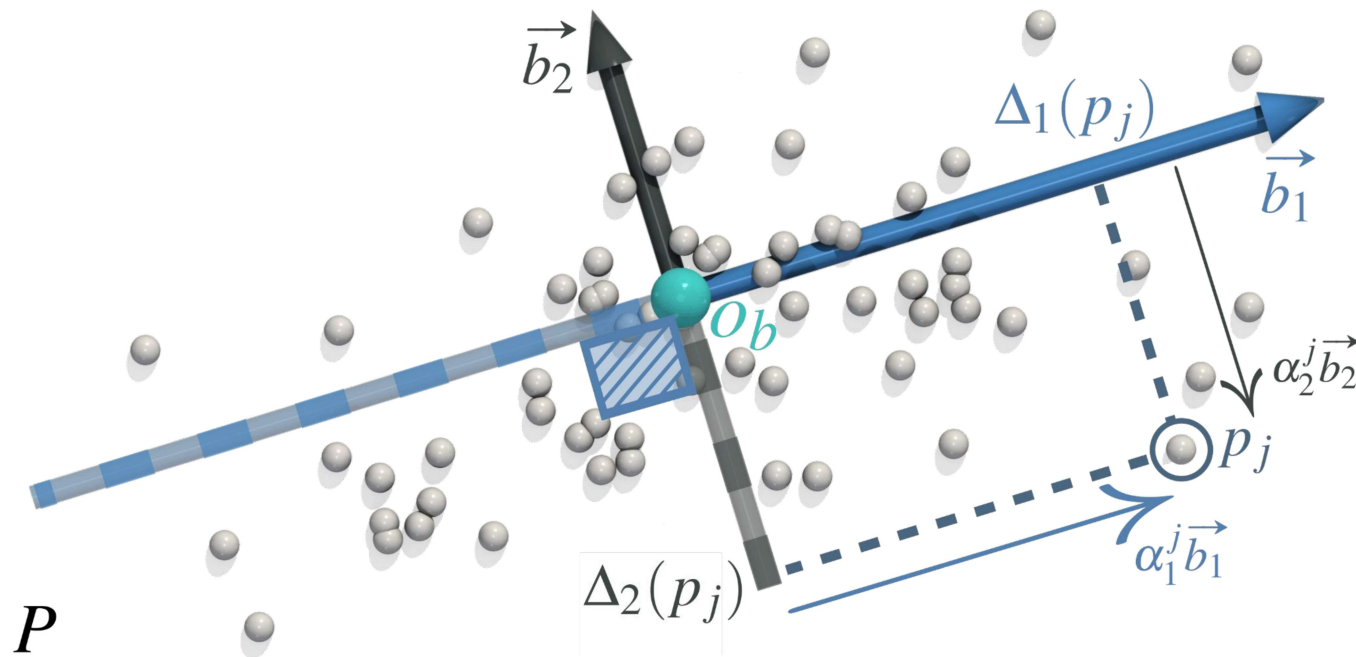
$$P = \{p_1, p_2, \dots, p_N\} \text{ in } \mathbb{R}^d$$

- PCA

$$B_{\mathbb{R}^d} = \{\vec{b}_1, \vec{b}_2, \dots, \vec{b}_d\}$$

$$E_{L_2}(B_{\mathbb{R}^d}) = \sum_{j=1}^N \|p_j - (o_b + \sum_{i=1}^{d'} \alpha_i^j \vec{b}_i)\|_2^2$$

Geometric interpretation of PCA



- Input

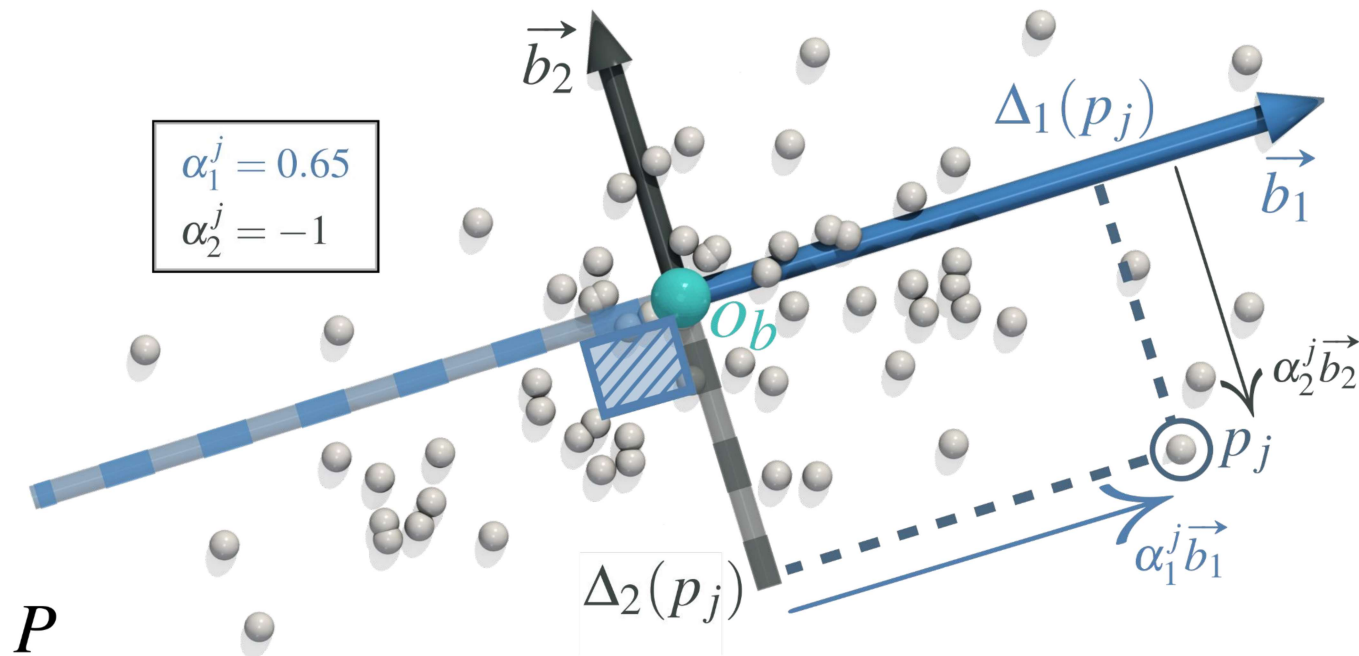
$$P = \{p_1, p_2, \dots, p_N\} \text{ in } \mathbb{R}^d$$

- PCA

$$B_{\mathbb{R}^d} = \{\vec{b}_1, \vec{b}_2, \dots, \vec{b}_d\}$$

$$E_{L_2}(B_{\mathbb{R}^d}) = \sum_{j=1}^N \|p_j - (o_b + \sum_{i=1}^{d'} \alpha_i^j \vec{b}_i)\|_2^2$$

Geometric interpretation of PCA



- Input

$$P = \{p_1, p_2, \dots, p_N\} \text{ in } \mathbb{R}^d$$

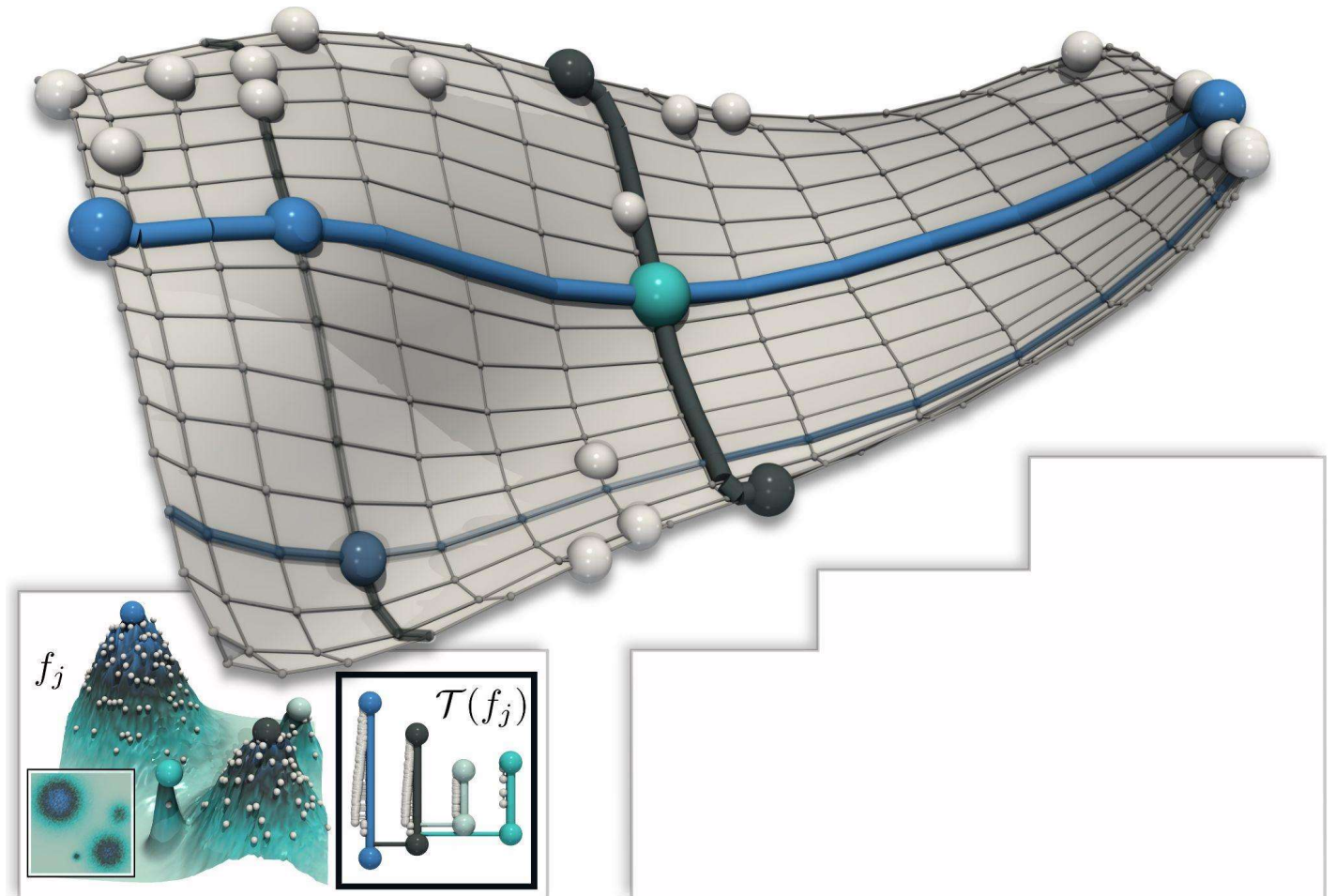
- PCA

$$B_{\mathbb{R}^d} = \{\vec{b}_1, \vec{b}_2, \dots, \vec{b}_d\}$$

$$E_{L_2}(B_{\mathbb{R}^d}) = \sum_{j=1}^N \|p_j - (o_b + \sum_{i=1}^{d'} \alpha_i^j \vec{b}_i)\|_2^2$$

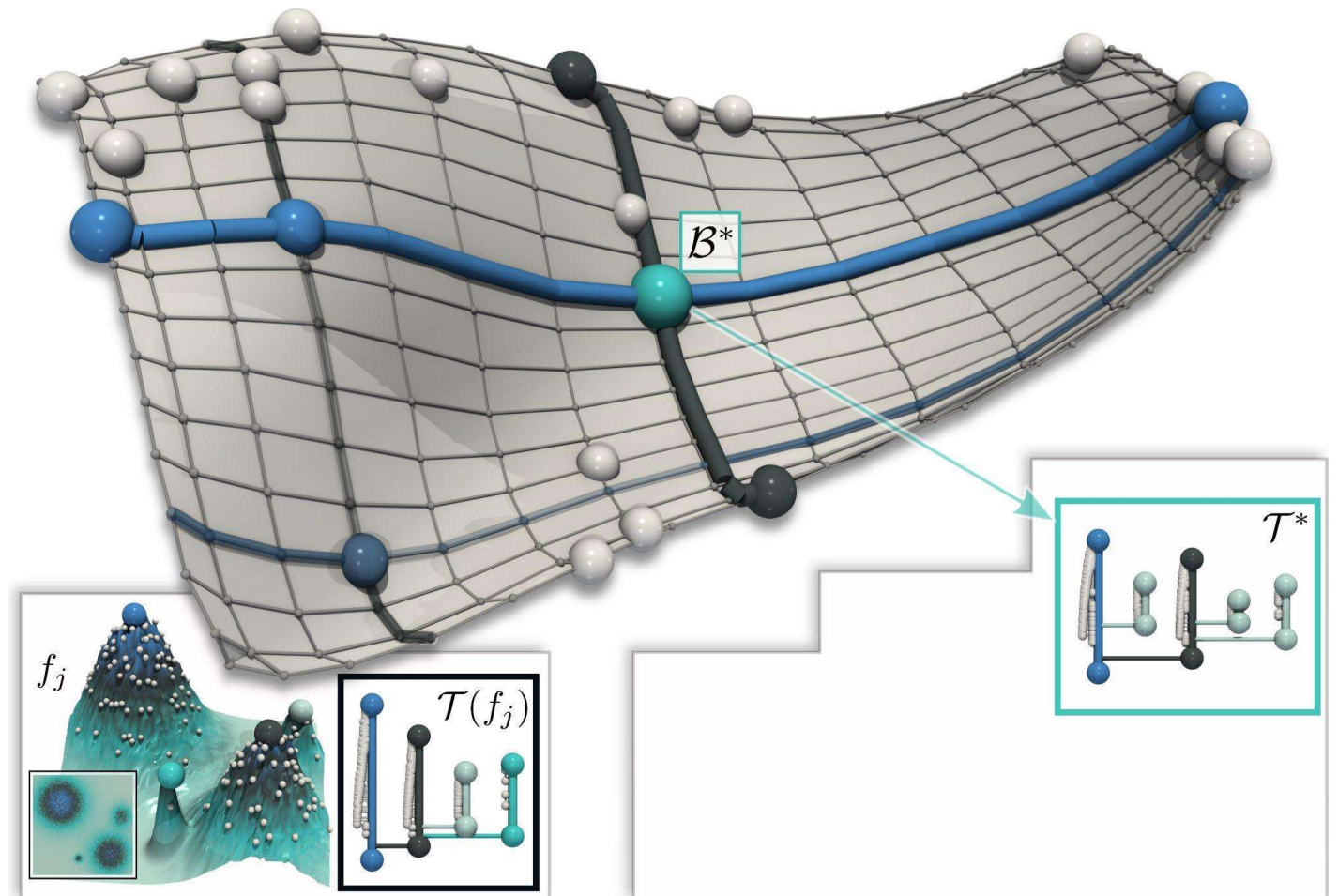
Overview

- BDT Basis



Overview

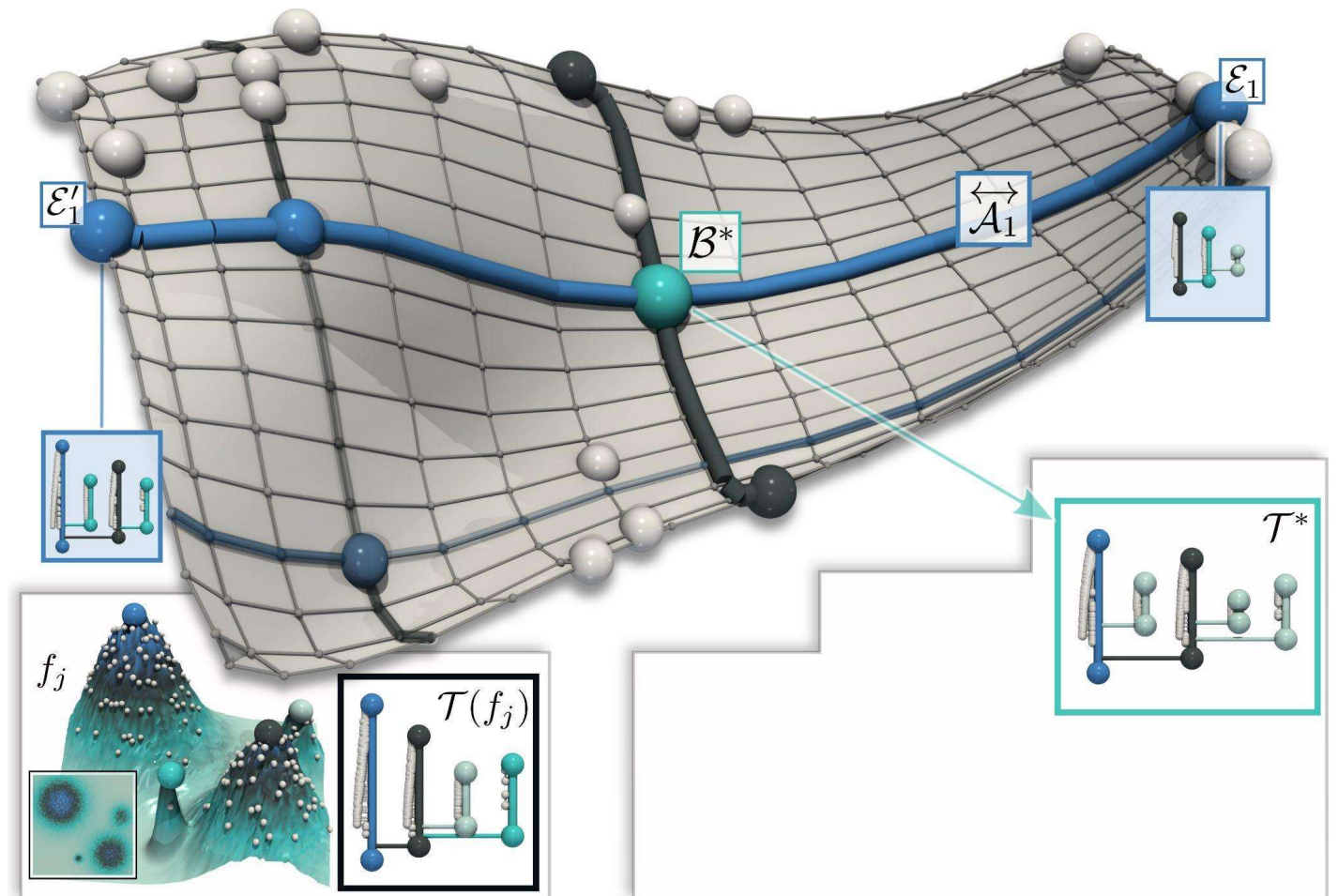
- BDT Basis
 - Barycenter



Overview

- **BDT Basis**

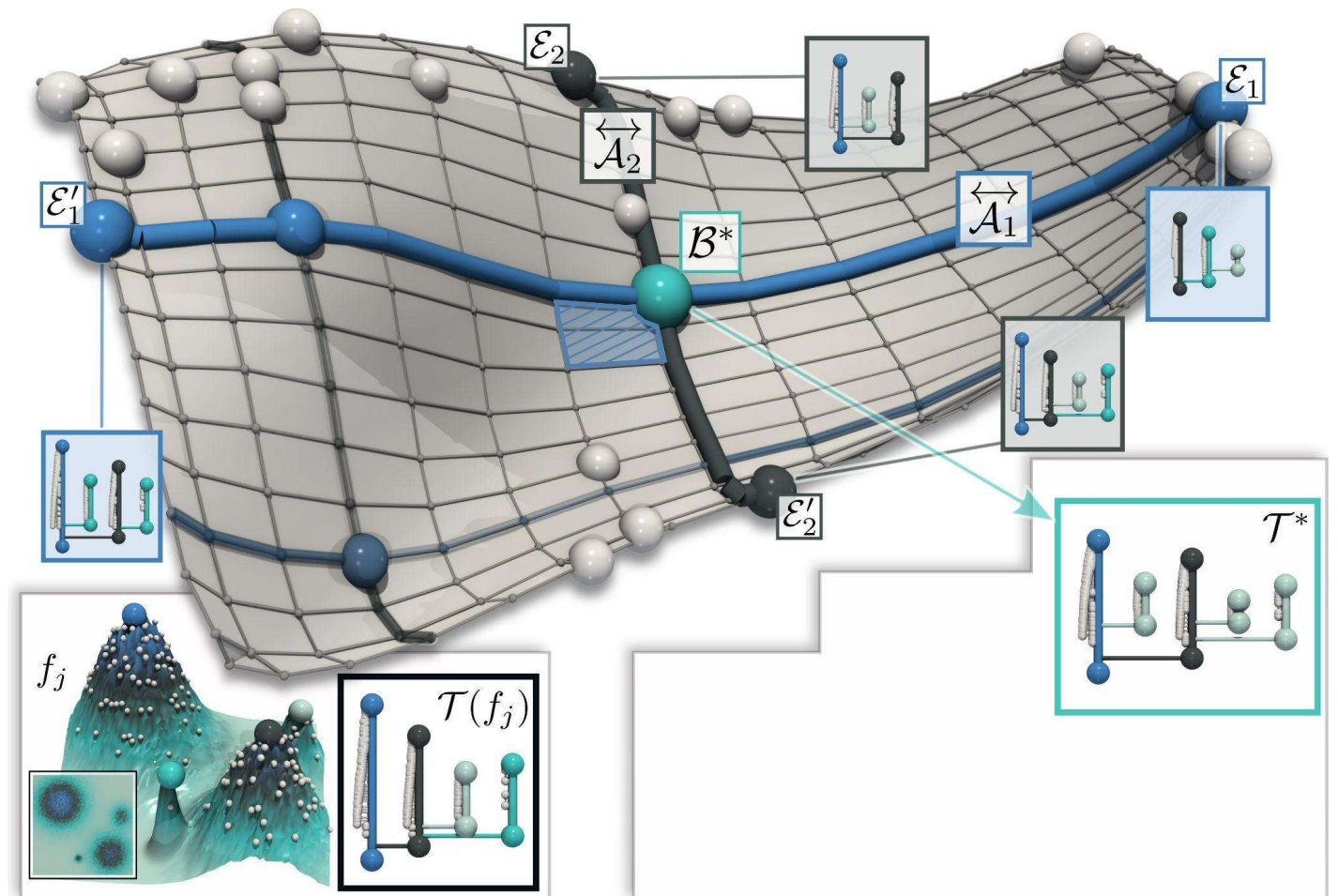
- Barycenter
- Geodesic axis



Overview

- **BDT Basis**

- Barycenter
- Geodesic axis
- Orthogonal axes



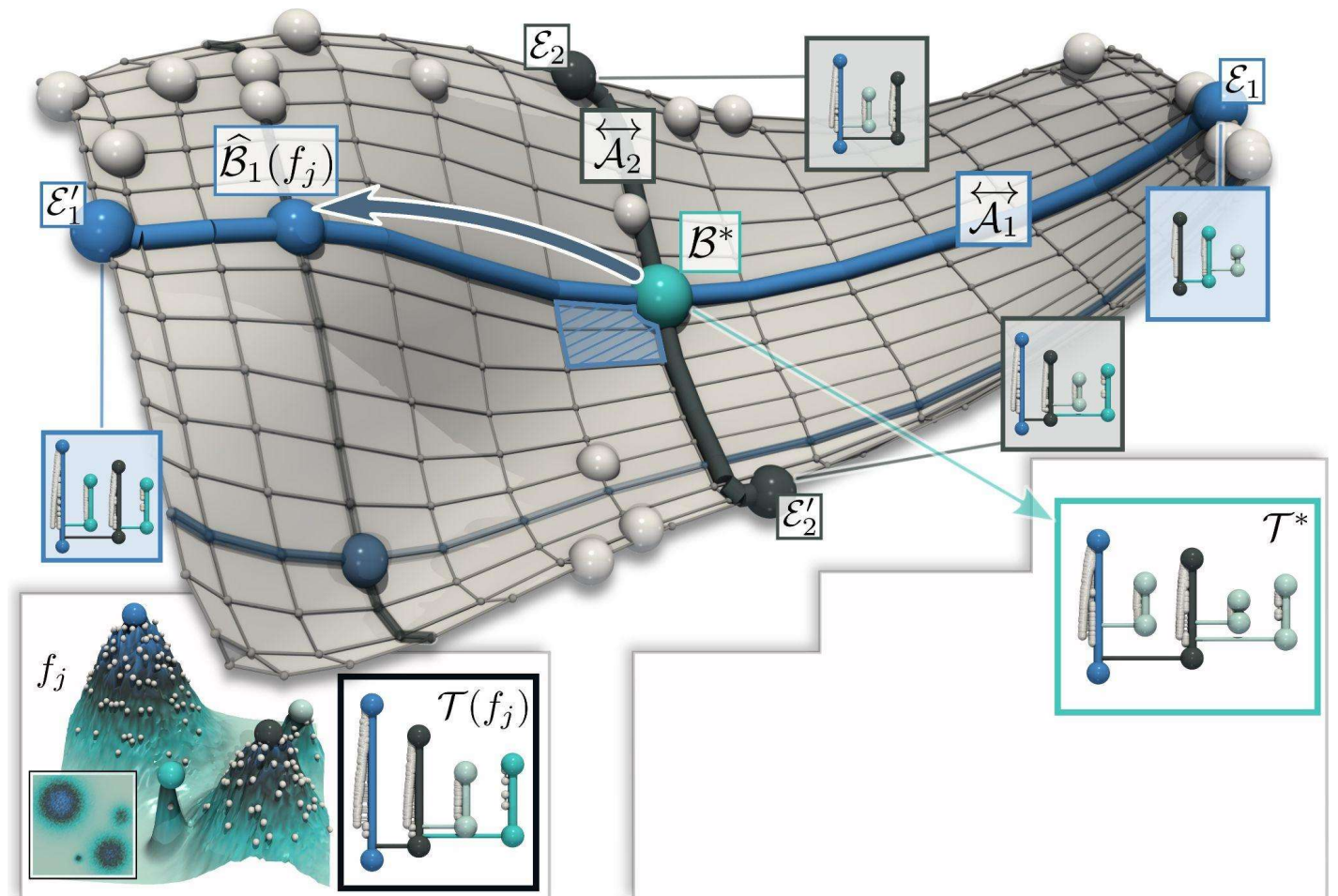
Overview

- **BDT Basis**

- Barycenter
- Geodesic axis
- Orthogonal axes

- **Estimation**

- Axis projection



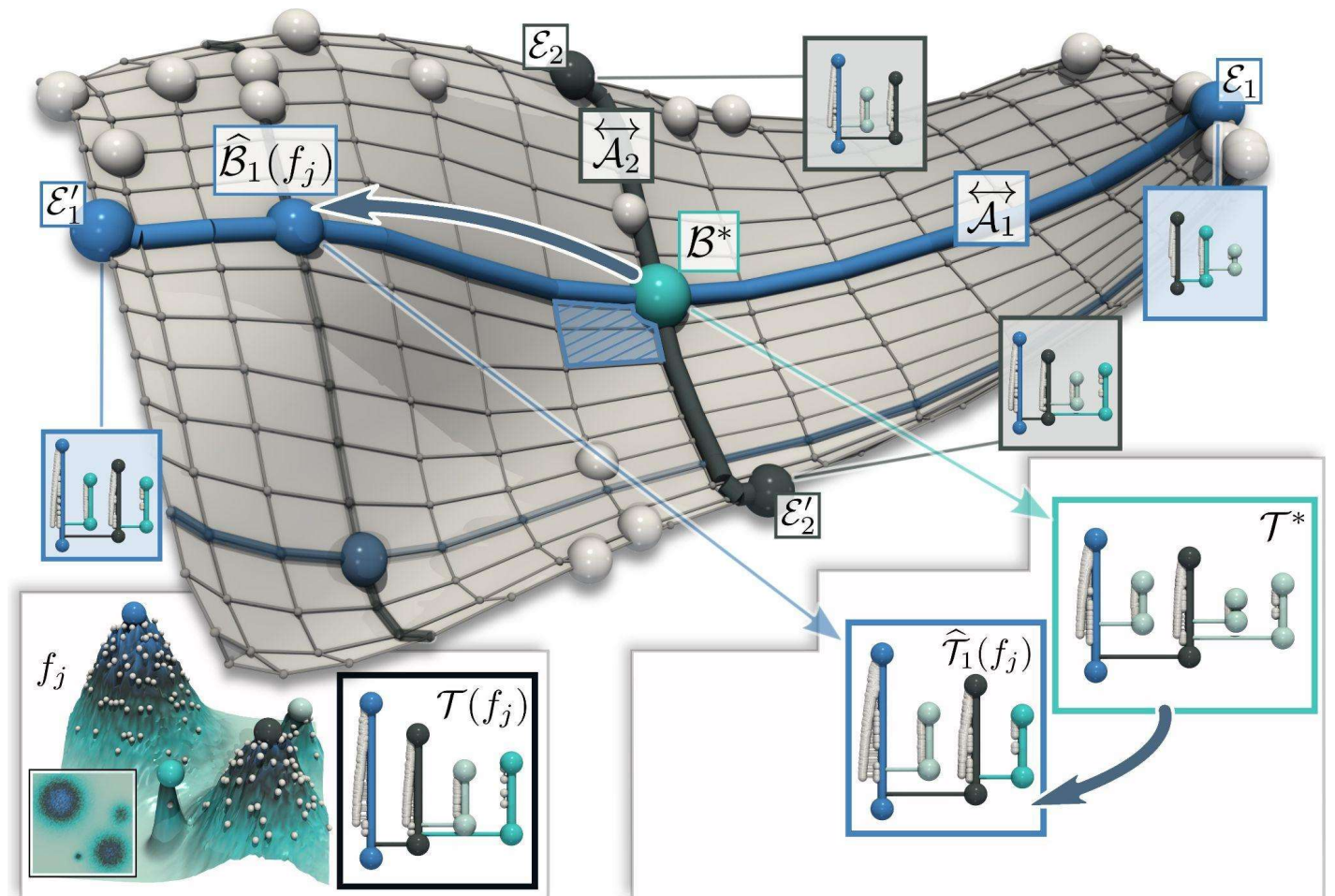
Overview

- **BDT Basis**

- Barycenter
- Geodesic axis
- Orthogonal axes

- **Estimation**

- Axis projection



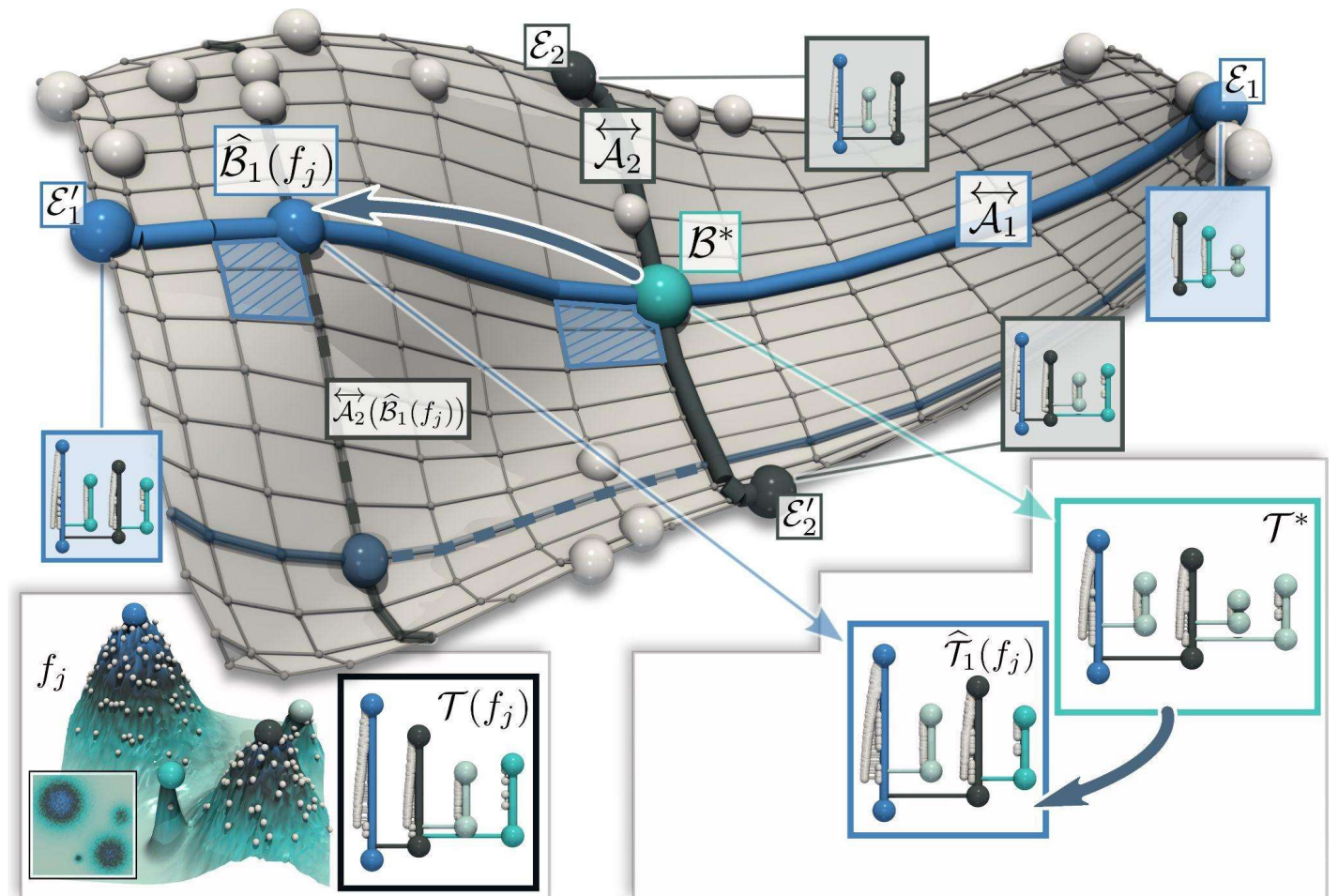
Overview

- **BDT Basis**

- Barycenter
- Geodesic axis
- Orthogonal axes

- **Estimation**

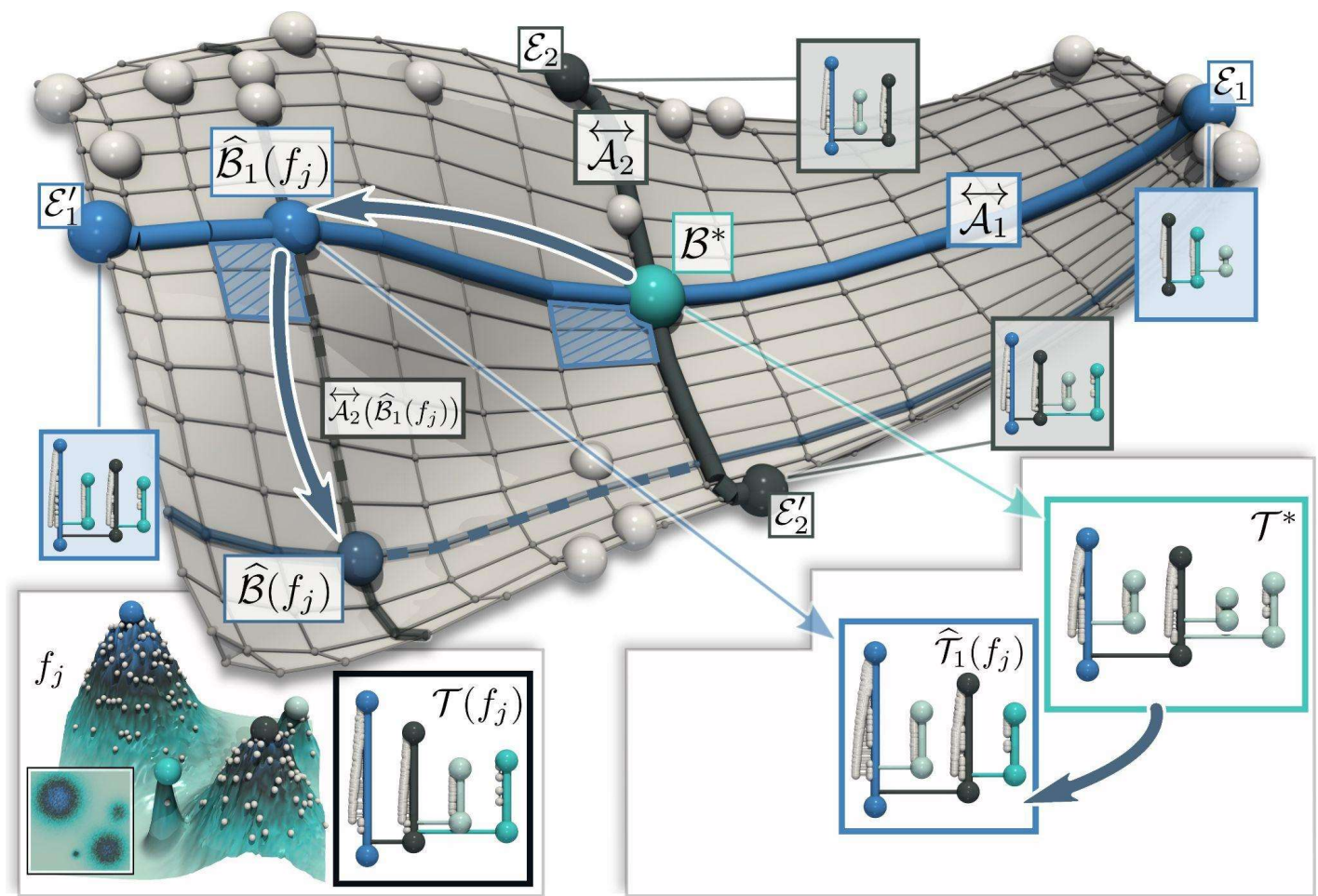
- Axis projection
- Translated axis



Overview

- **BDT Basis**
 - Barycenter
 - Geodesic axis
 - Orthogonal axes

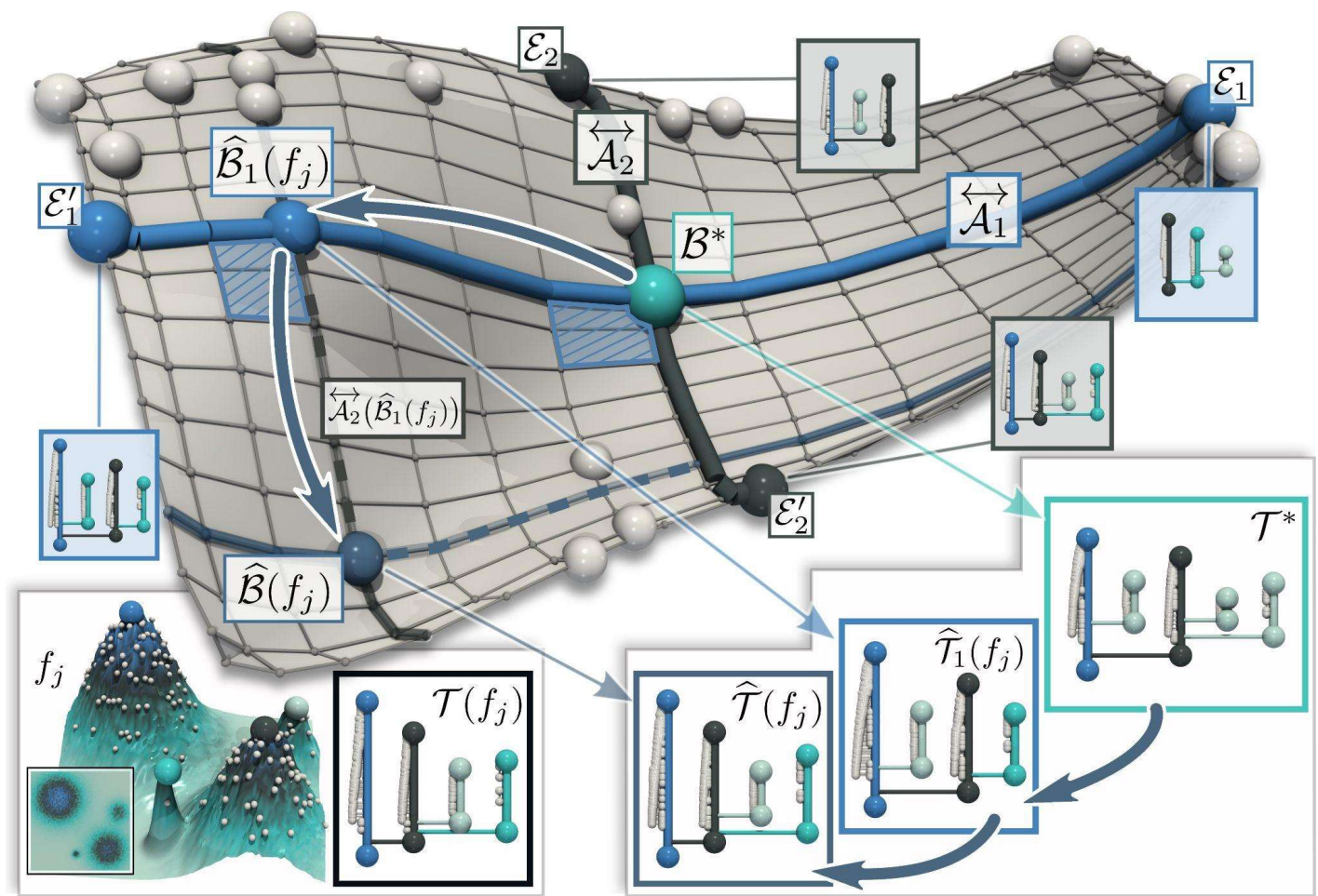
- **Estimation**
 - Axis projection
 - Translated axis



Overview

- **BDT Basis**
 - Barycenter
 - Geodesic axis
 - Orthogonal axes

- **Estimation**
 - Axis projection
 - Translated axis



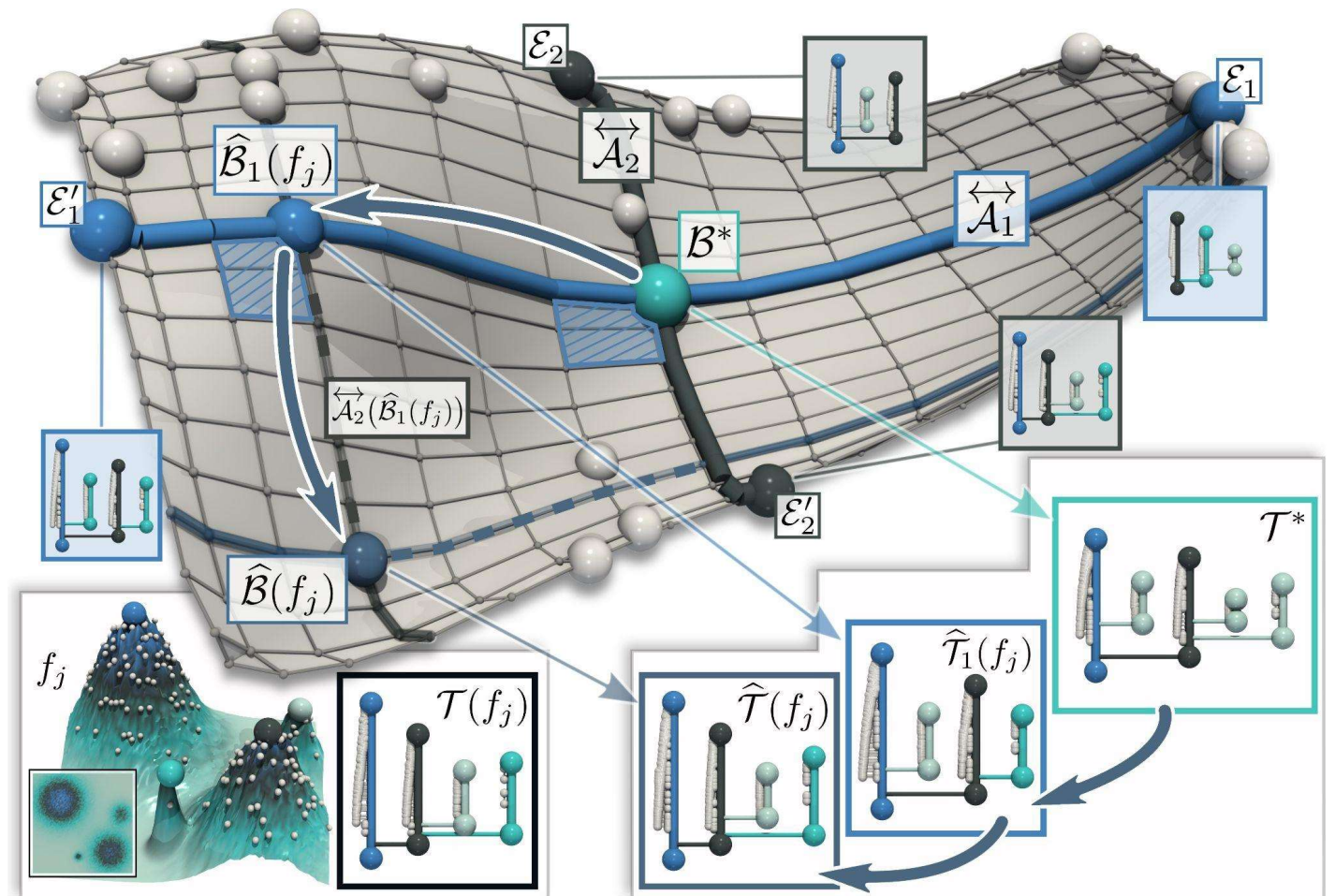
Overview

- **BDT Basis**

- Barycenter
- Geodesic axis
- Orthogonal axes

- **Estimation**

- Axis projection
- Translated axis



$$E_{W_2^{\mathcal{T}}}(\mathcal{B}_{\mathbb{B}}) = \sum_{j=1}^N W_2^{\mathcal{T}} \left(\mathcal{B}(f_j), \mathcal{B}^* + \sum_{i=1}^{d'} \vec{\mathcal{A}}_i(\hat{\mathcal{B}}_i(f_j)) \right)^2$$

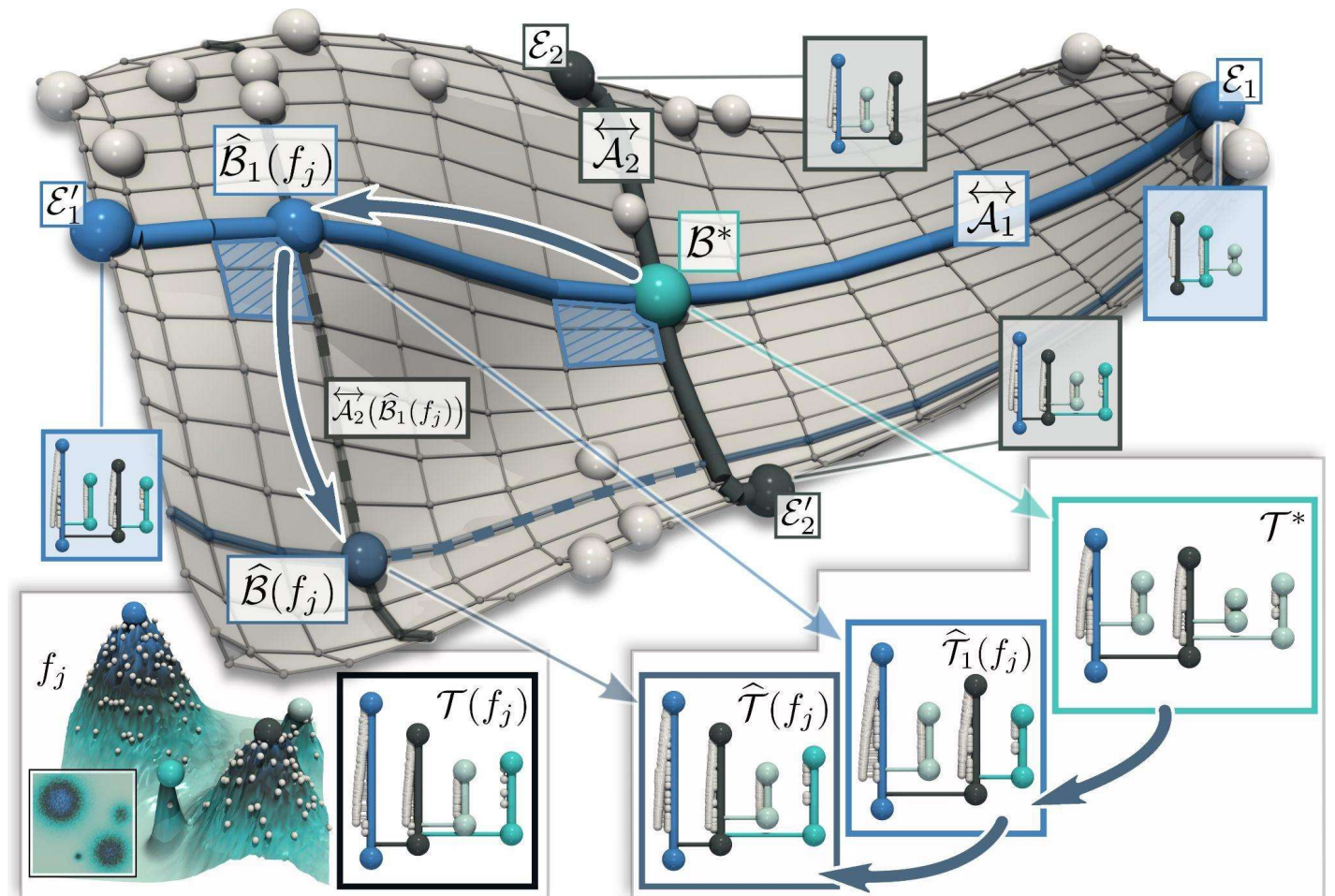
Overview

- **BDT Basis**

- Barycenter
- Geodesic axis
- Orthogonal axes

- **Estimation**

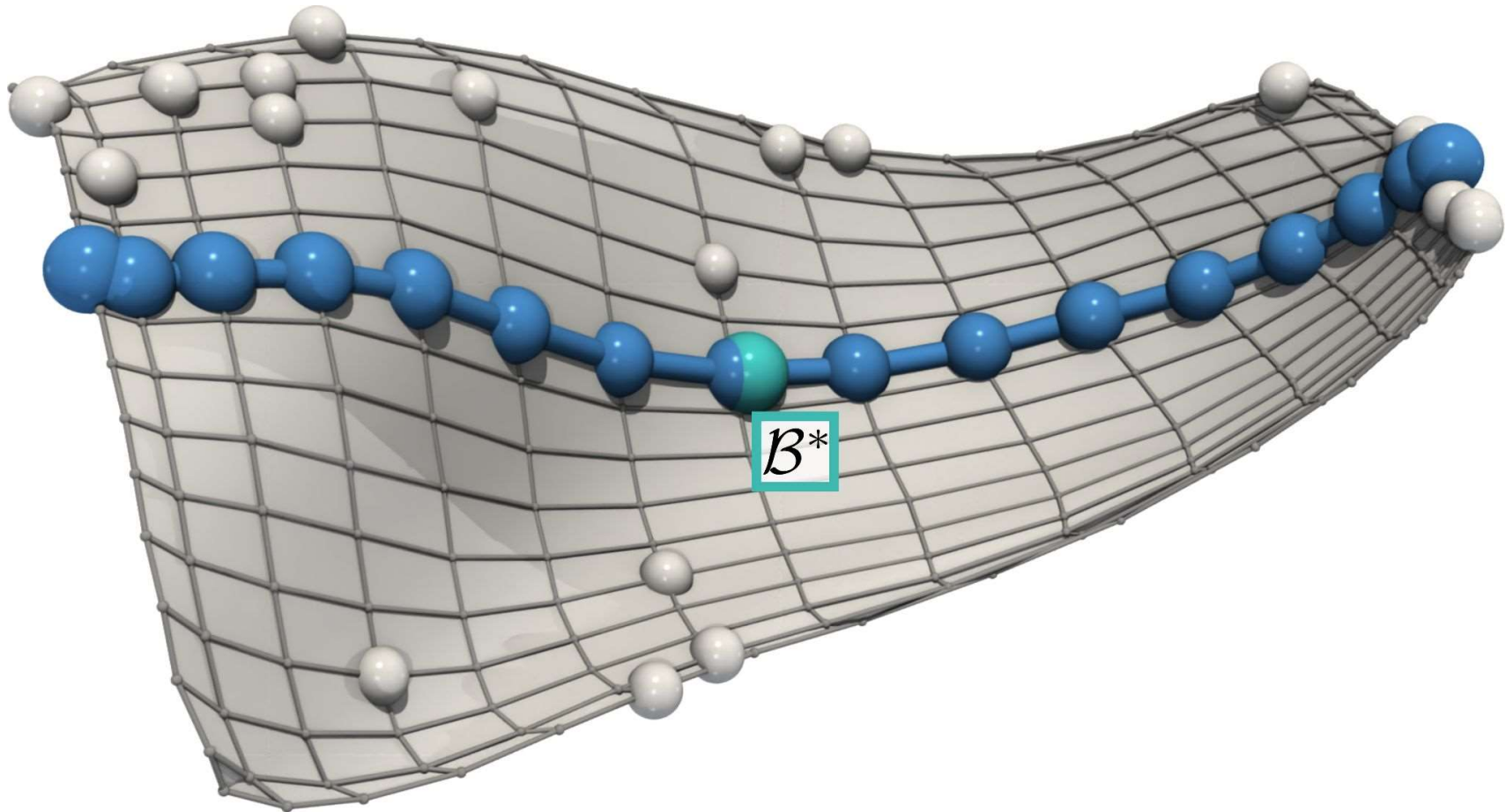
- Axis projection
- Translated axis



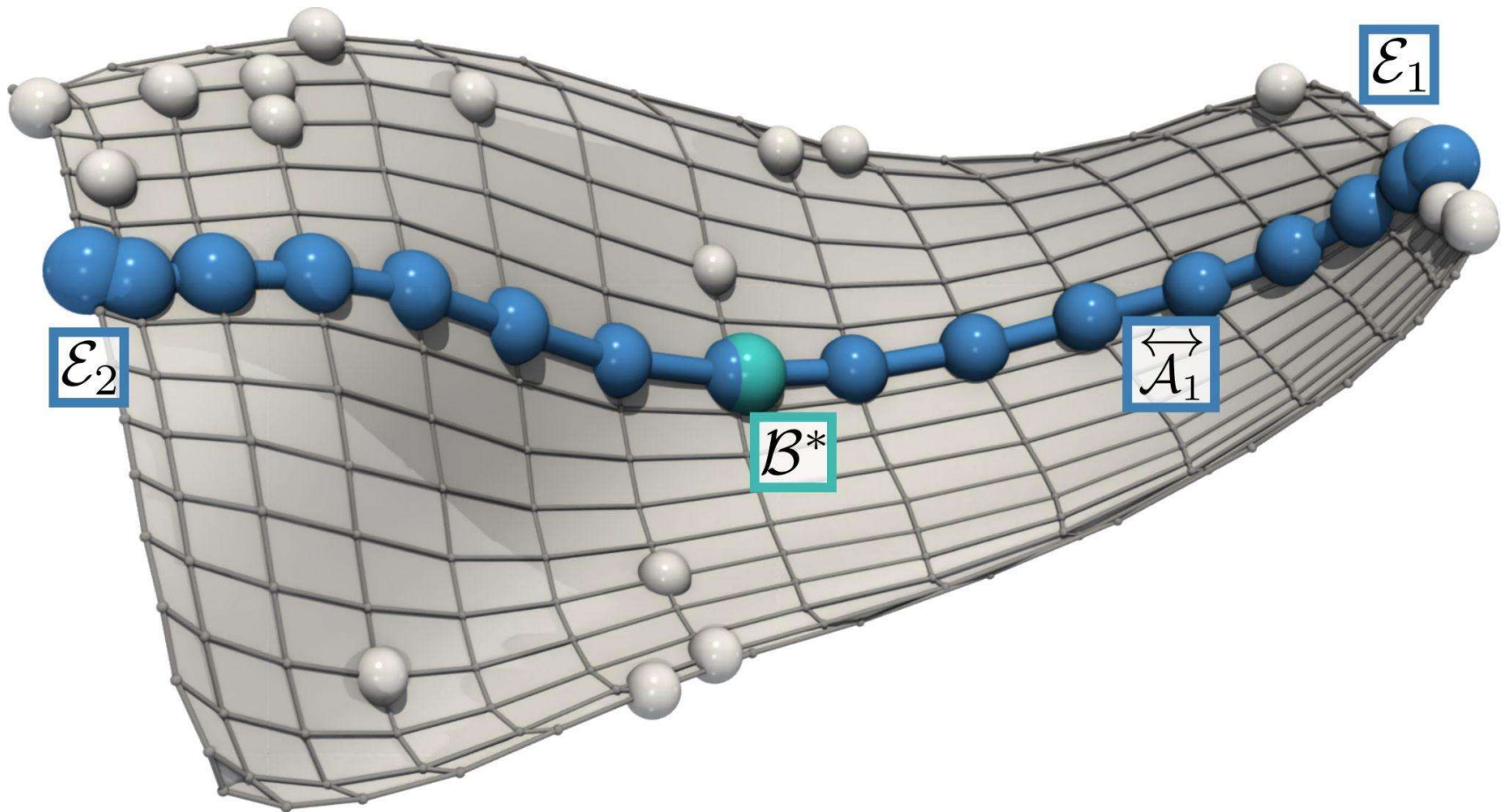
$$E_{W_2^{\mathcal{T}}}(B_{\mathbb{B}}) = \sum_{j=1}^N W_2^{\mathcal{T}} \left(B(f_j), B^* + \sum_{i=1}^{d'} \vec{A}_i(\hat{B}_i(f_j)) \right)^2$$

Algorithm

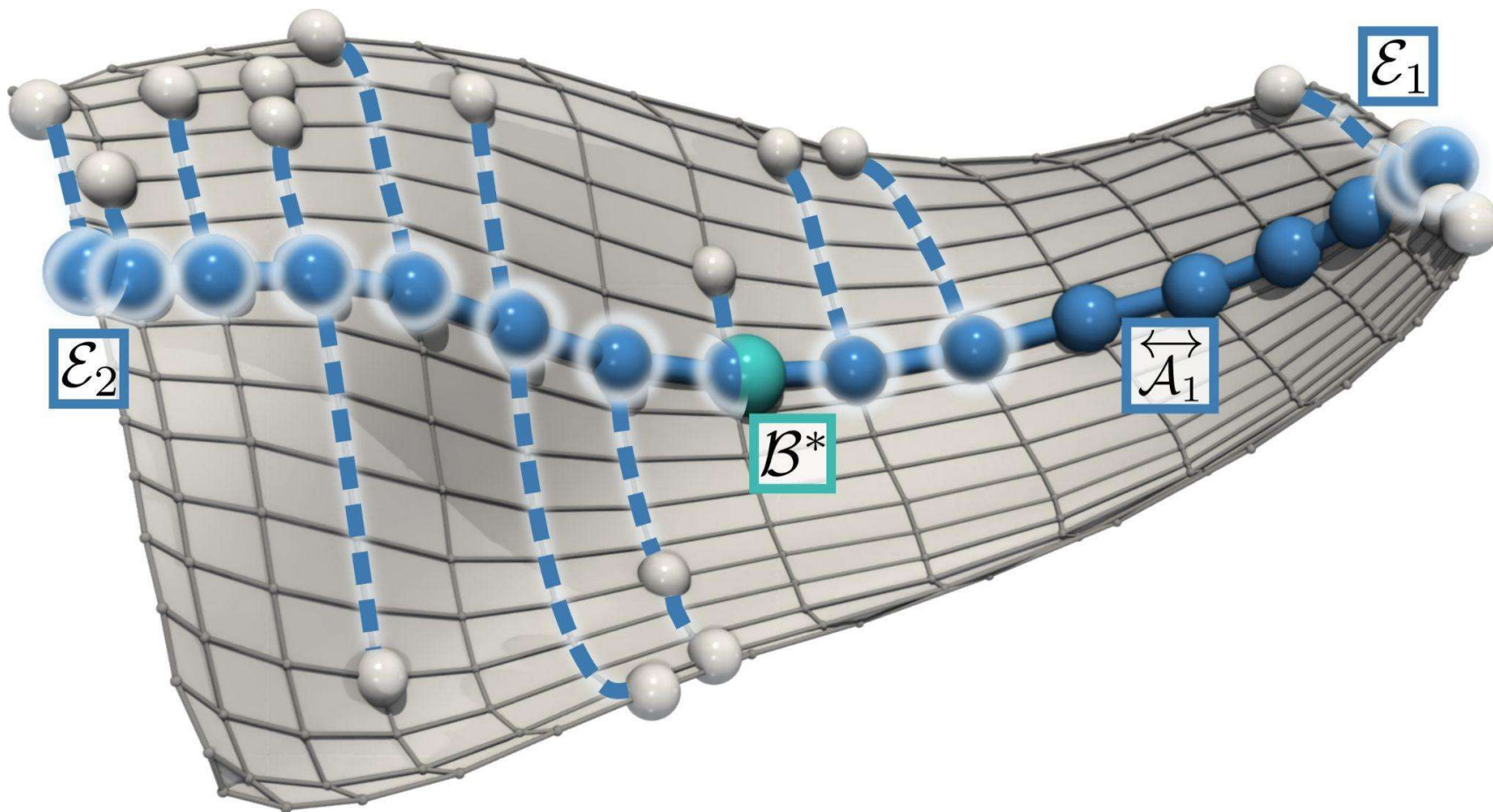
Axis initialization



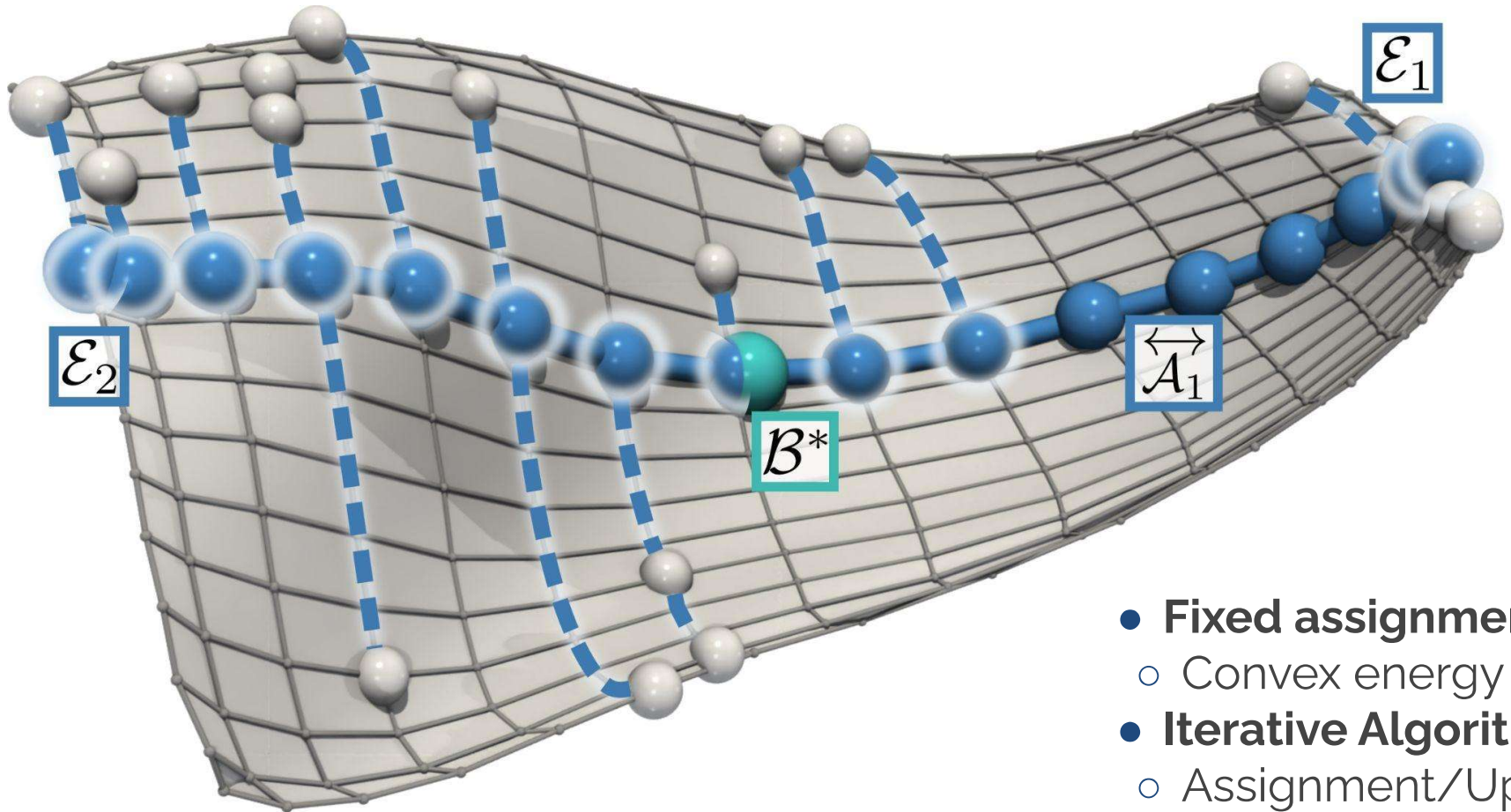
Axis optimization



Axis optimization



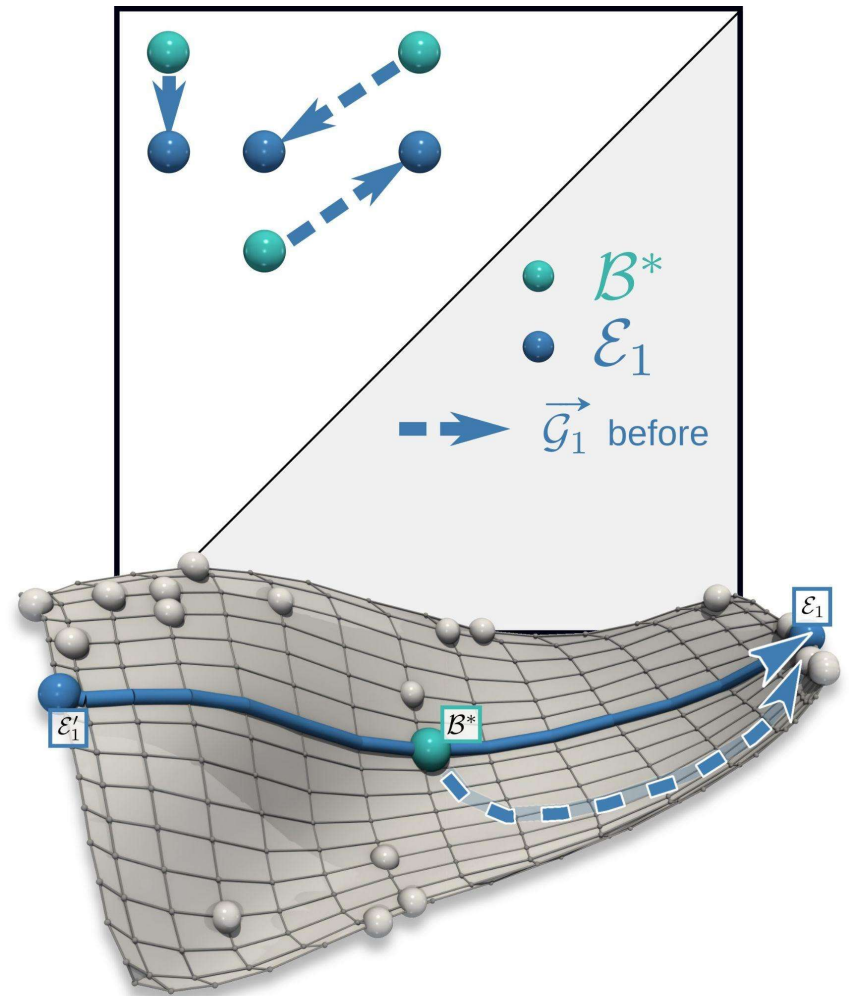
Axis optimization



- **Fixed assignments**
 - Convex energy
- **Iterative Algorithm**
 - Assignment/Update

Geodesic enforcement

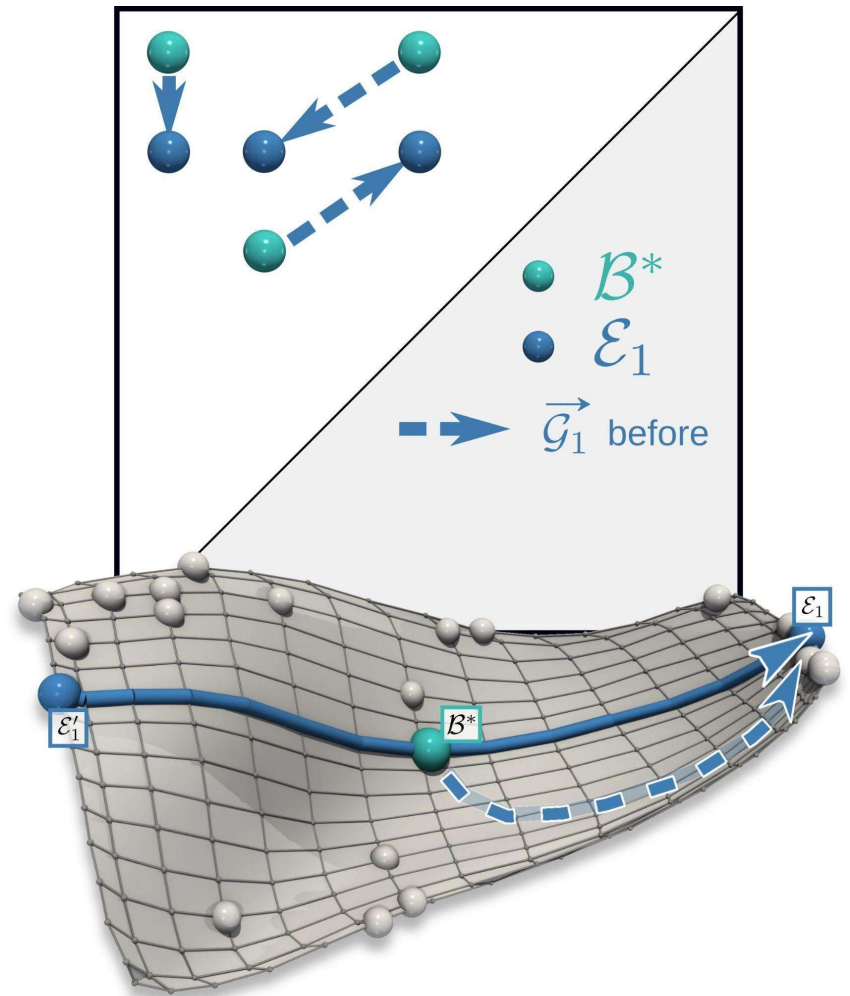
- **Axis optimization**
 - Axis vectors
 - Optimized freely



Geodesic enforcement

- **Axis optimization**

- Axis vectors
 - Optimized freely
 - May not be geodesic



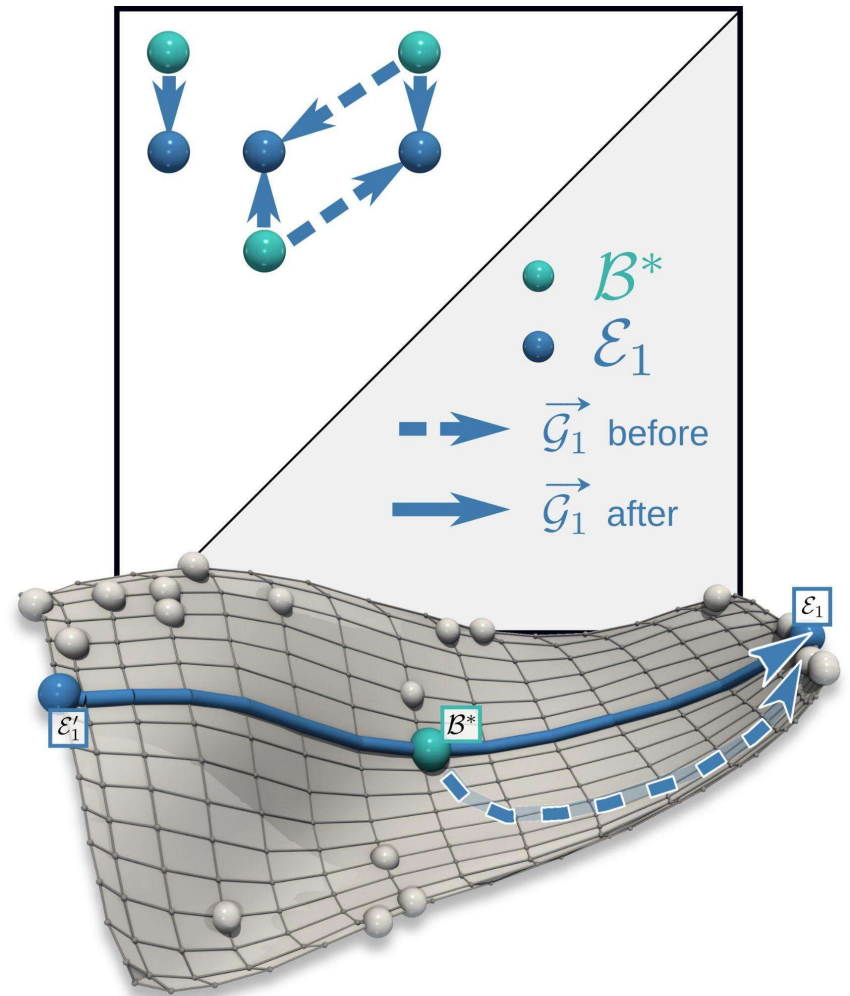
Geodesic enforcement

- **Axis optimization**

- Axis vectors
 - Optimized freely
 - May not be geodesic

- **Geodesic enforcement**

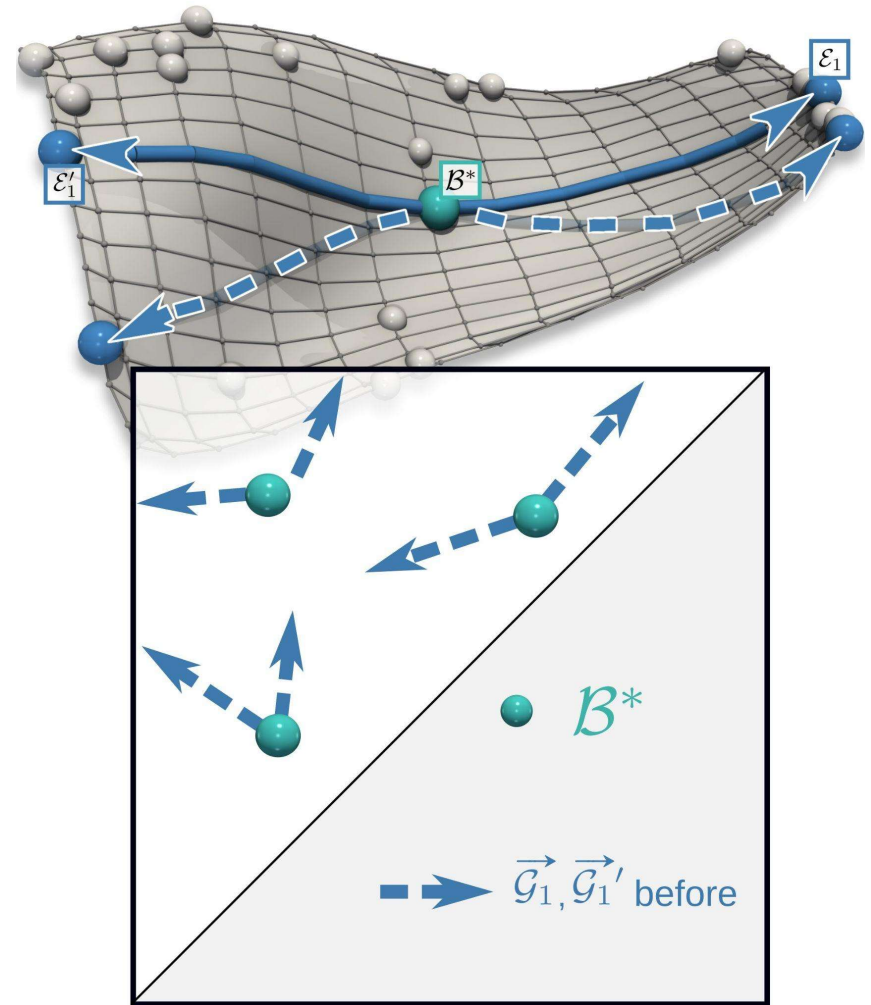
- Re-compute the assignments



Negative collinearity enforcement

- **Axis optimization**

- Axis vectors
 - Optimized independently
 - May not be collinear

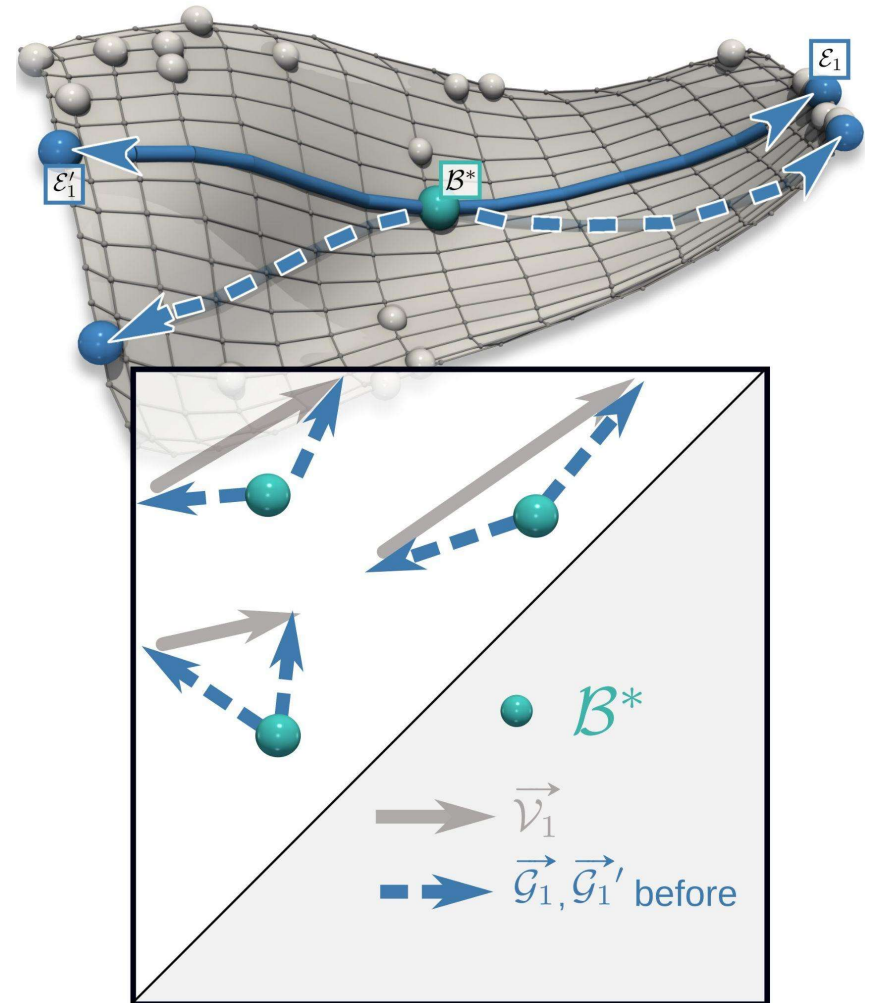


Negative collinearity enforcement

- **Axis optimization**

- Axis vectors
 - Optimized independently
 - May not be collinear

- **Negative collinearity enforcement**



Negative collinearity enforcement

- **Axis optimization**

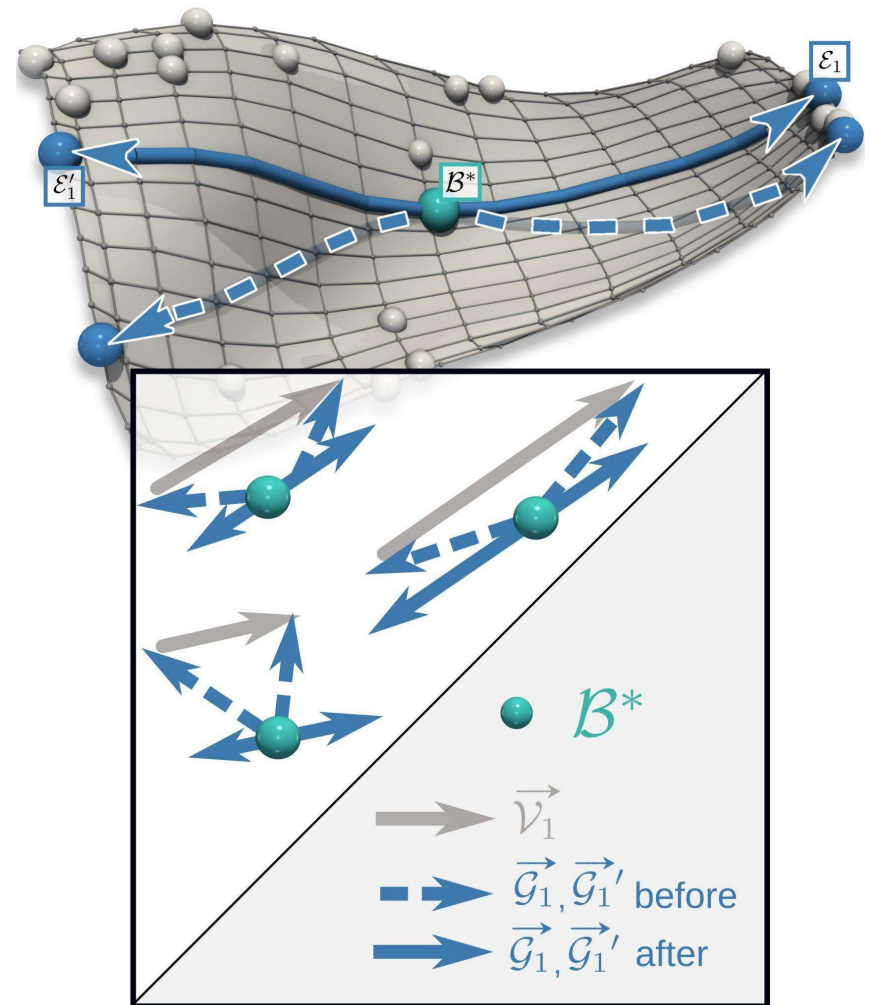
- Axis vectors
 - Optimized independently
 - May not be collinear

- **Negative collinearity enforcement**

- Updated geodesics

$$\vec{g}_{d'} \leftarrow \beta' \times \vec{v}_{d'}$$

$$\vec{g}'_{d'} \leftarrow -(1 - \beta') \times \vec{v}_{d'}$$



Negative collinearity enforcement

- **Axis optimization**

- Axis vectors
 - Optimized independently
 - May not be collinear

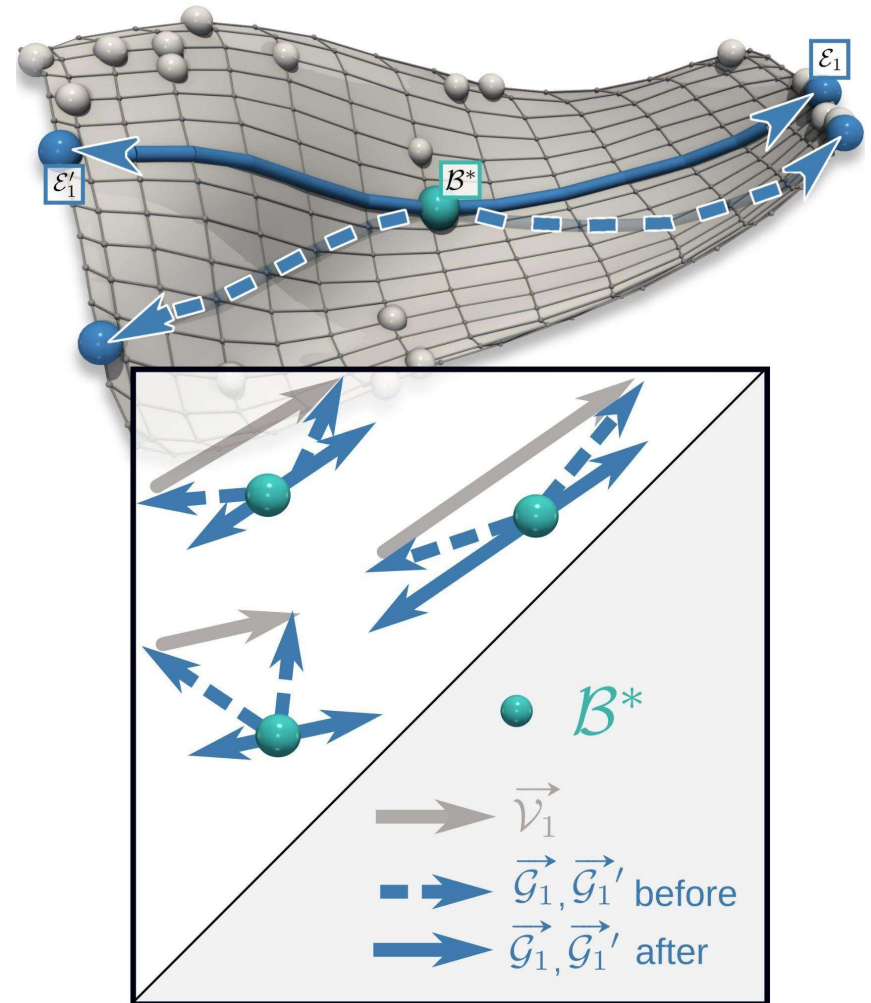
- **Negative collinearity enforcement**

- Updated geodesics

$$\vec{\mathcal{G}}_{d'} \leftarrow \beta' \times \vec{\mathcal{V}}_{d'}$$

$$\vec{\mathcal{G}}'_{d'} \leftarrow -(1 - \beta') \times \vec{\mathcal{V}}_{d'}$$

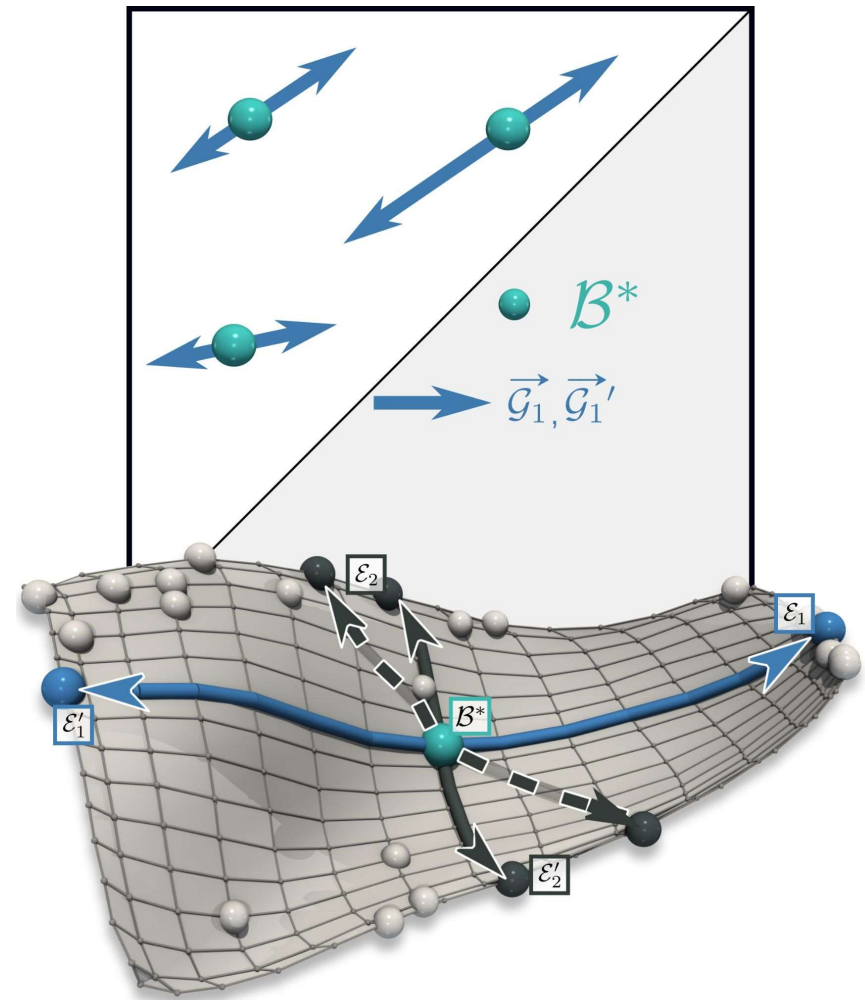
$$\beta' = \|\vec{\mathcal{G}}_{d'}\| / (\|\vec{\mathcal{G}}_{d'}\| + \|\vec{\mathcal{G}}'_{d'}\|)$$



Orthogonality enforcement

- **Axis optimization**

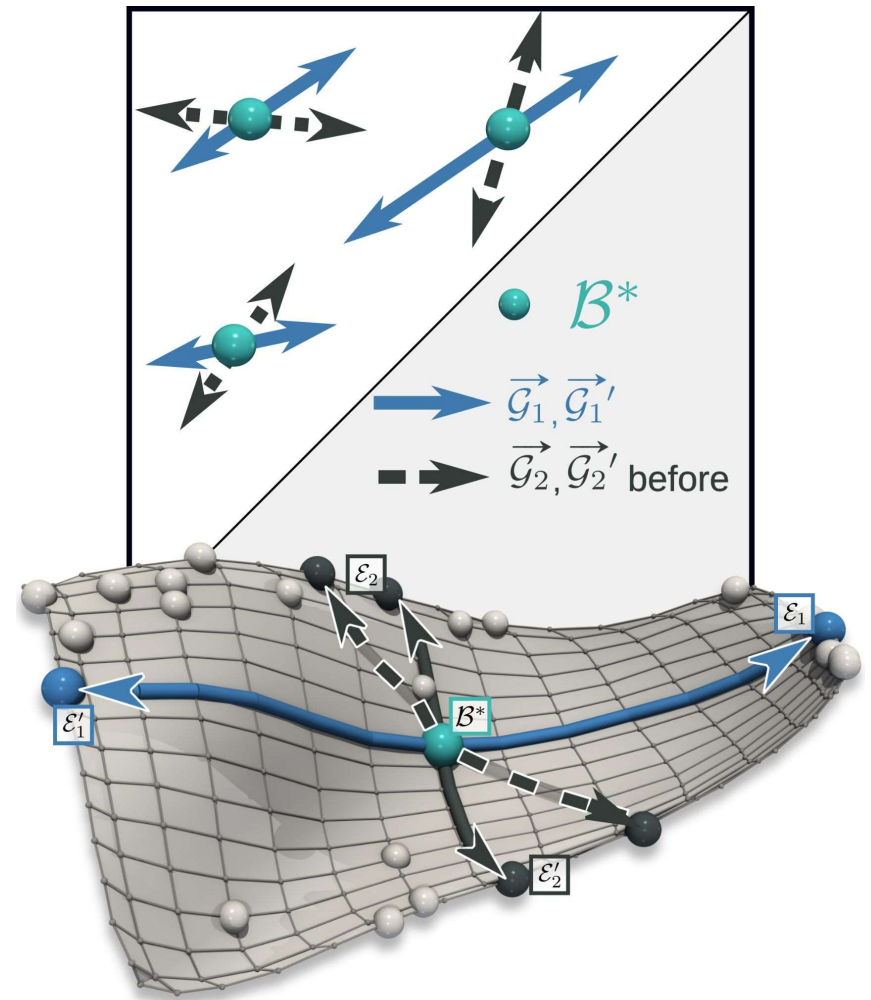
- Axis vectors
 - Optimized freely
 - May not be orthogonal



Orthogonality enforcement

- **Axis optimization**

- Axis vectors
 - Optimized freely
 - May not be orthogonal



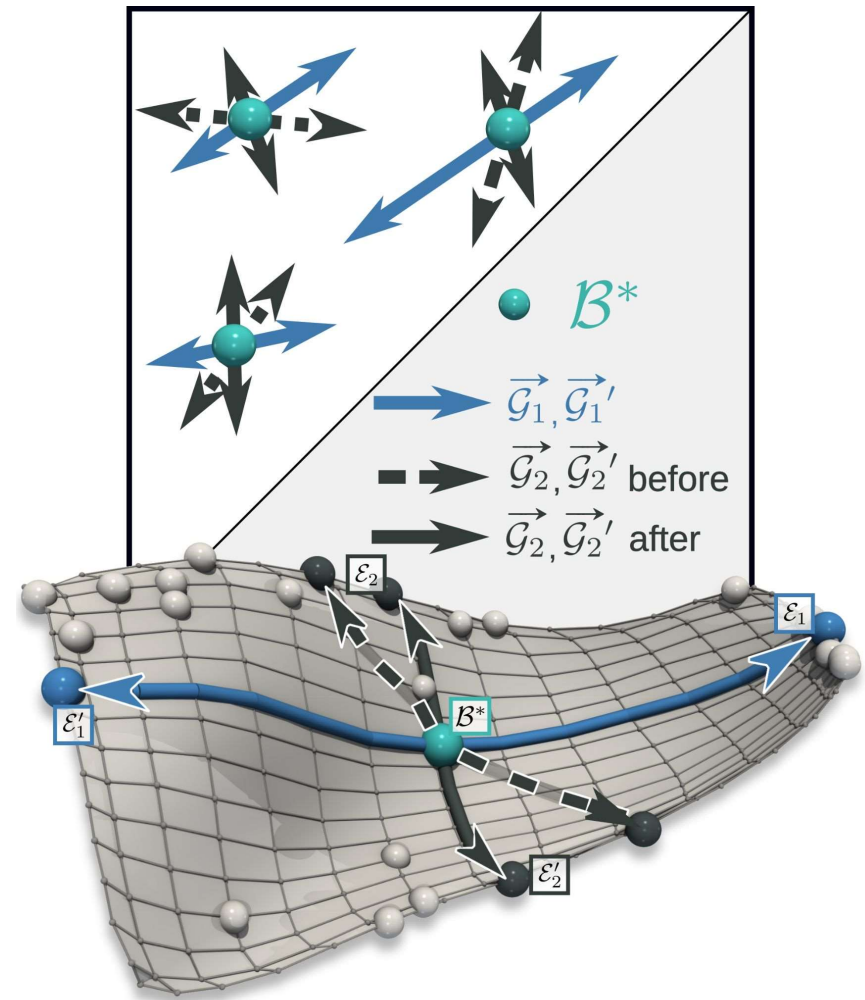
Orthogonality enforcement

- **Axis optimization**

- Axis vectors
 - Optimized freely
 - May not be orthogonal

- **Orthogonality enforcement**

- Gram-Schmidt orthogonalization



Experiments

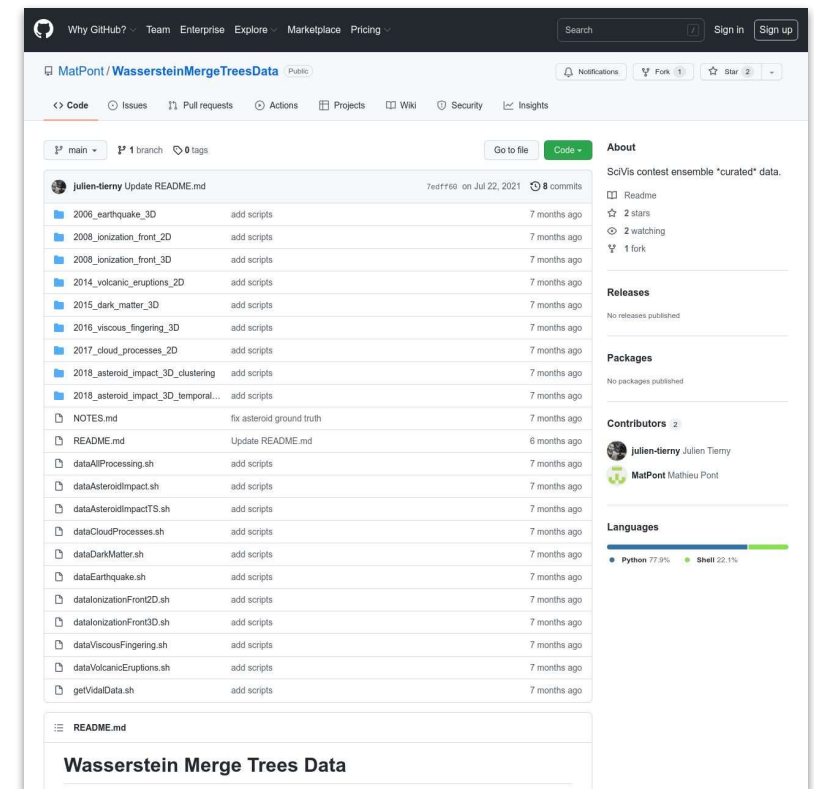
Benchmark data

- <https://github.com/MatPont/WassersteinMergeTreesData>

- Publicly available ensembles
 - <http://sciviscontest.ieeevis.org/>
 - 12 ensembles
 - $7 < N < 48$
 - Branches: Dozens to thousands
- Curation scripts & data & ground-truth

- **Application domains**

- Astrophysics
- Meteorology
- Material physics
- Volcanic measurements
- Fluid mechanics, etc.



Time performance

Dataset	N	$ \mathcal{B} $	PD-PGA			MT-PGA		
			1 c.	20 c.	Speedup	1 c.	20 c.	Speedup
Asteroid Impact (3D)	7	1,295	1,392.17	147.40	9.44	1,180.72	117.97	10.01
Cloud processes (2D)	12	1,209	817.64	61.88	13.21	517.94	38.49	13.46
Viscous fingering (3D)	15	118	86.69	9.17	9.45	42.89	4.71	9.11
Dark matter (3D)	40	2,592	18,388.86	1,366.45	13.46	24,480.42	1,758.04	13.92
Volcanic eruptions (2D)	12	811	460.17	37.99	12.11	1,004.37	81.75	12.29
Ionization front (2D)	16	135	104.74	12.00	8.73	55.73	6.26	8.90
Ionization front (3D)	16	763	3,750.00	300.96	12.46	4,029.71	294.29	13.69
Earthquake (3D)	12	1,203	3,896.52	338.64	11.51	1,973.49	158.12	12.48
Isabel (3D)	12	1,338	1,969.79	164.49	11.98	1,472.54	115.66	12.73
Starting Vortex (2D)	12	124	17.71	2.72	6.51	11.51	1.65	6.98
Sea Surface Height (2D)	48	1,787	12,420.98	670.00	18.54	27,791.00	1,669.52	16.65
Vortex Street (2D)	45	23	18.75	2.69	6.97	35.79	3.93	9.11

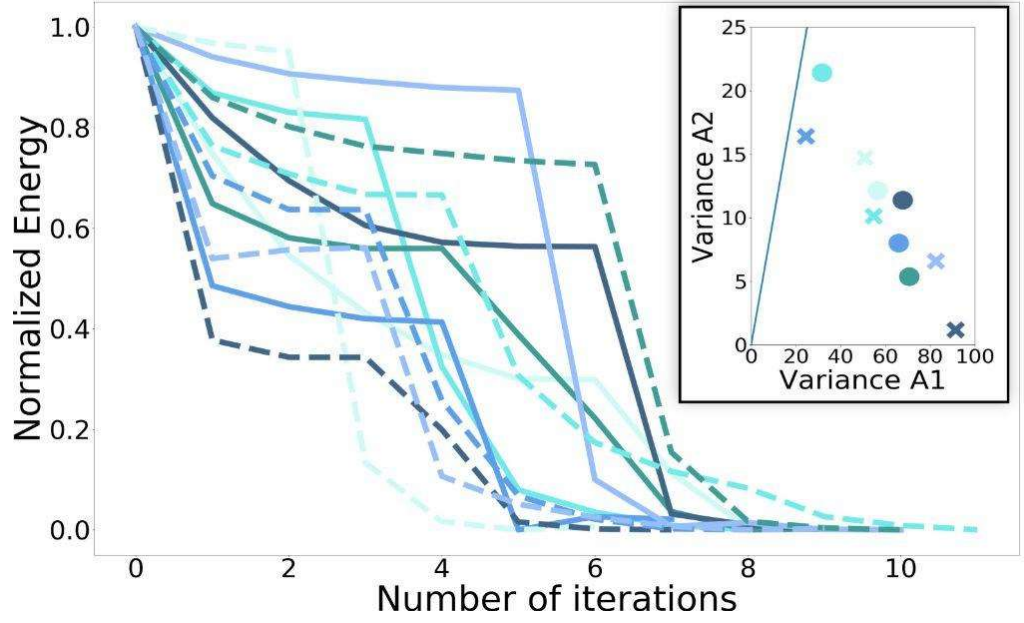
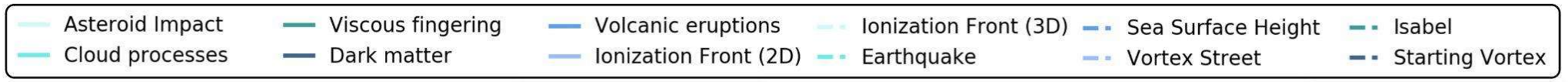
Computation time (s.)

Time performance

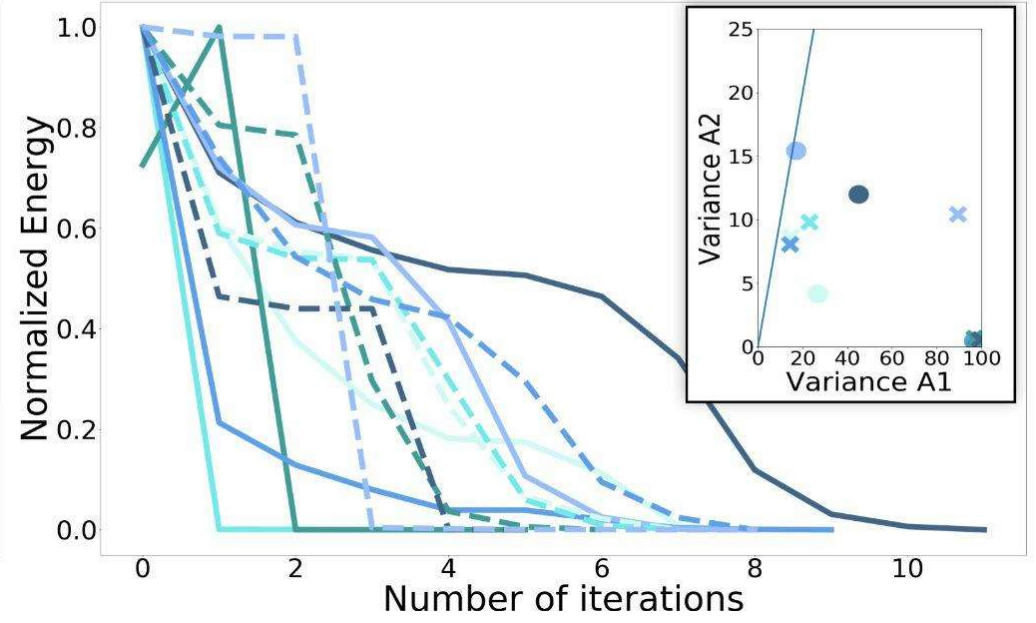
Dataset	N	$ \mathcal{B} $	PD-PGA			MT-PGA		
			1 c.	20 c.	Speedup	1 c.	20 c.	Speedup
Asteroid Impact (3D)	7	1,295	1,392.17	147.40	9.44	1,180.72	117.97	10.01
Cloud processes (2D)	12	1,209	817.64	61.88	13.21	517.94	38.49	13.46
Viscous fingering (3D)	15	118	86.69	9.17	9.45	42.89	4.71	9.11
Dark matter (3D)	40	2,592	18,388.86	1,366.45	13.46	24,480.42	1,758.04	13.92
Volcanic eruptions (2D)	12	811	460.17	37.99	12.11	1,004.37	81.75	12.29
Ionization front (2D)	16	135	104.74	12.00	8.73	55.73	6.26	8.90
Ionization front (3D)	16	763	3,750.00	300.96	12.46	4,029.71	294.29	13.69
Earthquake (3D)	12	1,203	3,896.52	338.64	11.51	1,973.49	158.12	12.48
Isabel (3D)	12	1,338	1,969.79	164.49	11.98	1,472.54	115.66	12.73
Starting Vortex (2D)	12	124	17.71	2.72	6.51	11.51	1.65	6.98
Sea Surface Height (2D)	48	1,787	12,420.98	670.00	18.54	27,791.00	1,669.52	16.65
Vortex Street (2D)	45	23	18.75	2.69	6.97	35.79	3.93	9.11

Computation time (s.)

Energy evolution



PD-PGA

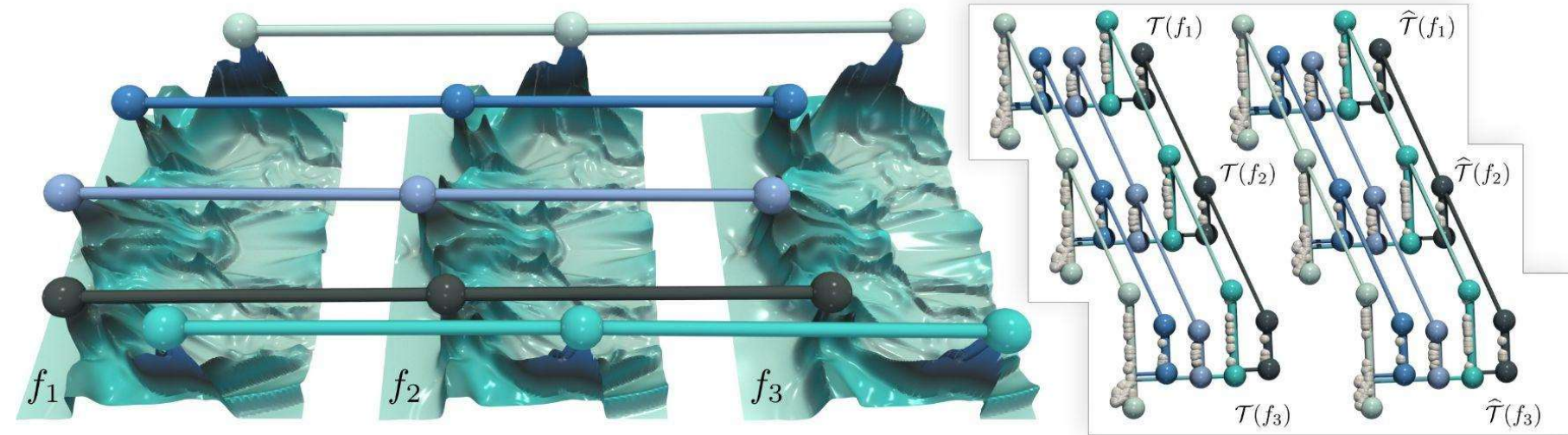


MT-PGA

Applications

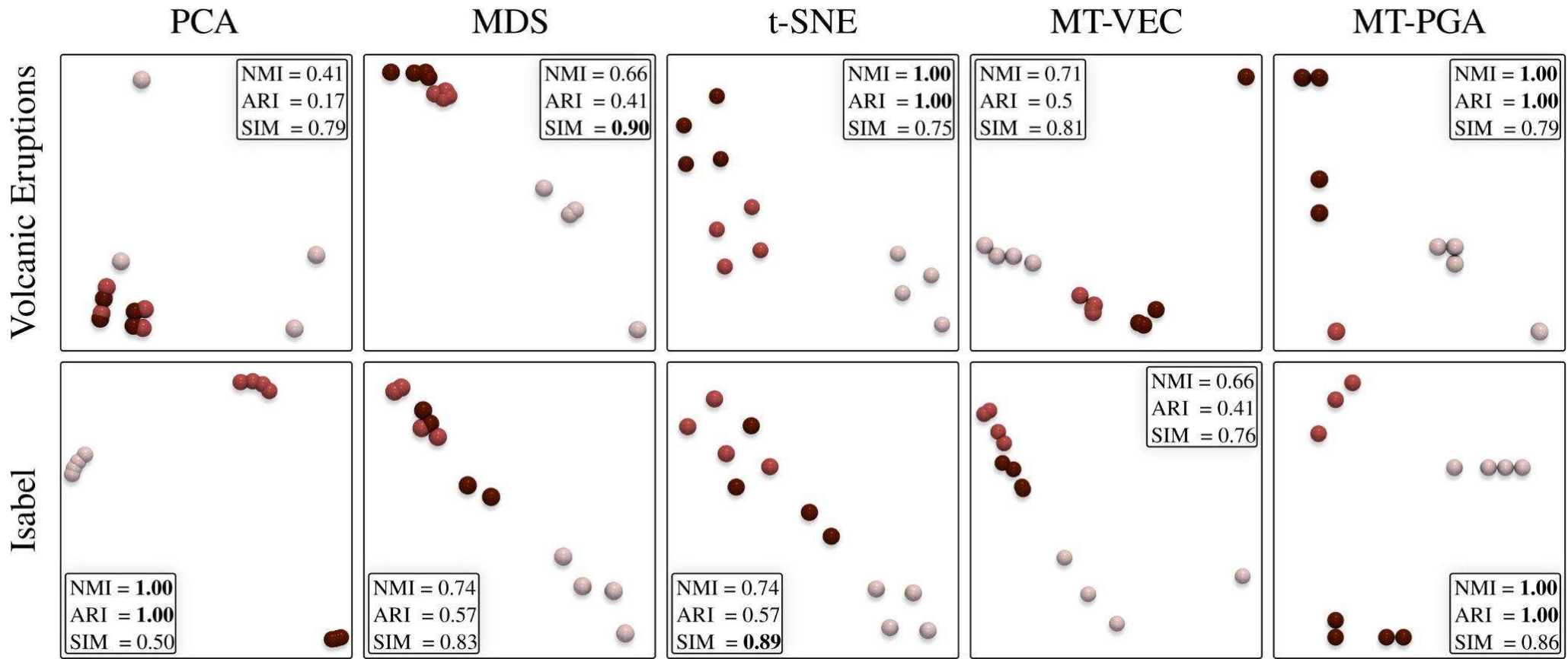
Data reduction

- Only store
 - The basis
 - The coordinates

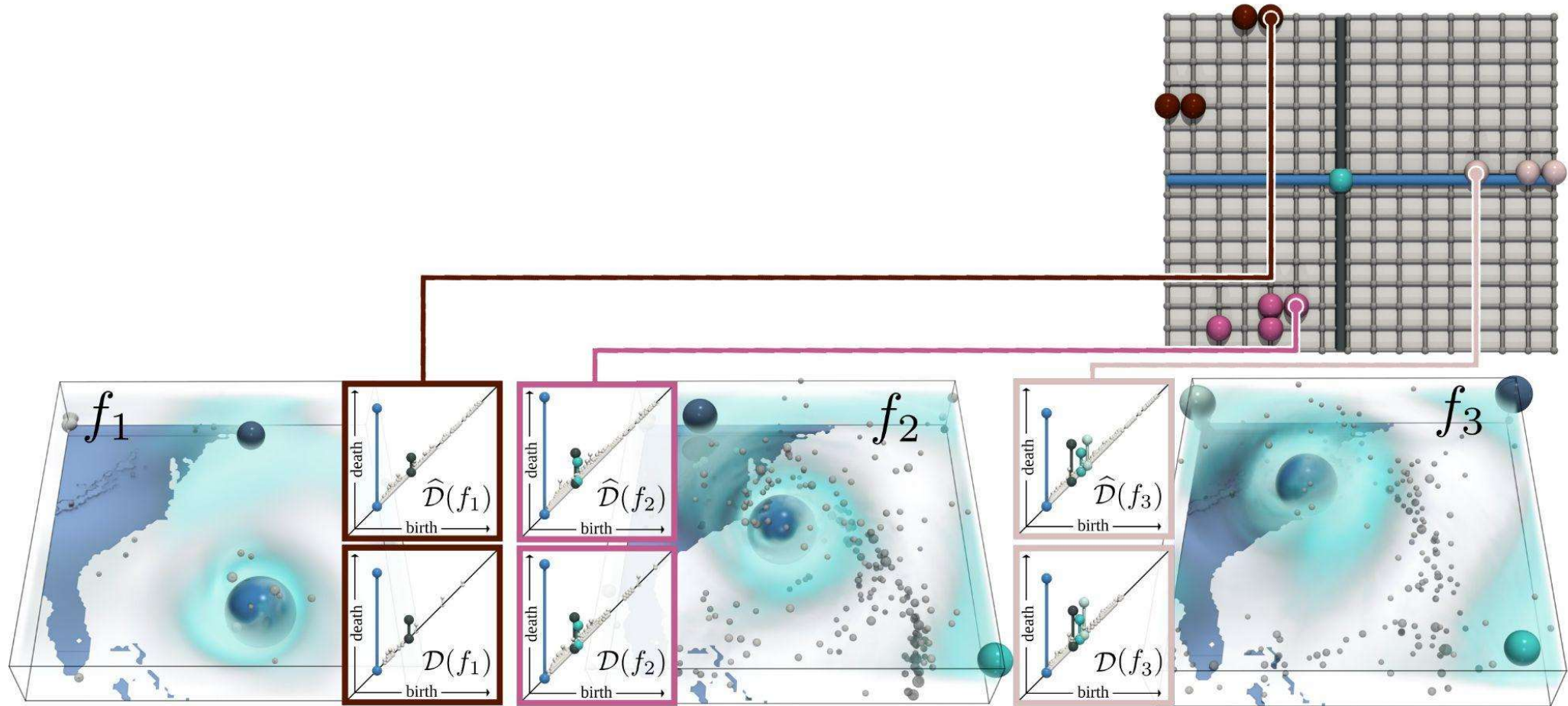


Compression factor: 5.12

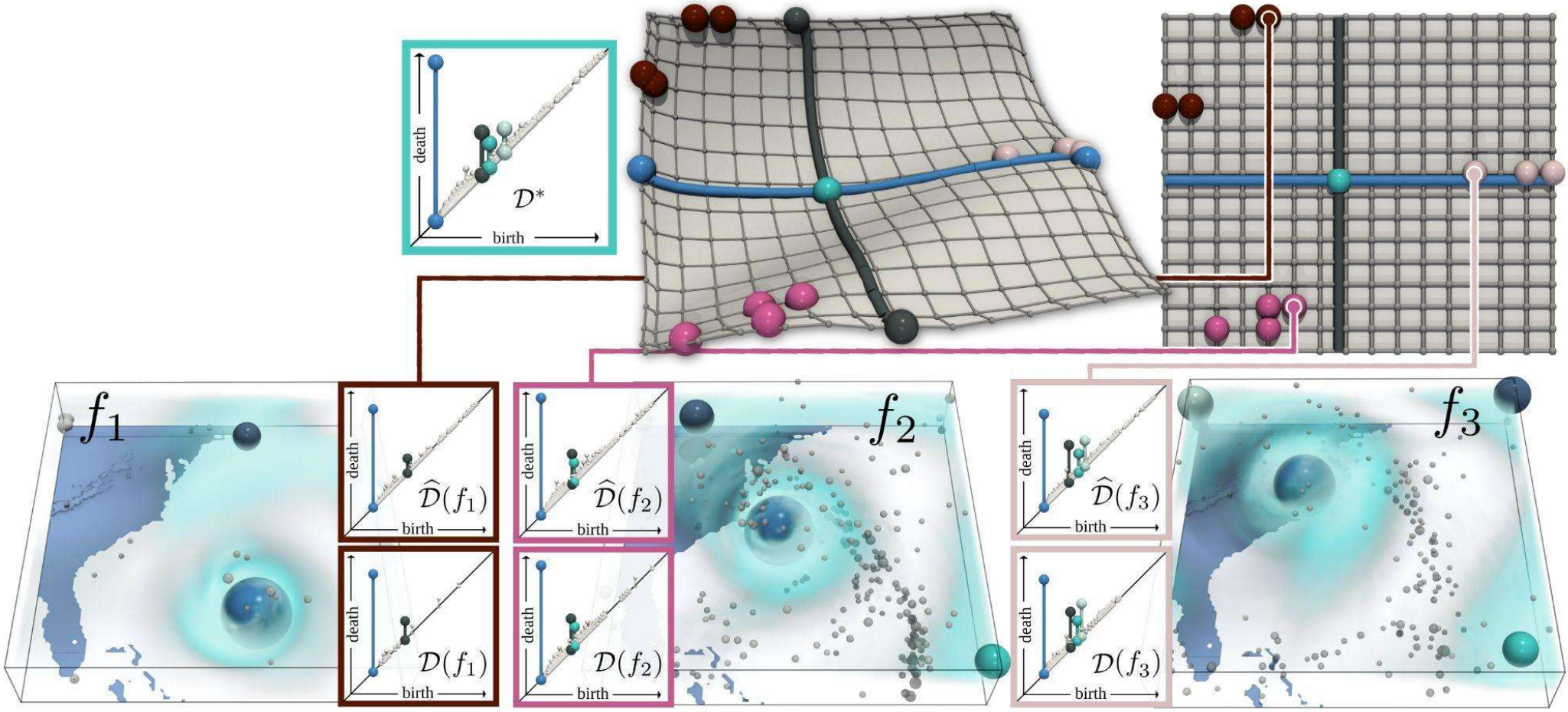
Dimensionality reduction



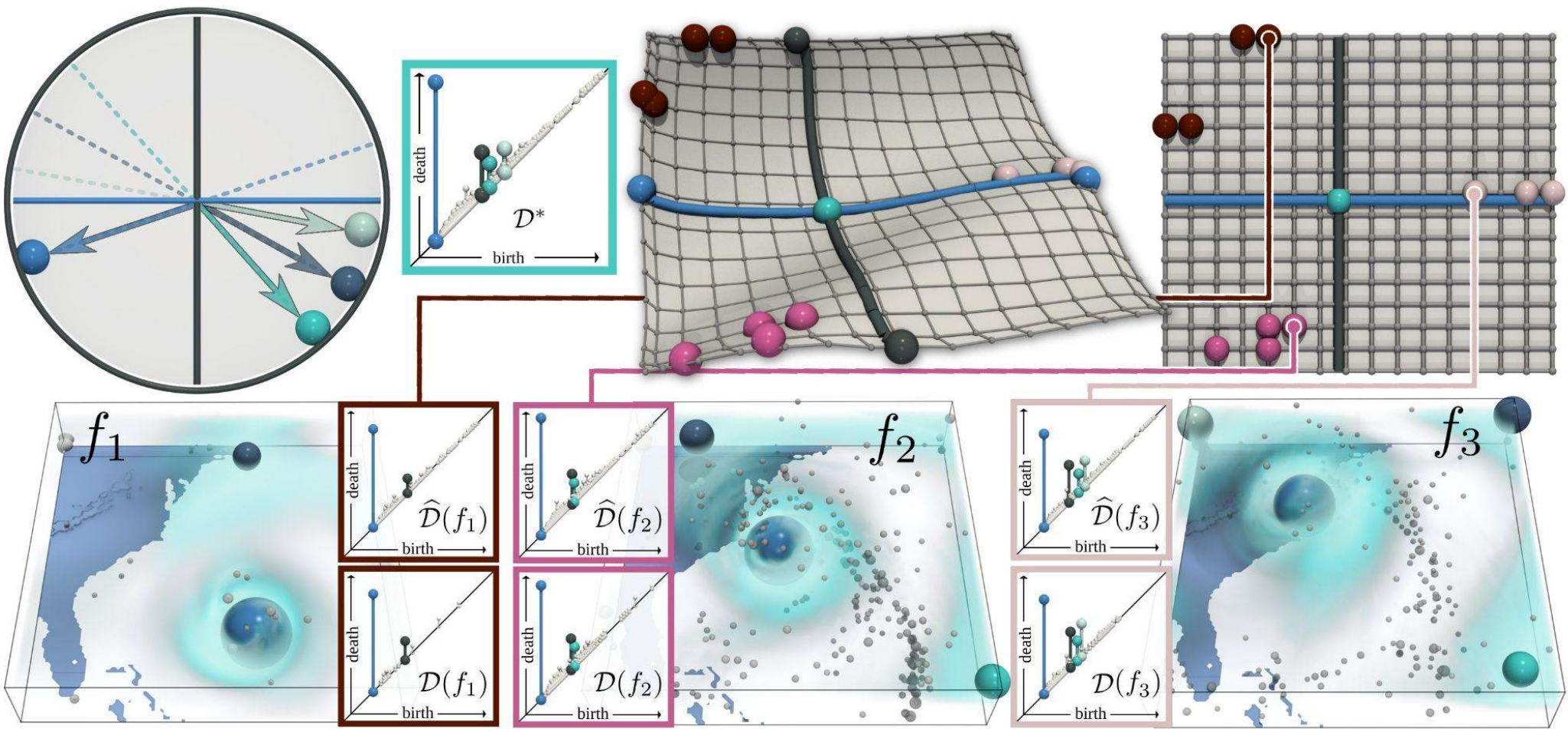
PD-PGA



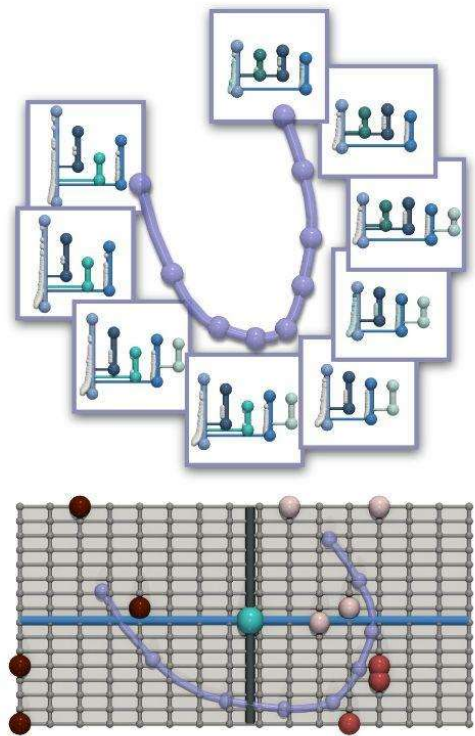
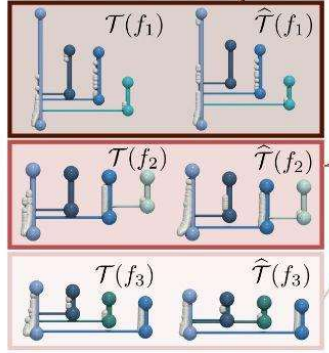
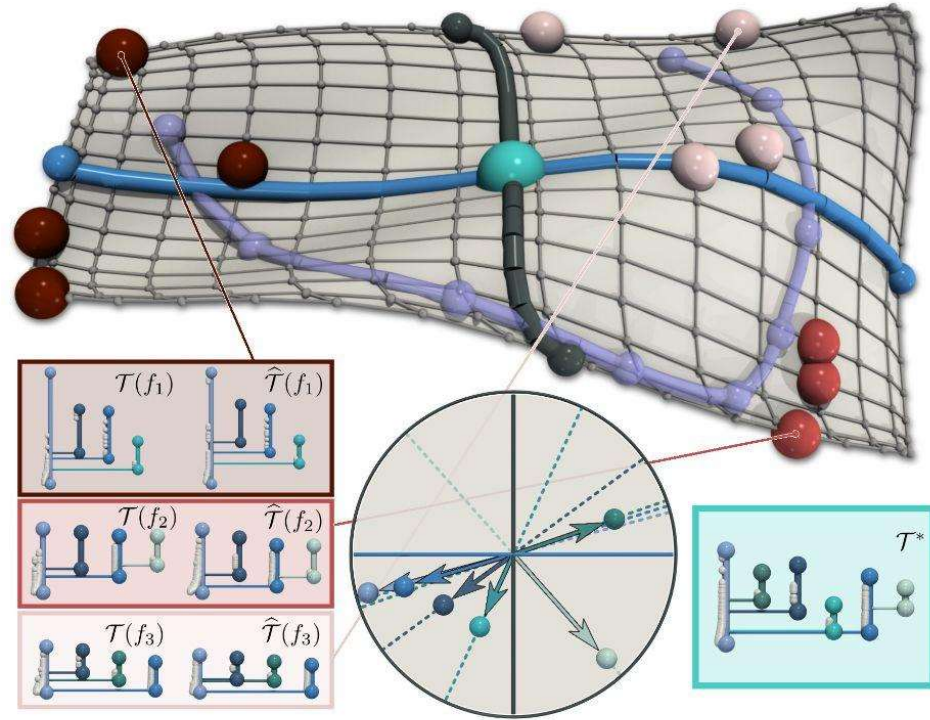
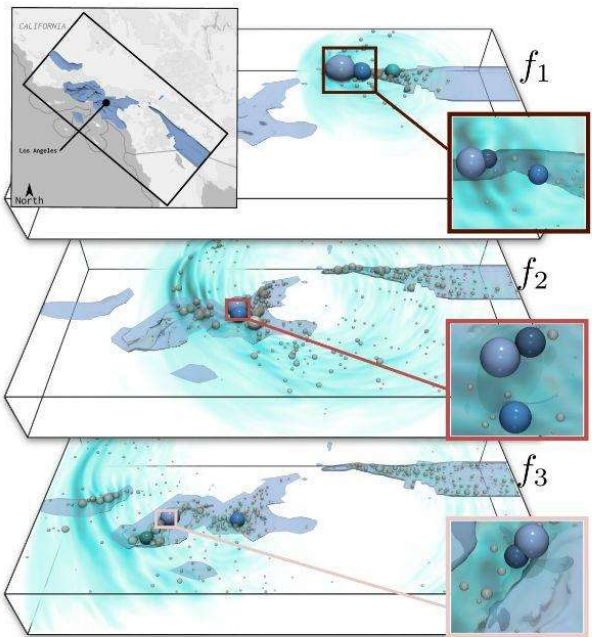
PD-PGA



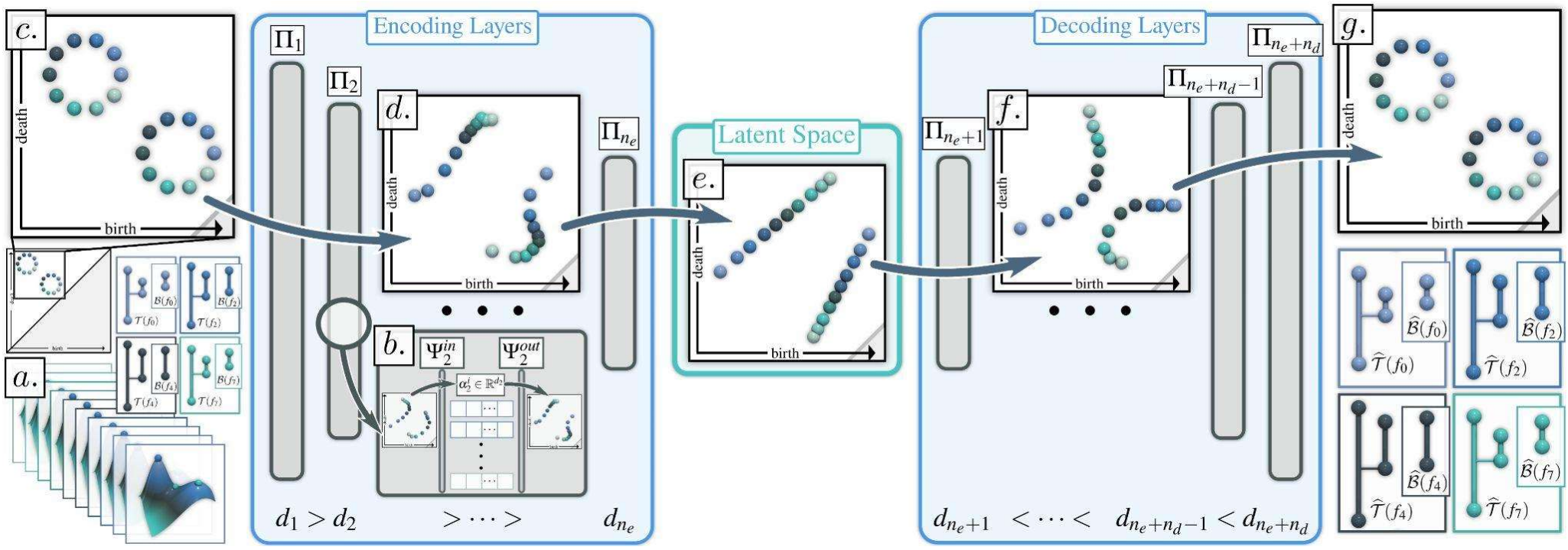
PD-PGA



MT-PGA



Beyond linear combinations

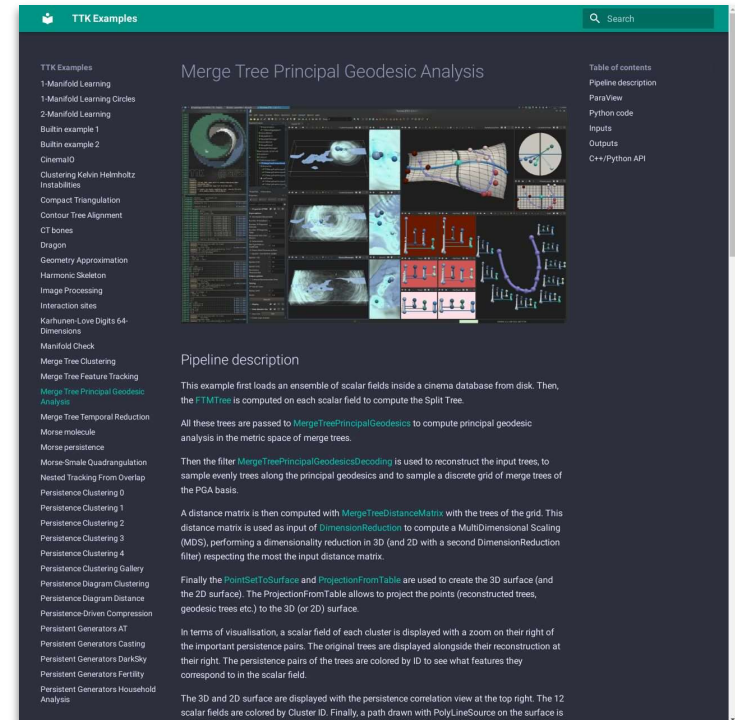


- **Wasserstein Auto-Encoders of Merge-Trees (and Persistence Diagrams)**
 - IEEE TVCG 2024 (presented at IEEE VIS 2024)

Try it with TTK!

- <https://topology-tool-kit.github.io/installation.html>
 - ParaView >= 5.10, Ubuntu, Anaconda, others

- <https://topology-tool-kit.github.io/examples/>
 - Screenshots
 - Pipeline description
 - ParaView statefile
 - Python script
 - C++/Python API pointers



The screenshot displays the TTK Examples website interface. The main content area is titled "Merge Tree Principal Geodesic Analysis" and features a grid of visualization panels. The left sidebar lists various TTK examples, with "Merge Tree Principal Geodesic Analysis" highlighted in green. The right sidebar contains a search bar and a table of contents with links for "Table of contents", "Pipeline description", "ParaView", "Python code", "Inputs", "Outputs", and "C++/Python API".

Pipeline description

This example first loads an ensemble of scalar fields inside a cinema database from disk. Then, the **FTMTree** is computed on each scalar field to compute the Split Tree.

All these trees are passed to **MergeTreePrincipalGeodesics** to compute principal geodesic analysis in the metric space of merge trees.

Then the filter **MergeTreePrincipalGeodesicsDecoding** is used to reconstruct the input trees, to sample evenly trees along the principal geodesics and to sample a discrete grid of merge trees of the PGA basis.

A distance matrix is then computed with **MergeTreeDistanceMatrix** with the trees of the grid. This distance matrix is used as input of **DimensionReduction** to compute a MultiDimensional Scaling (MDS), performing a dimensionality reduction in 3D (and 2D) with a second **DimensionReduction** filter) respecting the most the input distance matrix.

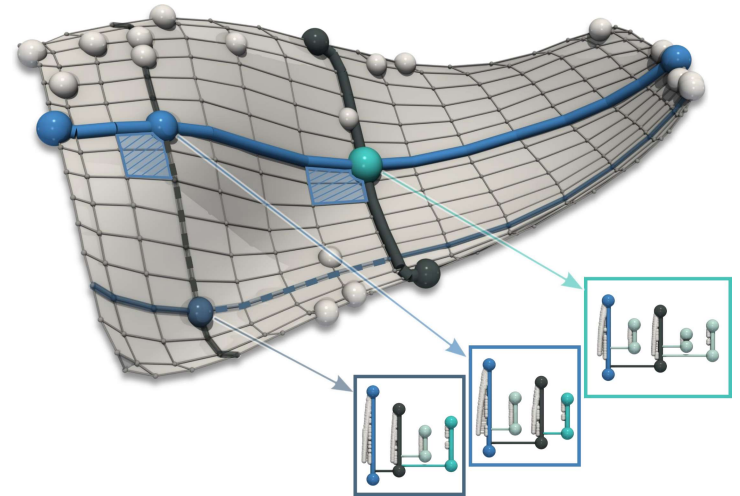
Finally the **PointSetToSurface** and **ProjectionFromTable** are used to create the 3D surface (and the 2D surface). The **ProjectionFromTable** allows to project the points (reconstructed trees, geodesic trees etc.) to the 3D (or 2D) surface.

In terms of visualisation, a scalar field of each cluster is displayed with a zoom on their right of the important persistence pairs. The original trees are displayed alongside their reconstruction at their right. The persistence pairs of the trees are colored by ID to see what features they correspond to in the scalar field.

The 3D and 2D surface are displayed with the persistence correlation view at the top right. The 12 scalar fields are colored by Cluster ID. Finally, a path drawn with **PolyLineSource** on the surface is

Conclusion

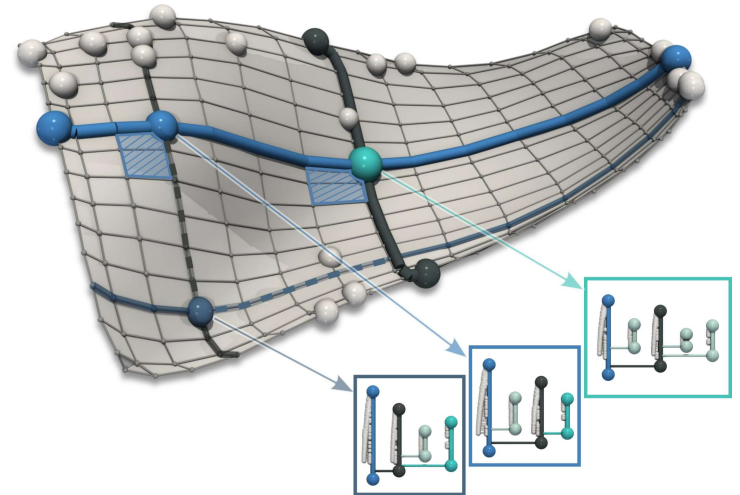
- **Computational framework for Principal Geodesic Analysis**
 - Merge trees
 - Persistence diagrams



Conclusion

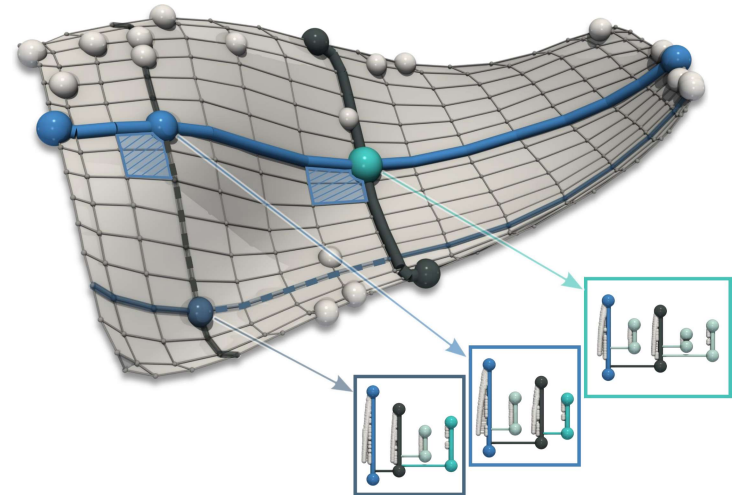
- **Computational framework for Principal Geodesic Analysis**
 - Merge trees
 - Persistence diagrams

- **Applications to ensemble analysis**



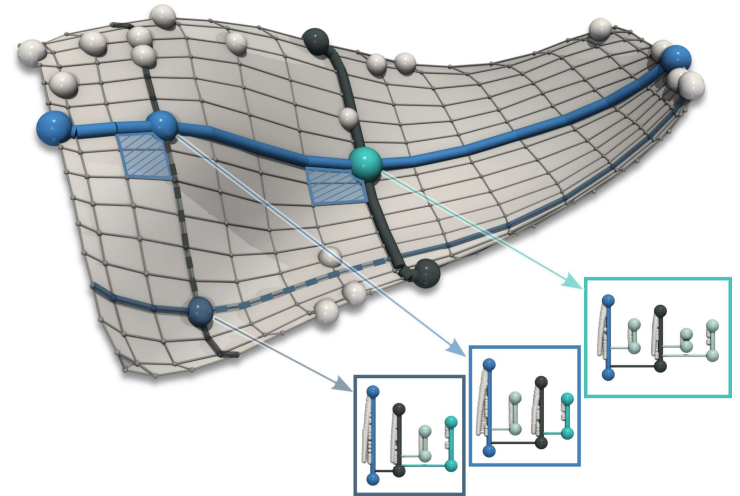
Conclusion

- **Computational framework for Principal Geodesic Analysis**
 - Merge trees
 - Persistence diagrams
- **Applications to ensemble analysis**
- **Replicable paper**
 - <http://www.replicabilitystamp.org/>
 - Exact code & data



Conclusion

- **Computational framework for Principal Geodesic Analysis**
 - Merge trees
 - Persistence diagrams
- **Applications to ensemble analysis**
- **Replicable paper**
 - <http://www.replicabilitystamp.org/>
 - Exact code & data
- **Perspectives**
 - Other topological descriptors
 - Towards more advanced ensemble analysis



Sylvain GERBAUD



A la recherche d'un poste de MCF – CNU 27 – pour la rentrée de septembre 2025

2020 - 2023 Thèse au laboratoire XLIM à Poitiers, équipe Informatique Graphique

2023 - 2024 ATER – Université de Poitiers

2024 - 2025 Post-Doctorant ERC TORI, LIP6 – Université de la Sorbonne
CV HAL – <https://cv.hal.science/sylvain-gerbaud>

Coordonnées

Tél. 06 44 88 47 82

Mail. sylvain.gerbaud@lip6.fr

Thématiques de recherche

Topologie – Modélisation volumique –

HPC – Imagerie Médicale

Thanks!

- Curated data
 - <https://github.com/MatPont/WassersteinMergeTreesData>
- Paper implementation
 - <https://github.com/MatPont/MT-PGA>
- TTK
 - <https://topology-tool-kit.github.io/>
 - <https://topology-tool-kit.github.io/examples/>
 - <https://github.com/topology-tool-kit/ttk/>
- ERC Project TORI
 - <https://erc-tori.github.io/>
 - We're hiring!!!



Metric spaces for Persistence diagrams

- Vast literature!

Stability of Persistence Diagrams

David Cohen-Steiner
David Cohen-Steiner
Clark University, Vermont
North Carolina, USA
dcohen@copeland.uvm.edu

Herbert Edelsbrunner
David Edelsbrunner
Clark University, Vermont
Rutledge Springs, VT
North Carolina, USA
edels@cs.ucla.edu

John Harer
David Edelsbrunner
Clark University, Vermont
North Carolina, USA
harer@math.duke.edu

ABSTRACT
The persistence diagram of a real-valued function on a topological space is a multiset of points in the extended plane. We prove that small perturbations of the function only result in small changes in the diagram. We apply this result to extending the homology of a space to a metric space and to comparing and classifying persistence diagrams.

Categories and Subject Descriptors
F.2.1 Analysis of Algorithms and Problem Complexity: Nonnumerical Algorithms and Problems—Computational problems and complexity; D.3.1 Formal Languages and Theory of Computation—Complexity Theory

General Terms
Algorithms, Theory

Keywords
Continuum topology, distance functions, homology groups, persistence stability

1. INTRODUCTION
The persistence diagram of a real-valued function on a topological space can be viewed as a multiset of points in the extended plane. We prove that small perturbations of the function only result in small changes in the diagram. We apply this result to extending the homology of a space to a metric space and to comparing and classifying persistence diagrams.

2. THE PERSISTENCE DIAGRAM
Let X be a topological space and $f: X \rightarrow \mathbb{R}$ a real-valued function. The persistence diagram of f is the set of points in the extended plane that are the birth and death times of the homology of the sublevel sets of f . We prove that small perturbations of the function only result in small changes in the diagram. We apply this result to extending the homology of a space to a metric space and to comparing and classifying persistence diagrams.

3. APPLICATIONS
We apply the stability result to extending the homology of a space to a metric space and to comparing and classifying persistence diagrams.

4. CONCLUSION
We conclude that small perturbations of the function only result in small changes in the diagram. We apply this result to extending the homology of a space to a metric space and to comparing and classifying persistence diagrams.

5. REFERENCES
[1] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, "Stability of persistence diagrams," *Mathematical Sciences Research Institute Publications*, vol. 104, pp. 165–191, 2005.

arXiv:1606.03574v1 [cs.CG] 10 Jun 2016

Geometry Helps to Compare Persistence Diagrams

Michael Kerber¹ Dmitry Morozov¹ Arnan Nijssen²

ABSTRACT
Ranking persistence diagrams requires the abstract comparison of discrete assignment problems. However, measurement errors and discretization problems often do not have a natural solution. We propose a geometric approach to compare persistence diagrams. We use the Hausdorff distance between persistence diagrams as a metric. We prove that this distance is a metric. We use this distance to compare persistence diagrams. We prove that this distance is a metric. We use this distance to compare persistence diagrams.

1. INTRODUCTION
The assignment problem is among the most famous problems in combinatorial optimization. Given a weighted bipartite graph G with n nodes, we seek for a perfect matching with minimal cost. A common cost function is the minimum of the sum of the edge weights of the matching edges. For $n=2$, we will add the solution in this case of the assignment problem and its cost for the Hausdorff distance. We use this distance to compare persistence diagrams.

2. THE ASSIGNMENT PROBLEM
The assignment problem is among the most famous problems in combinatorial optimization. Given a weighted bipartite graph G with n nodes, we seek for a perfect matching with minimal cost. A common cost function is the minimum of the sum of the edge weights of the matching edges. For $n=2$, we will add the solution in this case of the assignment problem and its cost for the Hausdorff distance. We use this distance to compare persistence diagrams.

3. THE HAUSDORFF DISTANCE
The Hausdorff distance between two sets A and B in a metric space is the maximum of the minimum distances from each point in A to the set B and vice versa. We use this distance to compare persistence diagrams.

4. PERSISTENCE DIAGRAMS
Persistence diagrams are multisets of points in the extended plane. We use the Hausdorff distance to compare persistence diagrams.

5. CONCLUSION
We conclude that the Hausdorff distance is a metric. We use this distance to compare persistence diagrams.

6. REFERENCES
[1] M. Kerber, D. Morozov, and A. Nijssen, "Geometry helps to compare persistence diagrams," *Mathematical Sciences Research Institute Publications*, vol. 104, pp. 192–210, 2005.

Friechet Means for Distributions of Persistence Diagrams

Katharin Turner¹ Yury Mikhlin² Saurabh Mukherjee¹ John Harer¹

ABSTRACT
Given a distribution μ on persistence diagrams and observations $X_1, \dots, X_n \in \mathcal{D}^p$, we introduce an algorithm in this paper that estimates a Fréchet mean from the set of diagrams X_1, \dots, X_n . If the underlying measure μ is a combination of Dirac masses $\mu = \sum_{i=1}^k \alpha_i \delta_{X_i}$, then we prove the algorithm converges to a local minimum and a law of large numbers result for a Fréchet mean computed by the algorithm given observation drawn iid from μ . We illustrate the convergence of an empirical mean computed by the algorithm to a population mean by simulations from Gaussian point clouds.

1. INTRODUCTION
The assignment problem is among the most famous problems in combinatorial optimization. Given a weighted bipartite graph G with n nodes, we seek for a perfect matching with minimal cost. A common cost function is the minimum of the sum of the edge weights of the matching edges. For $n=2$, we will add the solution in this case of the assignment problem and its cost for the Hausdorff distance. We use this distance to compare persistence diagrams.

2. THE ASSIGNMENT PROBLEM
The assignment problem is among the most famous problems in combinatorial optimization. Given a weighted bipartite graph G with n nodes, we seek for a perfect matching with minimal cost. A common cost function is the minimum of the sum of the edge weights of the matching edges. For $n=2$, we will add the solution in this case of the assignment problem and its cost for the Hausdorff distance. We use this distance to compare persistence diagrams.

3. THE HAUSDORFF DISTANCE
The Hausdorff distance between two sets A and B in a metric space is the maximum of the minimum distances from each point in A to the set B and vice versa. We use this distance to compare persistence diagrams.

4. PERSISTENCE DIAGRAMS
Persistence diagrams are multisets of points in the extended plane. We use the Hausdorff distance to compare persistence diagrams.

5. CONCLUSION
We conclude that the Hausdorff distance is a metric. We use this distance to compare persistence diagrams.

6. REFERENCES
[1] K. Turner, Y. Mikhlin, S. Mukherjee, and J. Harer, "Fréchet means for distributions of persistence diagrams," *Mathematical Sciences Research Institute Publications*, vol. 104, pp. 211–230, 2005.

arXiv:1805.08333v2 [stat.ML] 13 Nov 2018

Large Scale computation of Means and Clusters for Persistence Diagrams using Optimal Transport

This Lacombe
Marta Carlier
Yuri Mikhlin
Saurabh Mukherjee

Stéphane Guillot
John Harer
Saurabh Mukherjee

ABSTRACT
Persistence diagrams (PDs) are now routinely used to summarize the underlying topology of complex data. Despite several appealing properties, incorporating PDs in existing pipelines for analyzing big data is still a non-trivial task. In this paper, we propose a novel algorithm for computing the Fréchet mean and clustering of PDs. We prove that this algorithm converges to a local minimum and a law of large numbers result for a Fréchet mean computed by the algorithm given observation drawn iid from μ . We illustrate the convergence of an empirical mean computed by the algorithm to a population mean by simulations from Gaussian point clouds.

1. INTRODUCTION
Topological data analysis (TDA) has been used successfully in a wide array of applications, for instance in medical [Stuker et al., 2015] or material [Hirsh et al., 2016] sciences, computer vision [de Silva et al., 2015] or in ecology [MBA et al., 2015]. The goal of TDA is to capture and extract the topological features of complex data sets. The tools developed in TDA are used to represent complex data sets in a low-dimensional space. This low-dimensional space is used to analyze the data. We use this distance to compare persistence diagrams.

2. THE ASSIGNMENT PROBLEM
The assignment problem is among the most famous problems in combinatorial optimization. Given a weighted bipartite graph G with n nodes, we seek for a perfect matching with minimal cost. A common cost function is the minimum of the sum of the edge weights of the matching edges. For $n=2$, we will add the solution in this case of the assignment problem and its cost for the Hausdorff distance. We use this distance to compare persistence diagrams.

3. THE HAUSDORFF DISTANCE
The Hausdorff distance between two sets A and B in a metric space is the maximum of the minimum distances from each point in A to the set B and vice versa. We use this distance to compare persistence diagrams.

4. PERSISTENCE DIAGRAMS
Persistence diagrams are multisets of points in the extended plane. We use the Hausdorff distance to compare persistence diagrams.

5. CONCLUSION
We conclude that the Hausdorff distance is a metric. We use this distance to compare persistence diagrams.

6. REFERENCES
[1] This Lacombe, M. Carlier, Y. Mikhlin, S. Mukherjee, S. Guillot, J. Harer, and S. Mukherjee, "Large scale computation of means and clusters for persistence diagrams using optimal transport," *Mathematical Sciences Research Institute Publications*, vol. 104, pp. 231–250, 2005.

arXiv:1907.04565v2 [cs.GR] 01 Oct 2019

Progressive Wasserstein Barycenters of Persistence Diagrams

Alain Vial, Joseph Boile, and Jean-Louis Lataf

ABSTRACT
The persistence diagram of a real-valued function on a topological space is a multiset of points in the extended plane. We prove that small perturbations of the function only result in small changes in the diagram. We apply this result to extending the homology of a space to a metric space and to comparing and classifying persistence diagrams.

1. INTRODUCTION
The persistence diagram of a real-valued function on a topological space is a multiset of points in the extended plane. We prove that small perturbations of the function only result in small changes in the diagram. We apply this result to extending the homology of a space to a metric space and to comparing and classifying persistence diagrams.

2. THE PERSISTENCE DIAGRAM
Let X be a topological space and $f: X \rightarrow \mathbb{R}$ a real-valued function. The persistence diagram of f is the set of points in the extended plane that are the birth and death times of the homology of the sublevel sets of f . We prove that small perturbations of the function only result in small changes in the diagram. We apply this result to extending the homology of a space to a metric space and to comparing and classifying persistence diagrams.

3. APPLICATIONS
We apply the stability result to extending the homology of a space to a metric space and to comparing and classifying persistence diagrams.

4. CONCLUSION
We conclude that small perturbations of the function only result in small changes in the diagram. We apply this result to extending the homology of a space to a metric space and to comparing and classifying persistence diagrams.

5. REFERENCES
[1] A. Vial, J. Boile, and J.-L. Lataf, "Progressive Wasserstein barycenters of persistence diagrams," *Mathematical Sciences Research Institute Publications*, vol. 104, pp. 251–270, 2005.

[Cohen-Steiner '05]

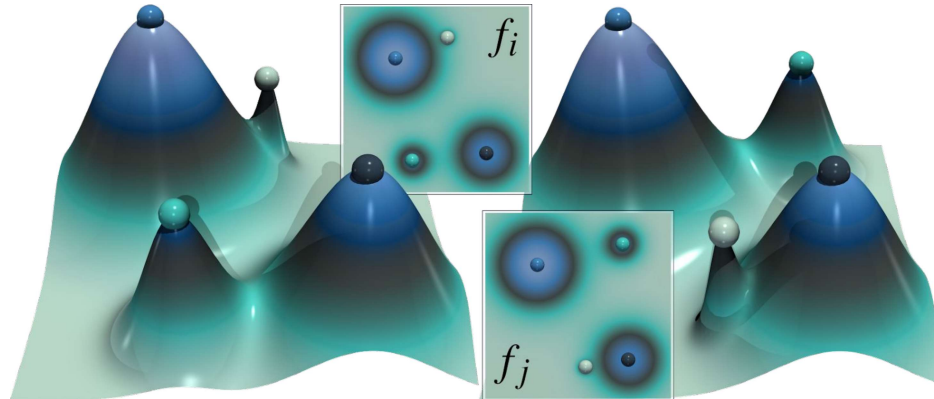
[Kerber '16]

[Turner '14]

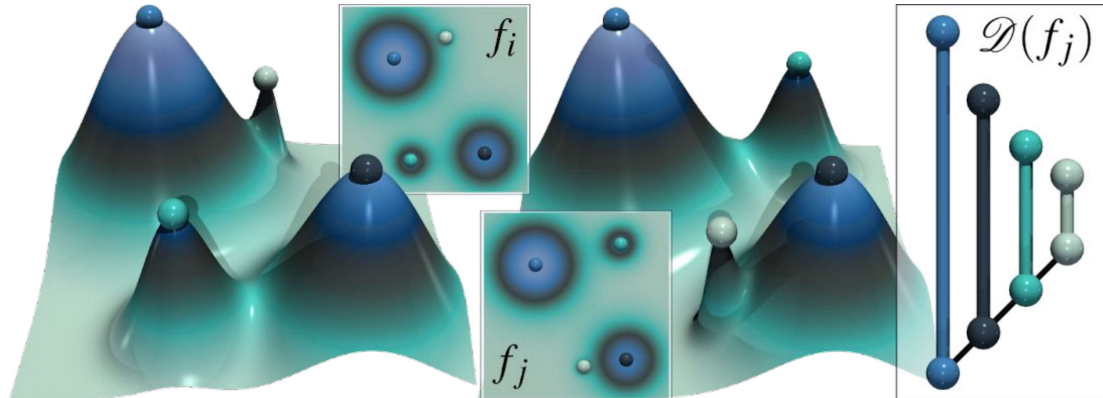
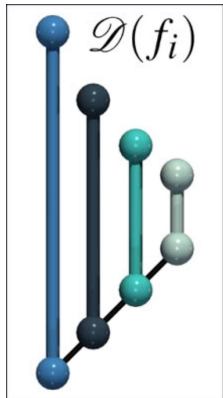
[Lacombe '18]

[Vidal '19]

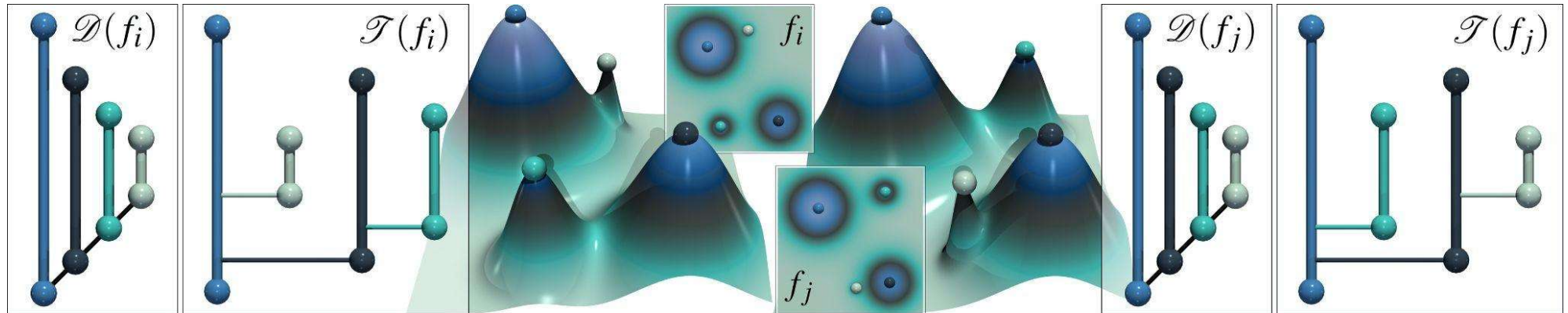
Limitations of Persistence diagrams



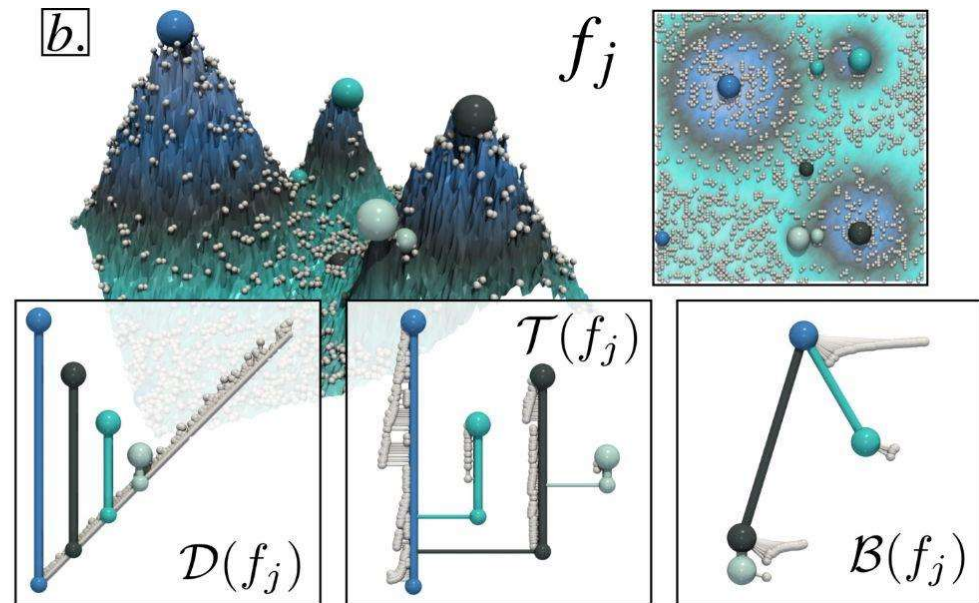
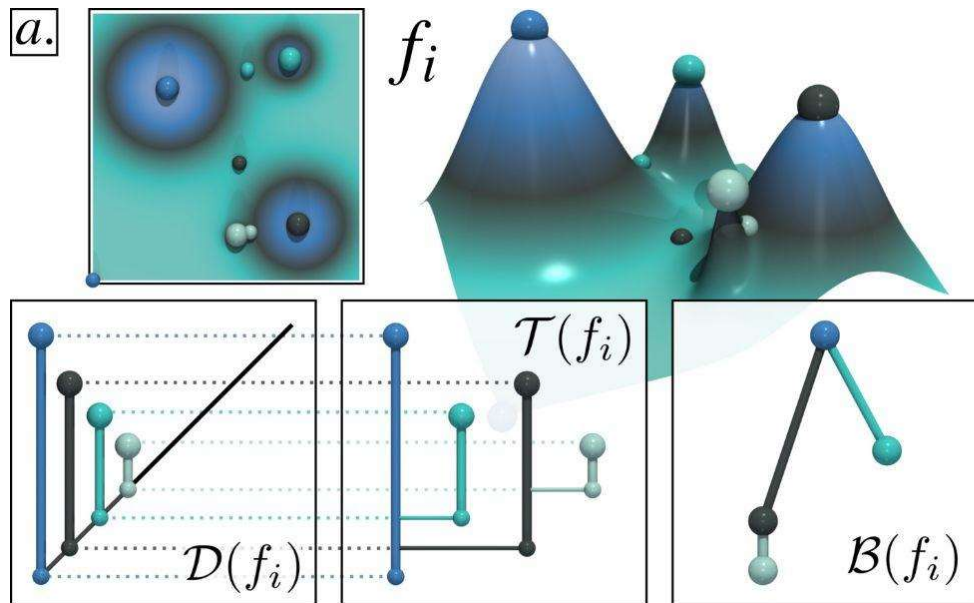
Limitations of Persistence diagrams



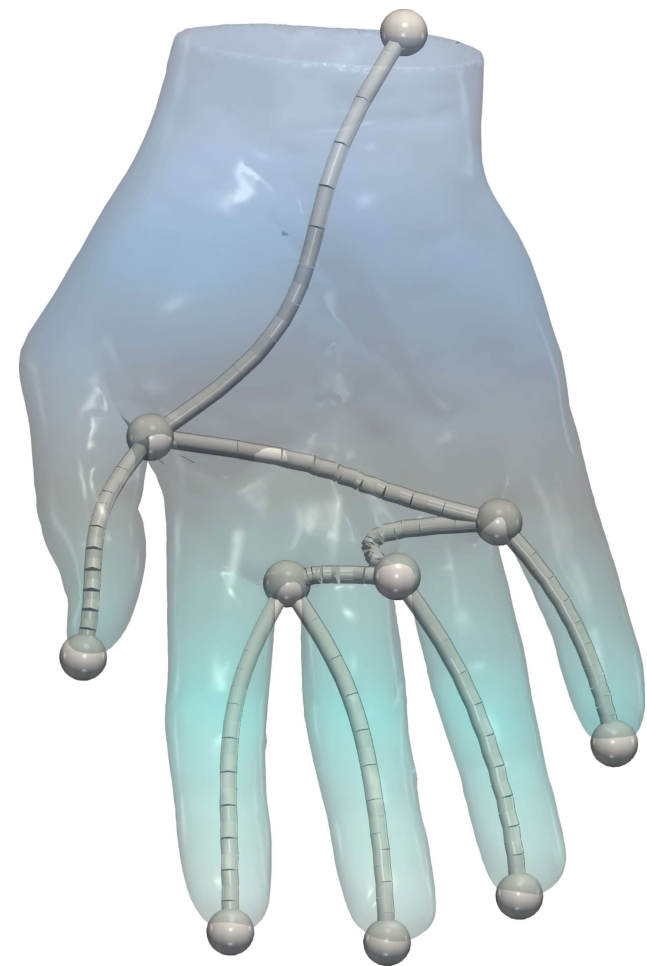
Limitations of Persistence diagrams



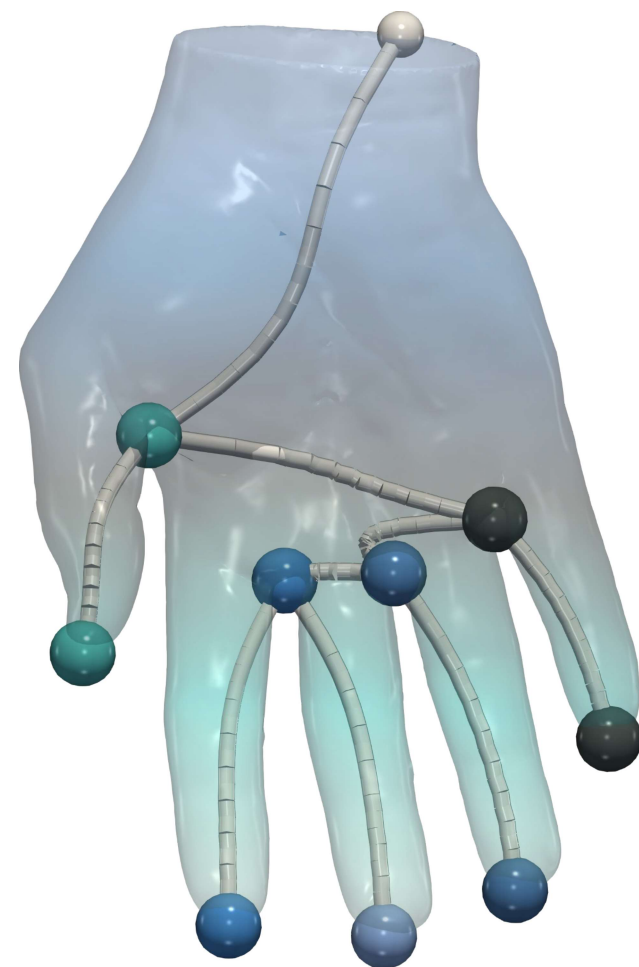
PDs, MTs, BDTs



Branch decomposition tree (BDT)



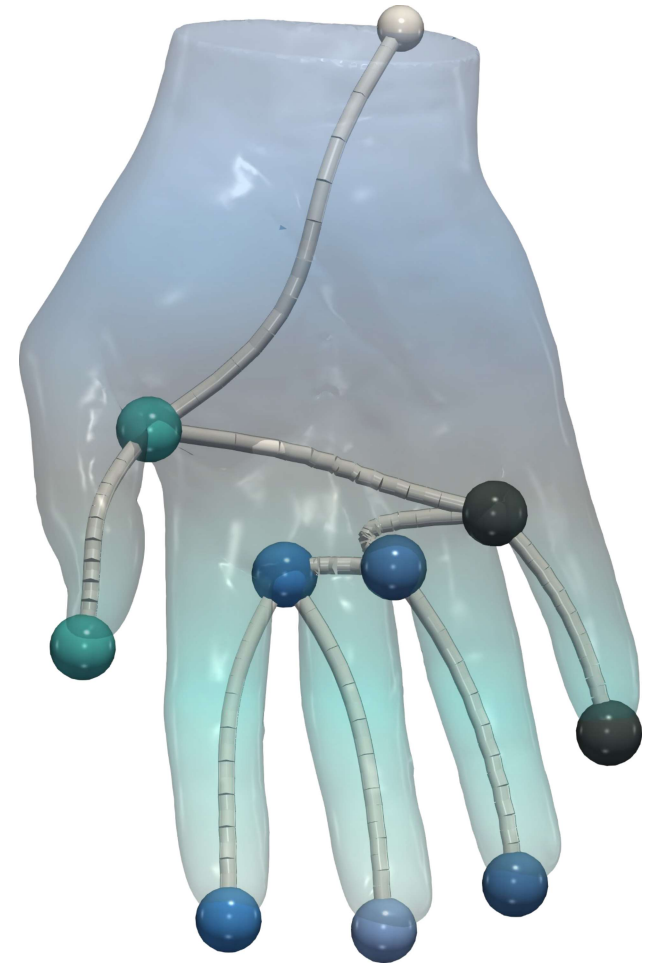
Branch decomposition tree (BDT)



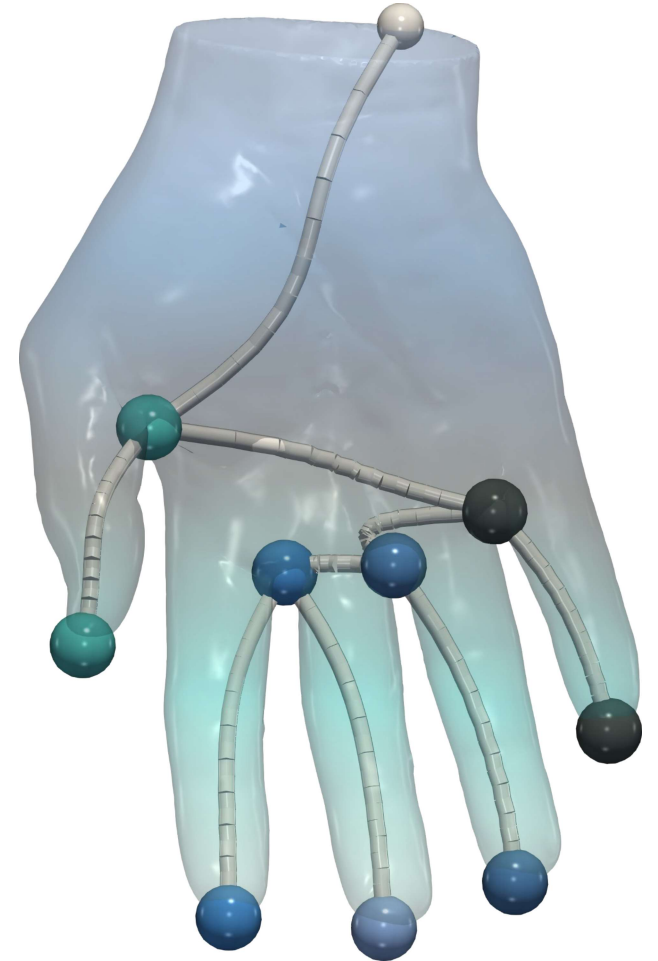
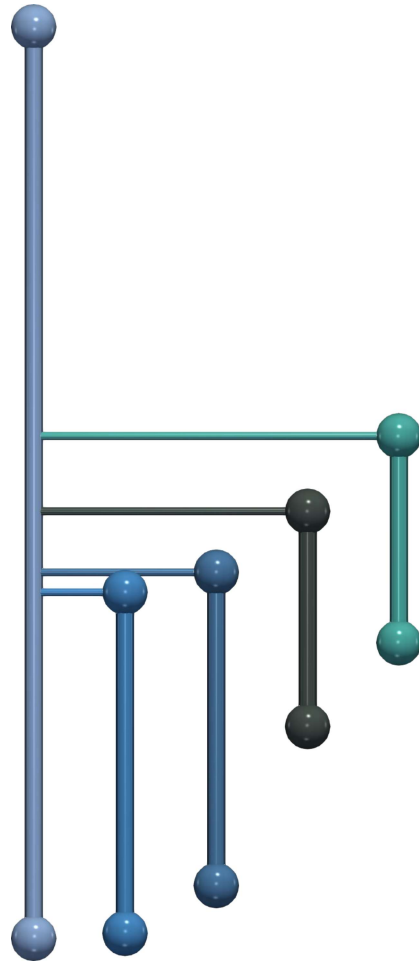
Branch decomposition tree (BDT)

- **Persistent branch**

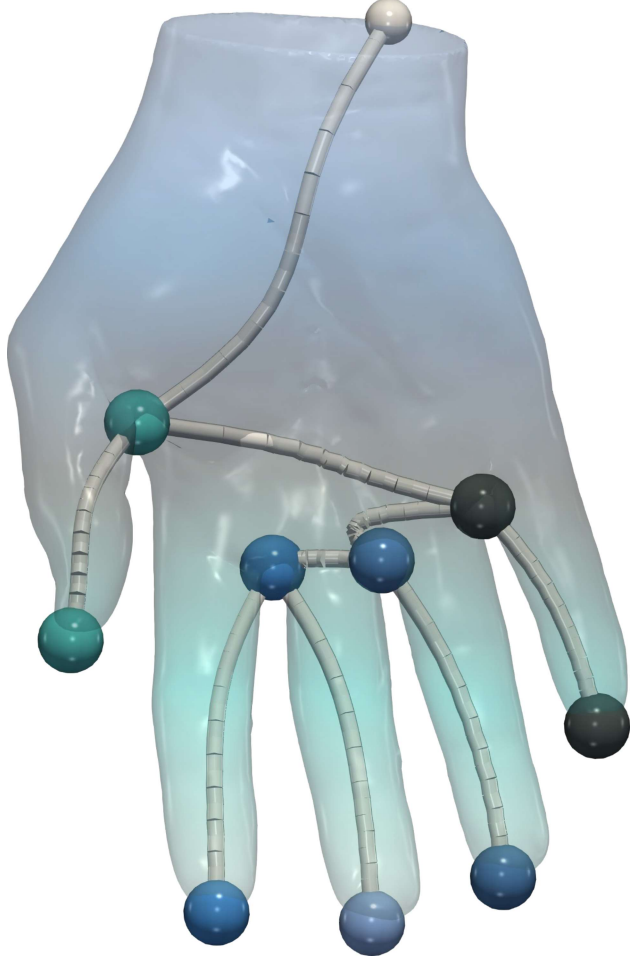
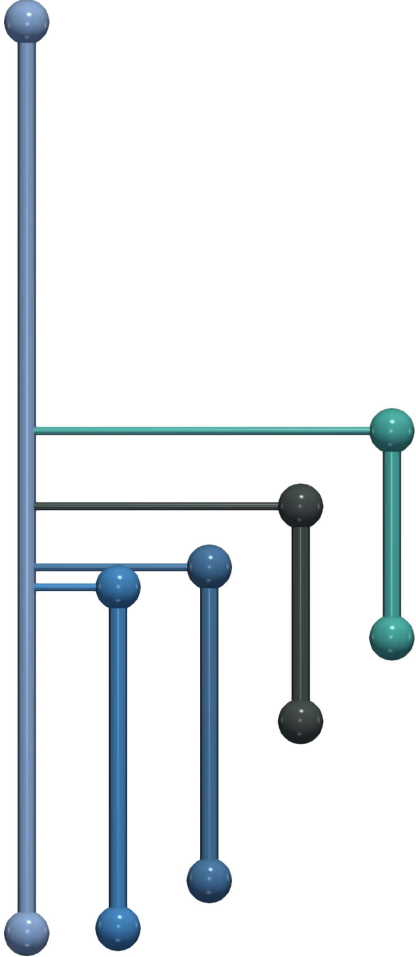
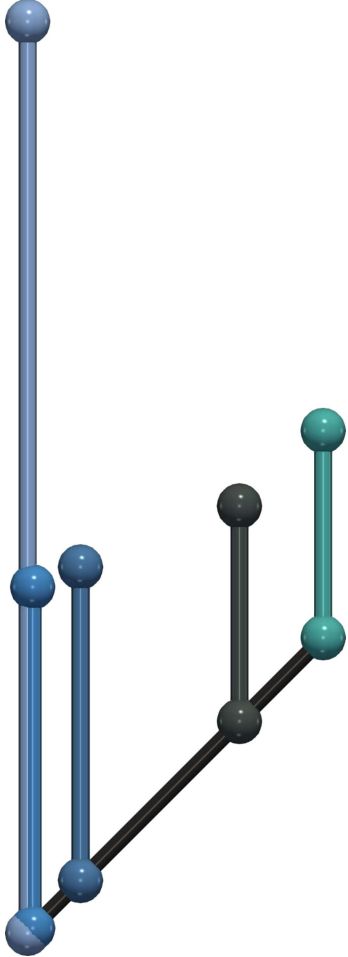
- Monotonic path
- From a minimum
- Up to its persistence-paired saddle



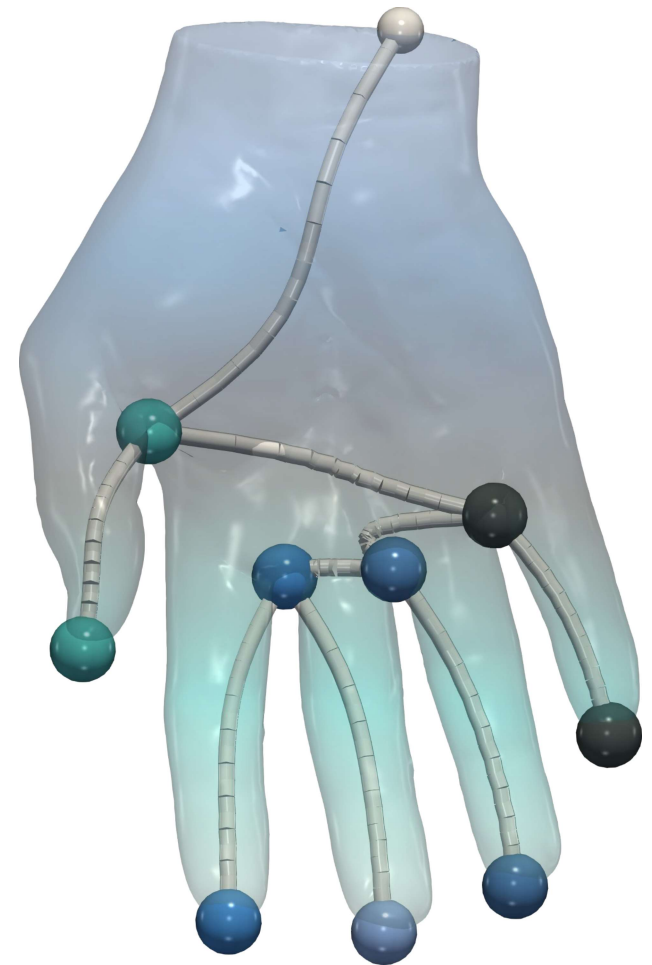
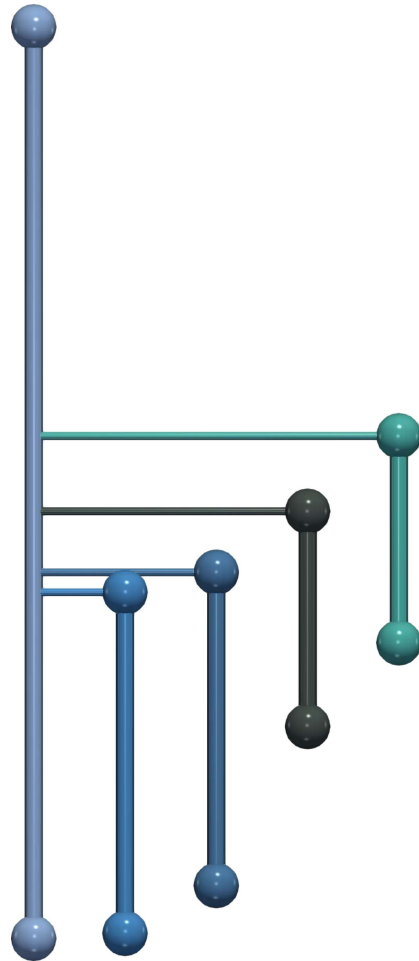
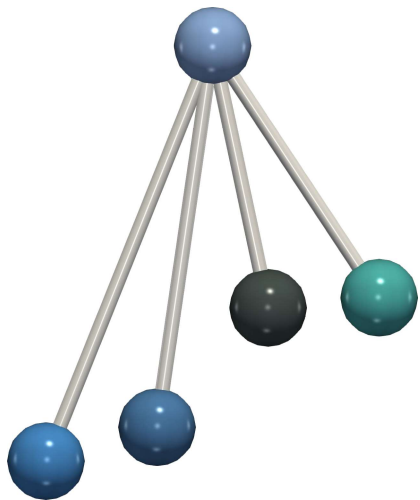
Branch decomposition tree (BDT)



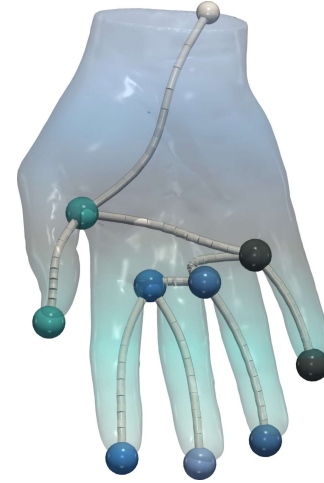
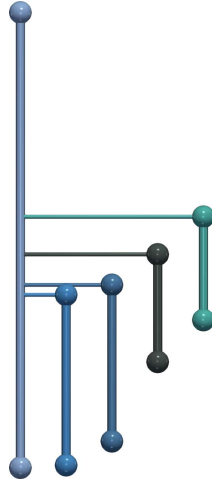
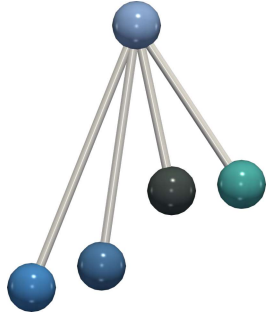
Branch decomposition tree (BDT)



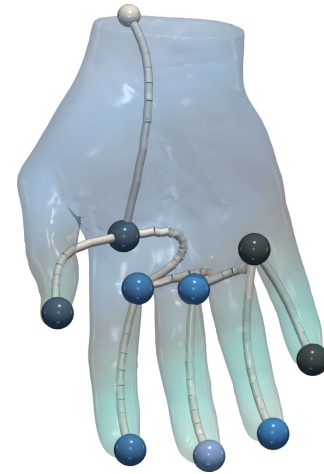
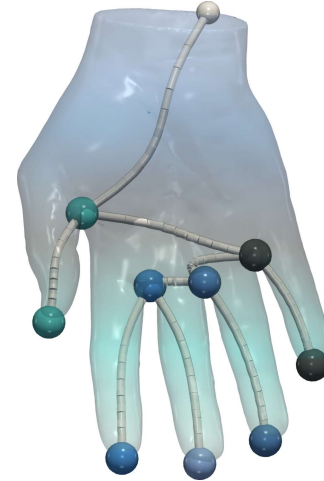
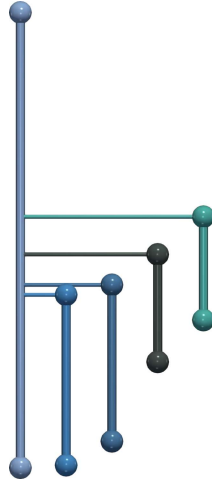
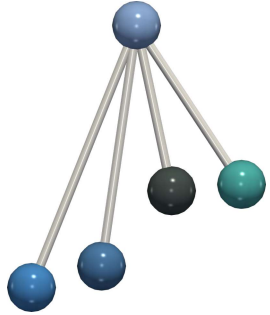
Branch decomposition tree (BDT)



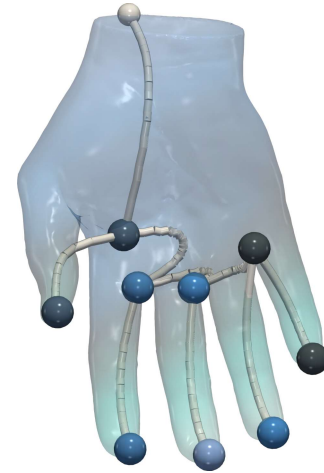
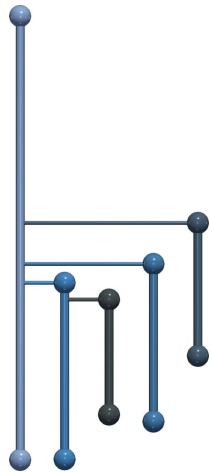
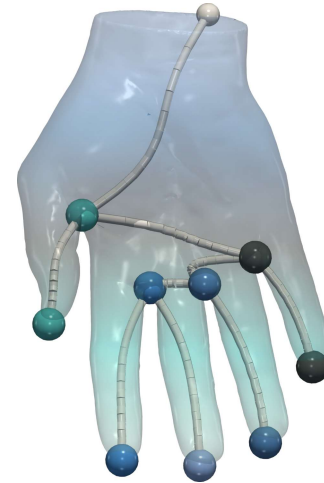
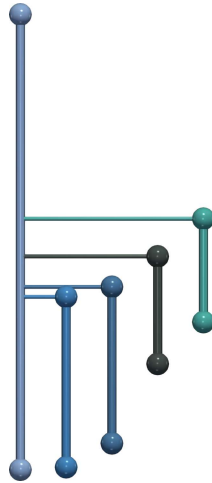
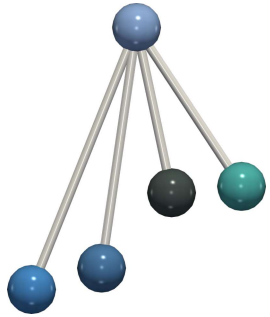
BDTs in practice



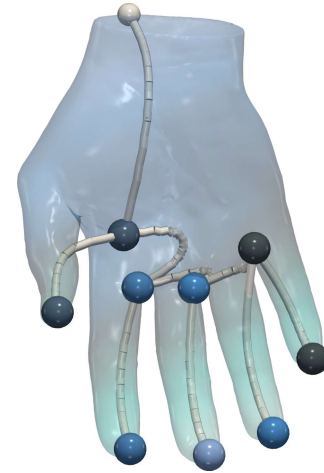
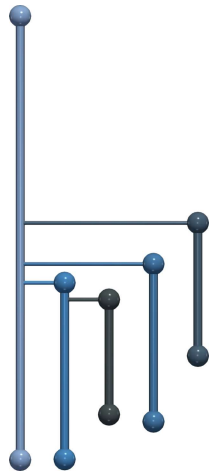
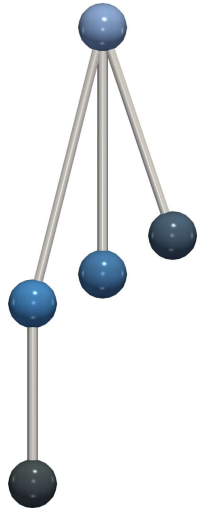
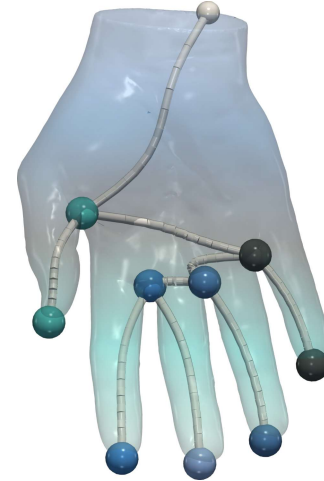
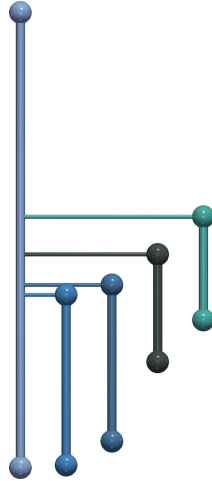
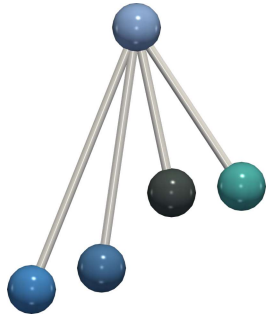
BDTs in practice



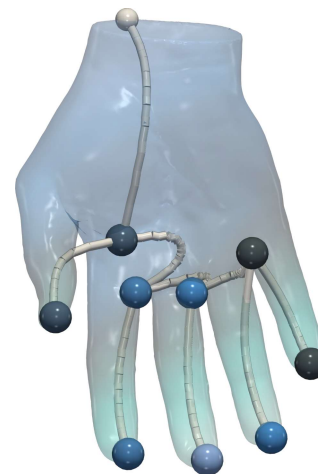
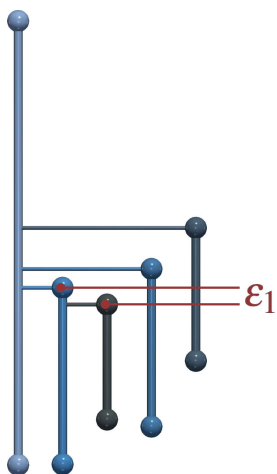
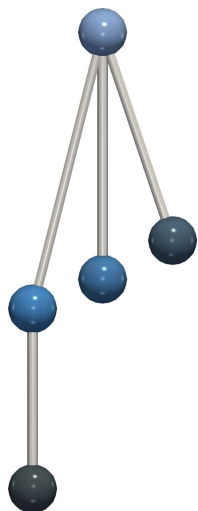
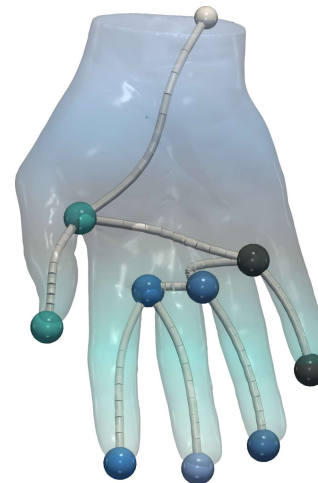
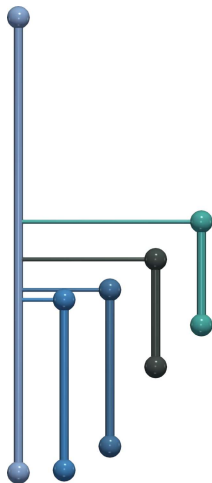
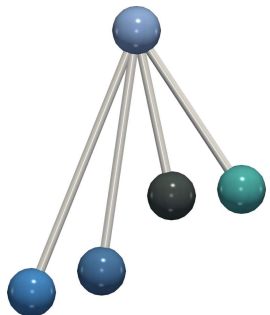
BDTs in practice



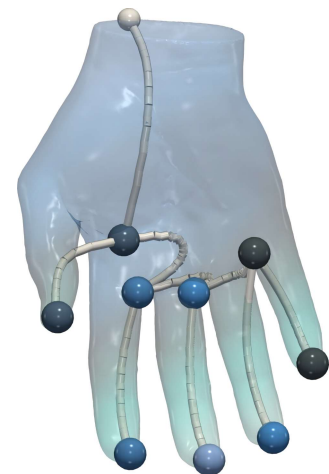
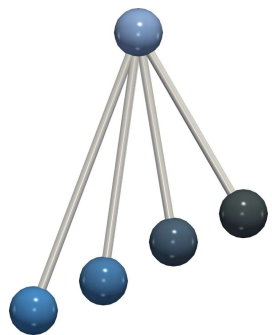
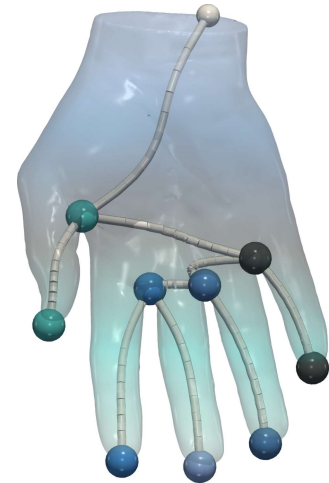
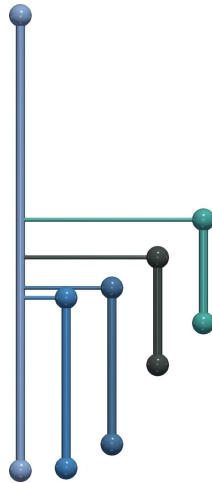
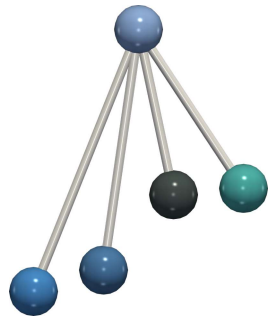
BDTs in practice



BDTs in practice



BDTs in practice



Merge trees for ensemble analysis

Recent works

arXiv:2007.08511v1 [cs.CG] 18 Jul 2022

Edit Distance between Merge Trees

Rajagwanda Sridharanurthy, Student Member, IEEE, Taha Bin Masood, Ashiya Karanaheladan, and Vijay Natarajan, Member, IEEE

Abstract Topological structures such as the merge tree provide an efficient and concise representation of scalar fields. The notions of edit distance and tree-to-tree alignment of merge trees are a natural way to compare and contrast the topological structures of scalar fields. This paper introduces a novel approach for the edit distance between merge trees. The proposed approach is based on the edit distance between merge trees. The proposed approach is based on the edit distance between merge trees. The proposed approach is based on the edit distance between merge trees.

Index Terms Merge trees, scalar field, distance measure, persistence, edit distance.

1 INTRODUCTION

The study of the behavior of physical quantities over time helps in understanding underlying scientific processes. Physical quantities are often measured using single data points or sampled via trajectories. In either case, they are often modeled as scalar functions over time. A scalar function being measured in a noisy environment, a good overview can be gained by first factoring the underlying data of such noisy function and extracting its essential features. The analysis often requires a way over a large subset of the domain or range of the function over which the features of interest may be contained within a small region. These features are captured when one uses a merge tree to represent the function. This idea has been used for feature detection and visualization. Topological structures such as the merge tree [1] show in Figure 1 provide a succinct representation of scalar functions. Topological structures such as the merge tree [1] show in Figure 1 provide a succinct representation of scalar functions. Topological structures such as the merge tree [1] show in Figure 1 provide a succinct representation of scalar functions.

1.1 Related Work

Amongst classical distance, RMS Distance, Chebyshev distance, and other norms such as L_1 , L_2 , L_∞ , etc., use the metric for comparing scalar functions. For example, distance or similarity measure between scalar functions is essential for analyzing persistence. However, edit distance measure of distance between merge trees is also used to compare two scalar functions. The merge trees of distance between merge trees is also used to compare two scalar functions. The merge trees of distance between merge trees is also used to compare two scalar functions.

References

- [1] R. Ghilardi, M. Nanni, and G. Traverso, "Topological data analysis: From persistence to classification," *Handbook of topological data analysis*, pp. 1-50, 2019.
- [2] J. Zomorodi, "Topological data analysis: A survey," *arXiv preprint arXiv:1906.00013*, 2019.
- [3] M. Nanni, R. Ghilardi, and G. Traverso, "Topological data analysis: A survey," *arXiv preprint arXiv:1906.00013*, 2019.

© 2019 IEEE. This is the author's version of the article that has been published in IEEE Transactions on Visualization and Computer Graphics. This final version of the article is available at: 10.1109/TVCG.2019.2934242

A Structural Average of Labeled Merge Trees for Uncertainty Visualization

Lin Yao, Yuesi Wang, Shiqiang Mo, Han Ge, and Jun Wang

Fig. 1. Computing a 1-center tree as a structural average of an ensemble of six labeled merge trees (partial alignment) for the set of all nodes in the ensemble. The nodes are colored by their label. The nodes are colored by their label. The nodes are colored by their label.

Abstract Topological structures such as merge trees, contour trees, and tree graphs are commonly used to describe scalar fields. In this paper, we propose a structural average of an ensemble of labeled merge trees (partial alignment) for the set of all nodes in the ensemble. The nodes are colored by their label. The nodes are colored by their label. The nodes are colored by their label.

Index Terms Topological data analysis, uncertainty visualization, merge trees.

1 INTRODUCTION

In recent years, topological data analysis (TDA) has become a popular topic in data science. TDA provides a way to analyze data by looking at its shape and structure. TDA provides a way to analyze data by looking at its shape and structure. TDA provides a way to analyze data by looking at its shape and structure.

References

- [1] M. Nanni, R. Ghilardi, and G. Traverso, "Topological data analysis: A survey," *arXiv preprint arXiv:1906.00013*, 2019.
- [2] J. Zomorodi, "Topological data analysis: A survey," *arXiv preprint arXiv:1906.00013*, 2019.
- [3] M. Nanni, R. Ghilardi, and G. Traverso, "Topological data analysis: A survey," *arXiv preprint arXiv:1906.00013*, 2019.

arXiv:2107.07789v1 [cs.GR] 16 Jul 2021

Wasserstein Distances, Geodesics and Barycenters of Merge Trees

Mathieu Pont, Jörn Voll, Julia Delort and Julien Terno

Fig. 1. The merge tree of a scalar field (left) is the natural structure that encodes the topology of the field. The merge tree of a scalar field (left) is the natural structure that encodes the topology of the field. The merge tree of a scalar field (left) is the natural structure that encodes the topology of the field.

Abstract The paper presents a natural topological framework for the definition of distances, geodesics and barycenters of merge trees. The paper presents a natural topological framework for the definition of distances, geodesics and barycenters of merge trees. The paper presents a natural topological framework for the definition of distances, geodesics and barycenters of merge trees.

Index Terms Topological data analysis, merge trees, scalar data, barycenter data.

1 INTRODUCTION

Topological data analysis (TDA) has become a popular topic in data science. TDA provides a way to analyze data by looking at its shape and structure. TDA provides a way to analyze data by looking at its shape and structure. TDA provides a way to analyze data by looking at its shape and structure.

References

- [1] M. Nanni, R. Ghilardi, and G. Traverso, "Topological data analysis: A survey," *arXiv preprint arXiv:1906.00013*, 2019.
- [2] J. Zomorodi, "Topological data analysis: A survey," *arXiv preprint arXiv:1906.00013*, 2019.
- [3] M. Nanni, R. Ghilardi, and G. Traverso, "Topological data analysis: A survey," *arXiv preprint arXiv:1906.00013*, 2019.

[Sridharanurthy '18]

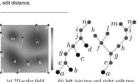
[Yan '19]

[Pont '21]

Merge trees for ensemble analysis

Recent works

EDIT DISTANCE BETWEEN MERGE TREES
Raghavendra Sridharanurthy, Student Member, IEEE, Taha Bin Masood, Ashiya Karanahakshani, and Vijay Natarajan, Member, IEEE




Abstract Topological structures such as the merge tree provide an efficient and compact representation of scalar fields. The Hoffman distance and its extension to merge trees provide a way to compare the topology of scalar fields and their merge trees. This paper introduces a novel approach for comparing merge trees. The proposed distance is based on the Hoffman distance between the merge trees of the two scalar fields. The proposed distance is based on the Hoffman distance between the merge trees of the two scalar fields. The proposed distance is based on the Hoffman distance between the merge trees of the two scalar fields.

Index Terms Merge trees, scalar field, distance measure, persistence, edit distance

1 INTRODUCTION
The study of the behavior of physical quantities over time helps in understanding underlying scientific processes. Physical quantities are often measured using single-point observations or averaged observations. In either case, they are often modeled as scalar functions over time. A scalar function being measured in a scalar field. A scalar function being measured in a scalar field. A scalar function being measured in a scalar field.

A Structural Average of Labeled Merge Trees for Uncertainty Visualization
Lin Yao, Yuesi Wang, Shiqiang Mo, Han Ge, and Jian Wang




Abstract Physical phenomena in a space and a parameter space are frequently modeled using scalar fields. In scalar field topology, the merge tree is a natural way to capture the topological information of a scalar field. The merge tree is a natural way to capture the topological information of a scalar field. The merge tree is a natural way to capture the topological information of a scalar field.

Index Terms Topological analysis, uncertainty visualization, merge trees

1 INTRODUCTION
In recent days, machine and visualization, topological structure has been used to analyze and understand the data. In recent days, machine and visualization, topological structure has been used to analyze and understand the data. In recent days, machine and visualization, topological structure has been used to analyze and understand the data.

Wasserstein Distances, Geodesics and Barycenters of Merge Trees
Mathew Poon, James Vass, Julia Deane and Julian Terry

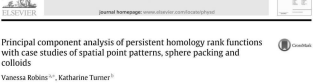


Abstract Merge trees are a natural way to capture the topological information of a scalar field. The merge tree is a natural way to capture the topological information of a scalar field. The merge tree is a natural way to capture the topological information of a scalar field.

Index Terms Topological data analysis, merge trees, Wasserstein distance

1 INTRODUCTION
Merge trees are a natural way to capture the topological information of a scalar field. The merge tree is a natural way to capture the topological information of a scalar field. The merge tree is a natural way to capture the topological information of a scalar field.

Principal component analysis of persistent homology rank functions with case studies of spatial point patterns, sphere packing and colloids
Vanessa Robins¹, Katharine Turner¹



Abstract Persistent homology rank functions are a natural way to capture the topological information of a scalar field. The rank function is a natural way to capture the topological information of a scalar field. The rank function is a natural way to capture the topological information of a scalar field.

Index Terms Persistent homology, rank functions, spatial point patterns, sphere packing, colloids

1 INTRODUCTION
Persistent homology rank functions are a natural way to capture the topological information of a scalar field. The rank function is a natural way to capture the topological information of a scalar field. The rank function is a natural way to capture the topological information of a scalar field.

[Sridharanurthy '18]

[Yan '19]

[Pont '21]

[Robins '16]

Merge trees for ensemble analysis

Recent works

arXiv:2007.08511v1 [cs.CG] 18 Jul 2022

Edit Distance between Merge Trees

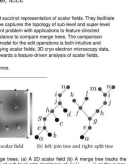
Raghavendra Sridharamurthy, Student Member, IEEE, Taha Bin Masood, Ashiya Karanahakshasa, and Vijay Natarajan, Member, IEEE

Abstract Topological structures such as the merge tree provide an efficient and compact representation of scalar fields. The Hoffman affine distance and its extension to merge trees (HMT) is a merge tree distance that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field. In this paper, we propose a new distance between merge trees, the edit distance, which is sensitive to topological changes and is also sensitive to changes in the underlying scalar field. The edit distance is a distance between merge trees that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field. The edit distance is a distance between merge trees that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field.

Index Terms Merge trees, scalar field, distance measures, persistence, distance

1 INTRODUCTION

The study of the behavior of physical quantities over time helps in understanding underlying scientific processes. Physical quantities are often measured using time-series data or sampled at discrete time intervals. In either case, they are often modeled as scalar functions over time. In this paper, we propose a new distance between merge trees, the edit distance, which is sensitive to topological changes and is also sensitive to changes in the underlying scalar field. The edit distance is a distance between merge trees that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field.



2 RELATED WORK

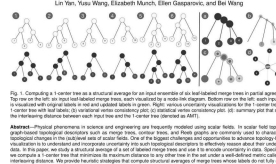
Distance measures between scalar fields are used to quantify the difference between two scalar fields. The Hoffman affine distance and its extension to merge trees (HMT) are two such distance measures. The edit distance is a distance between merge trees that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field.

2020 IEEE. This is the author's version of an article that has been published in IEEE Transactions on Visualization and Computer Graphics. This final version of the article is available at: <https://doi.org/10.1109/TVCG.2020.3042422>

A Structural Average of Labeled Merge Trees for Uncertainty Visualization

Lin Yao, Yuxi Wang, Shiqiang Mo, Yuesi Ge, and Yanyan Tang

Abstract Topological structures such as the merge tree provide an efficient and compact representation of scalar fields. The Hoffman affine distance and its extension to merge trees (HMT) is a merge tree distance that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field. In this paper, we propose a new distance between merge trees, the structural average, which is sensitive to topological changes and is also sensitive to changes in the underlying scalar field. The structural average is a distance between merge trees that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field.



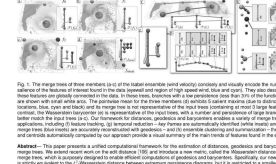
1 INTRODUCTION

Topological structures such as the merge tree provide an efficient and compact representation of scalar fields. The Hoffman affine distance and its extension to merge trees (HMT) is a merge tree distance that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field. In this paper, we propose a new distance between merge trees, the structural average, which is sensitive to topological changes and is also sensitive to changes in the underlying scalar field.

Wasserstein Distances, Geodesics and Barycenters of Merge Trees

Mehmet Polat, James Vass, Julia Deiters and Julian Terry

Abstract Topological structures such as the merge tree provide an efficient and compact representation of scalar fields. The Hoffman affine distance and its extension to merge trees (HMT) is a merge tree distance that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field. In this paper, we propose a new distance between merge trees, the Wasserstein distance, which is sensitive to topological changes and is also sensitive to changes in the underlying scalar field. The Wasserstein distance is a distance between merge trees that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field.



1 INTRODUCTION

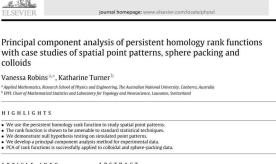
Topological structures such as the merge tree provide an efficient and compact representation of scalar fields. The Hoffman affine distance and its extension to merge trees (HMT) is a merge tree distance that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field. In this paper, we propose a new distance between merge trees, the Wasserstein distance, which is sensitive to topological changes and is also sensitive to changes in the underlying scalar field.

Physics D

Principal component analysis of persistent homology rank functions with case studies of spatial point patterns, sphere packing and colloids

Vanesa Robles¹, Katharine Turner¹

Abstract Topological structures such as the merge tree provide an efficient and compact representation of scalar fields. The Hoffman affine distance and its extension to merge trees (HMT) is a merge tree distance that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field. In this paper, we propose a new distance between merge trees, the principal component analysis, which is sensitive to topological changes and is also sensitive to changes in the underlying scalar field. The principal component analysis is a distance between merge trees that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field.



1 INTRODUCTION

Topological structures such as the merge tree provide an efficient and compact representation of scalar fields. The Hoffman affine distance and its extension to merge trees (HMT) is a merge tree distance that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field. In this paper, we propose a new distance between merge trees, the principal component analysis, which is sensitive to topological changes and is also sensitive to changes in the underlying scalar field.

Principal Geodesic Analysis for Probability Measures under the Optimal Transport Metric

Viktor Gajda, Graduate School of Informatics, Kyoto University
v.gajda@informatics.kyoto-u.ac.jp

Meow Choi, Graduate School of Informatics, Kyoto University
meowchoi@informatics.kyoto-u.ac.jp

Abstract Topological structures such as the merge tree provide an efficient and compact representation of scalar fields. The Hoffman affine distance and its extension to merge trees (HMT) is a merge tree distance that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field. In this paper, we propose a new distance between merge trees, the principal geodesic analysis, which is sensitive to topological changes and is also sensitive to changes in the underlying scalar field. The principal geodesic analysis is a distance between merge trees that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field.

1 INTRODUCTION

Topological structures such as the merge tree provide an efficient and compact representation of scalar fields. The Hoffman affine distance and its extension to merge trees (HMT) is a merge tree distance that is sensitive to topological changes and is also sensitive to changes in the underlying scalar field. In this paper, we propose a new distance between merge trees, the principal geodesic analysis, which is sensitive to topological changes and is also sensitive to changes in the underlying scalar field.

[Sridharamurthy '18]

[Yan '19]

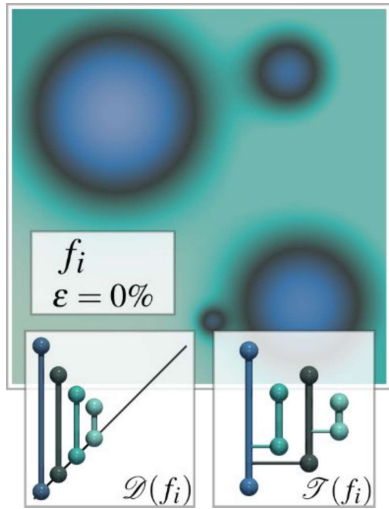
[Pont '21]

[Robins '16]

[Seguy '15]

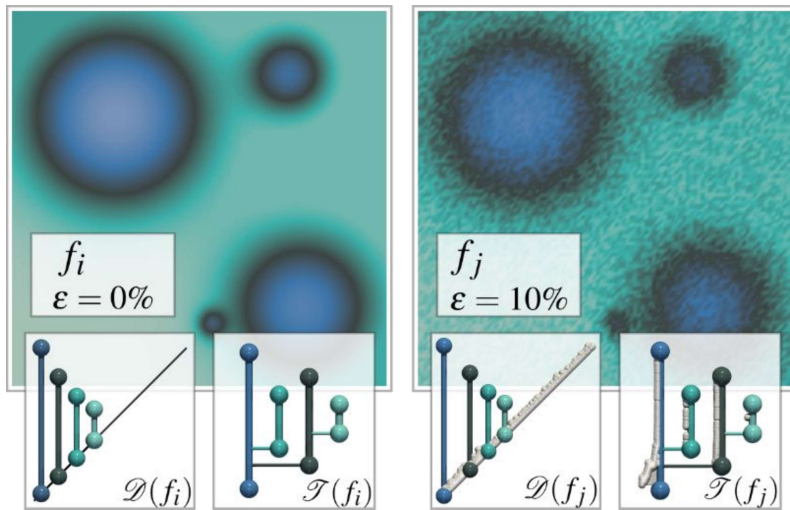
Evaluation

- Stability



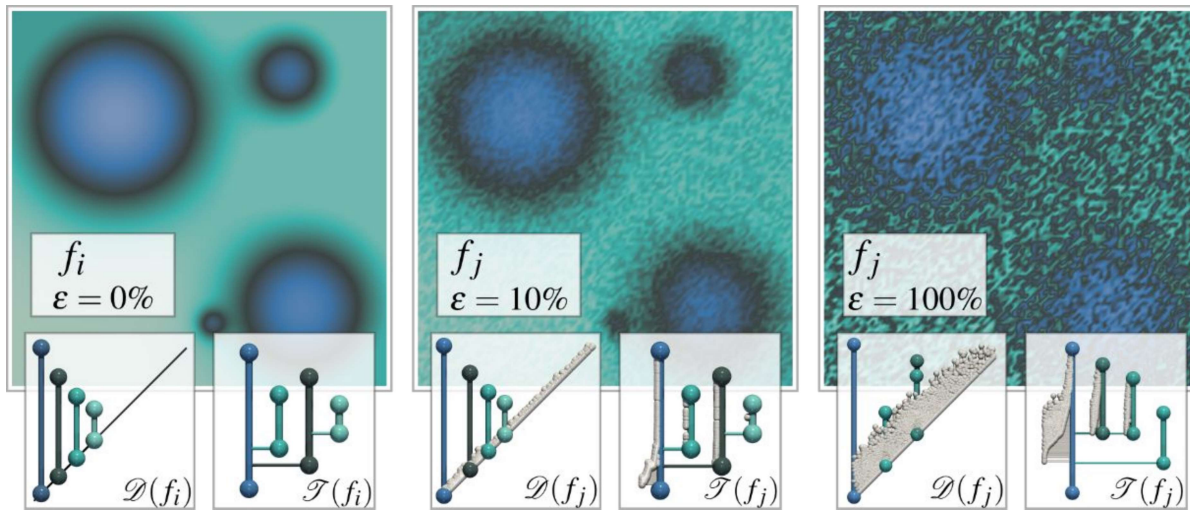
Evaluation

- Stability



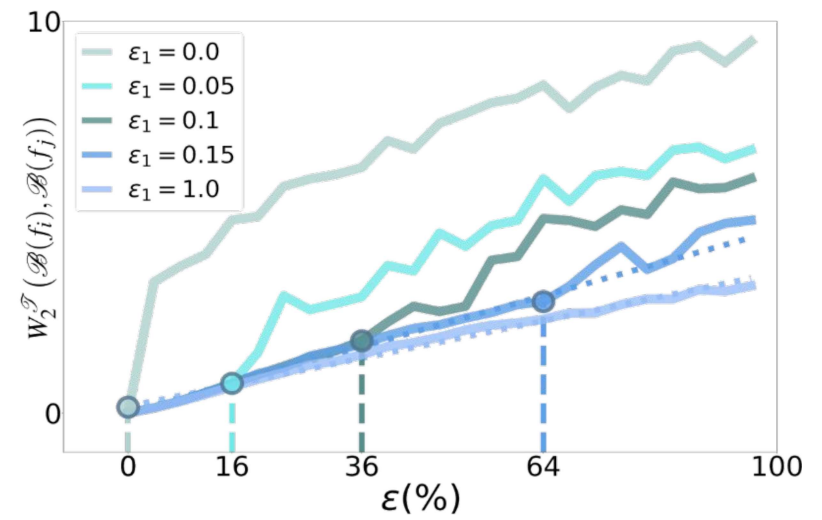
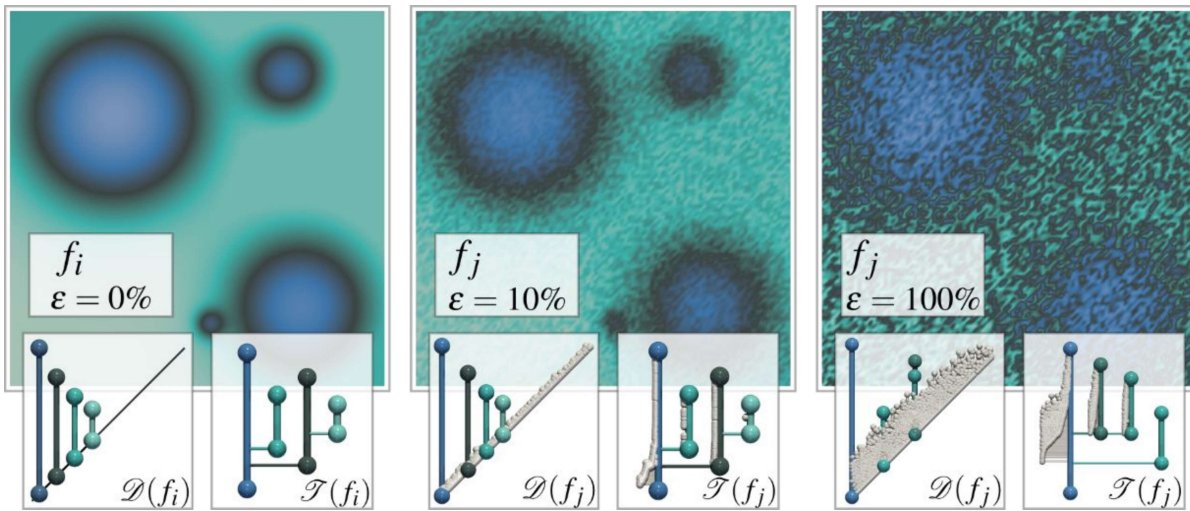
Evaluation

- Stability



Evaluation

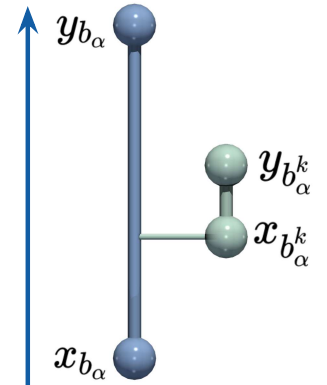
- Stability



Local normalization

- Nesting condition

- $[x_{b_\alpha}^k, y_{b_\alpha}^k] \subseteq [x_{b_\alpha}, y_{b_\alpha}]$



Local normalization

- **Nesting condition**

- $[x_{b_\alpha}^k, y_{b_\alpha}^k] \subseteq [x_{b_\alpha}, y_{b_\alpha}]$

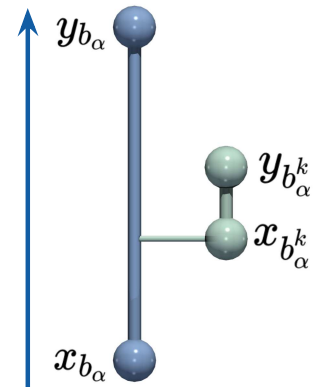
- **Strategy**

- Local normalization
- Distance/Interpolation
- Normalization reversal

$$\mathcal{N}(b_i^k) = (\mathcal{N}_x(b_i^k), \mathcal{N}_y(b_i^k))$$

$$\mathcal{N}_x(b_i^k) = (x_{b_i^k} - x_{b_i}) / (y_{b_i} - x_{b_i})$$

$$\mathcal{N}_y(b_i^k) = (y_{b_i^k} - x_{b_i}) / (y_{b_i} - x_{b_i})$$



Local normalization

- **Nesting condition**

- $[x_{b_\alpha}^k, y_{b_\alpha}^k] \subseteq [x_{b_\alpha}, y_{b_\alpha}]$

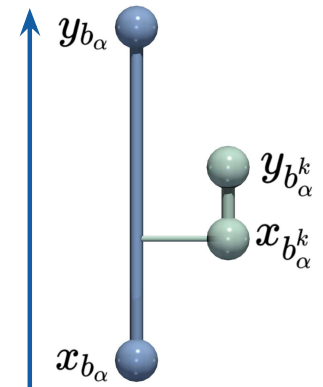
- **Strategy**

- Local normalization
- Distance/Interpolation
- Normalization reversal

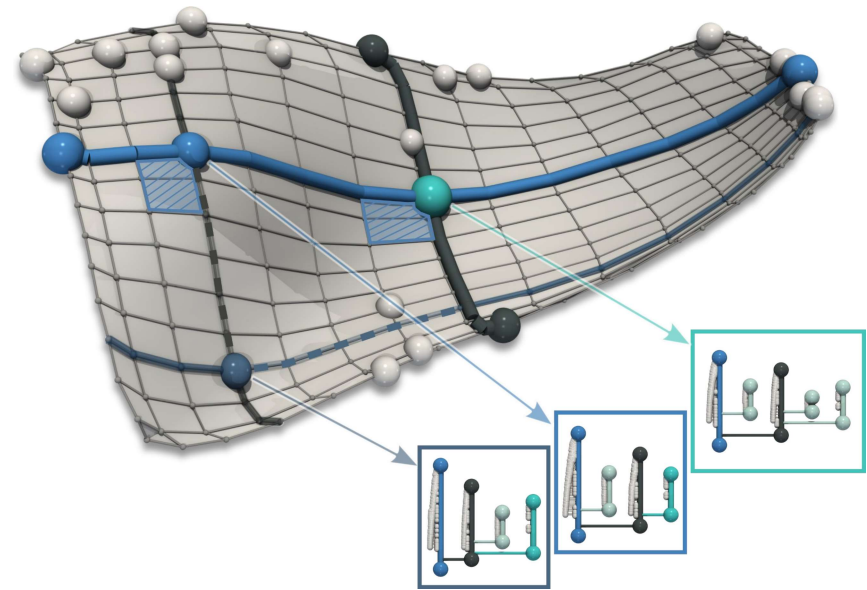
- **BDT pre-process**

- Relative persistence $< \varepsilon_3$
- Until ε_2 of their parents
- Default: $\varepsilon_1 = 0.05$, $\varepsilon_2 = 0.95$, $\varepsilon_3 = 0.9$

$$\mathcal{N}(b_i^k) = (\mathcal{N}_x(b_i^k), \mathcal{N}_y(b_i^k))$$
$$\mathcal{N}_x(b_i^k) = (x_{b_i^k} - x_{b_i}) / (y_{b_i} - x_{b_i})$$
$$\mathcal{N}_y(b_i^k) = (y_{b_i^k} - x_{b_i}) / (y_{b_i} - x_{b_i})$$

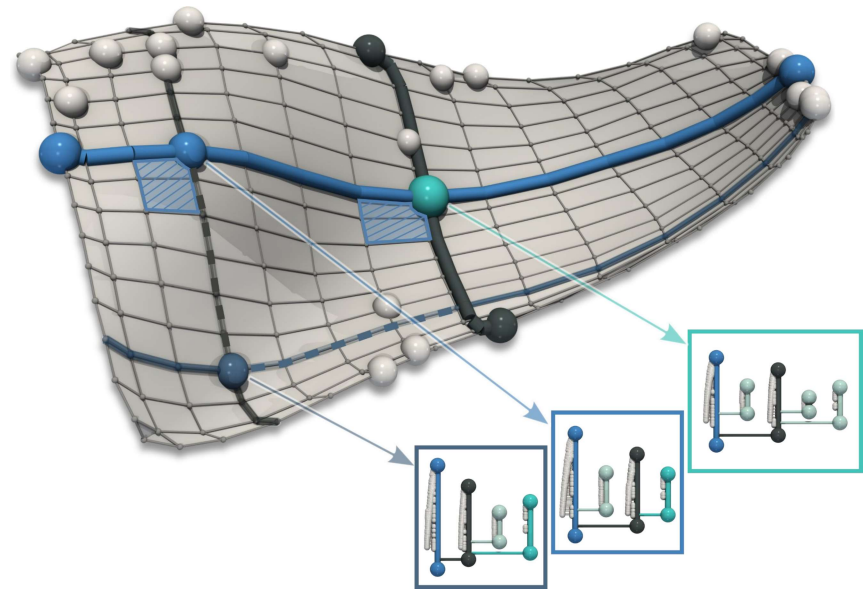


From PCA to MT-PGA



From PCA to MT-PGA

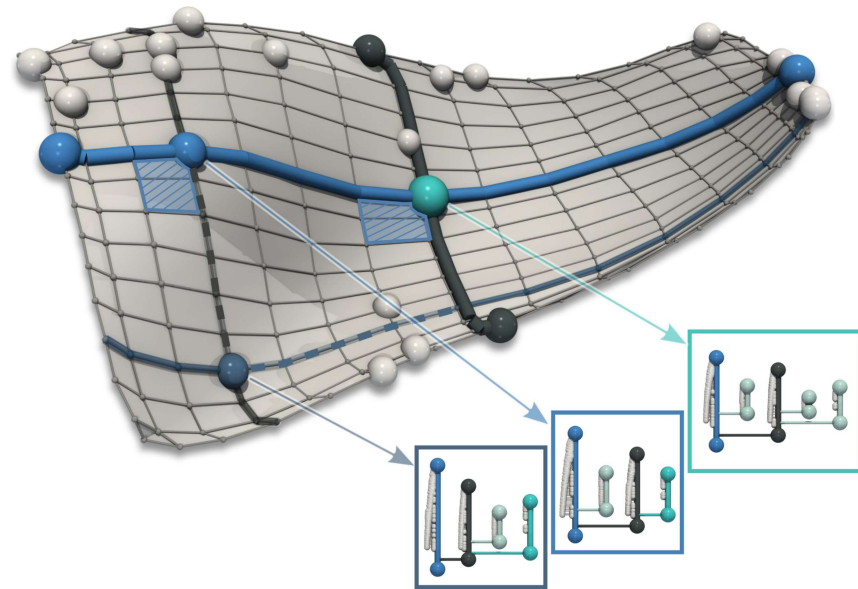
- Required low-level tools



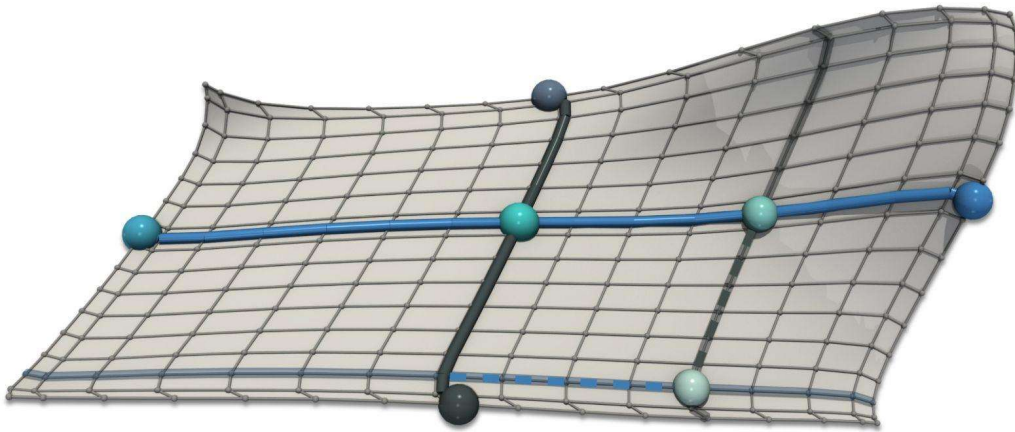
From PCA to MT-PGA

- **Required low-level tools**

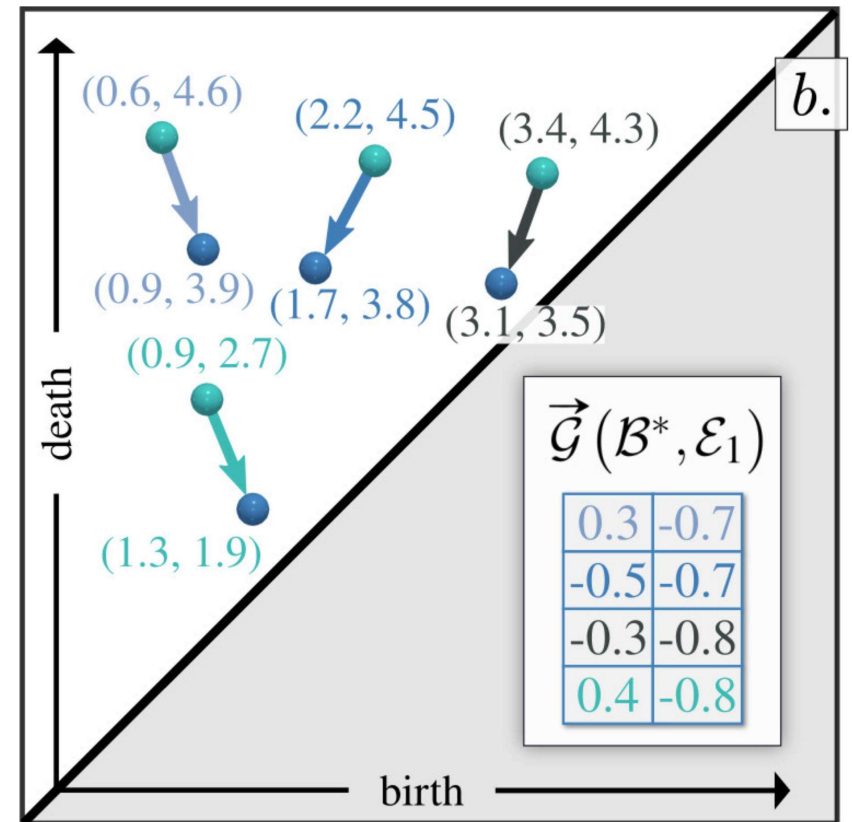
- Geodesic
- Orthogonal geodesics
- Collinear geodesics
- Geodesic axis
- Axis projection
- Orthogonal axes
- Axis translation
- Orthogonal basis



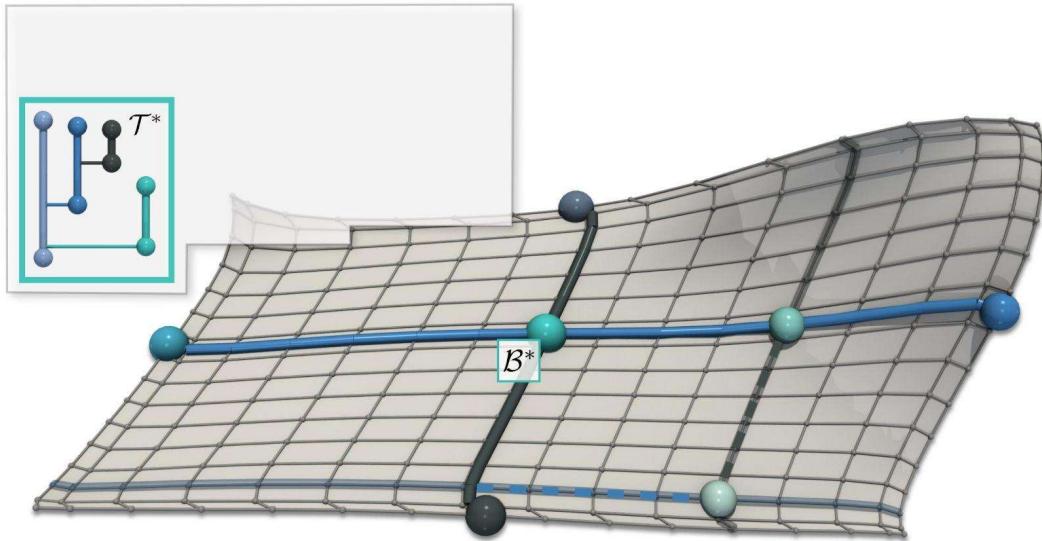
Geodesics



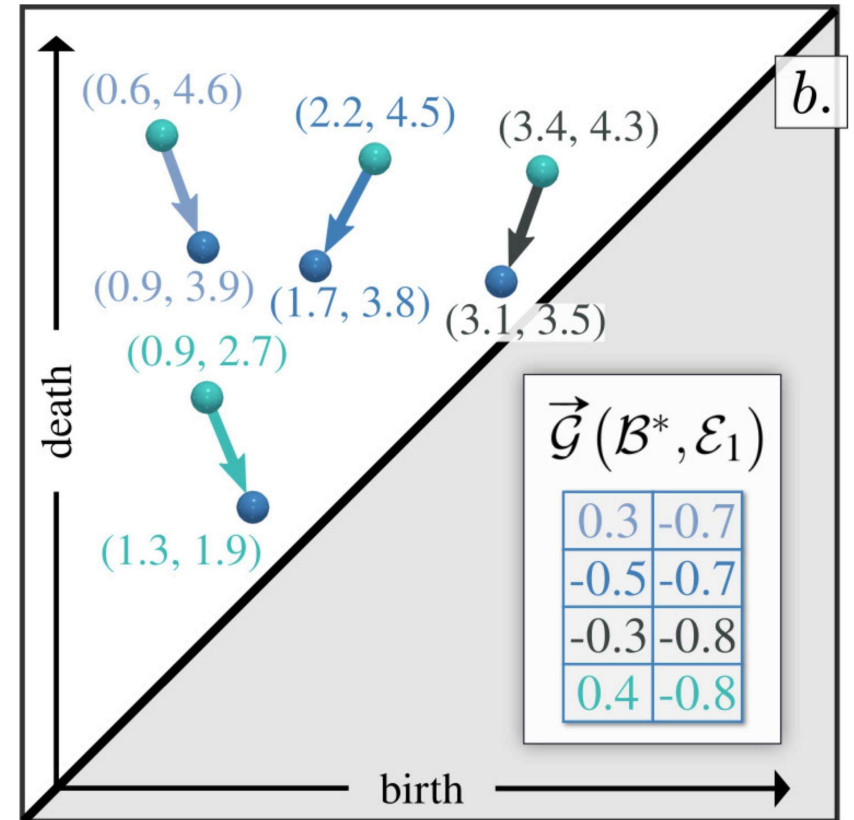
$$\vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}') \in \mathbb{R}^{2 \times |\mathcal{E}|}$$



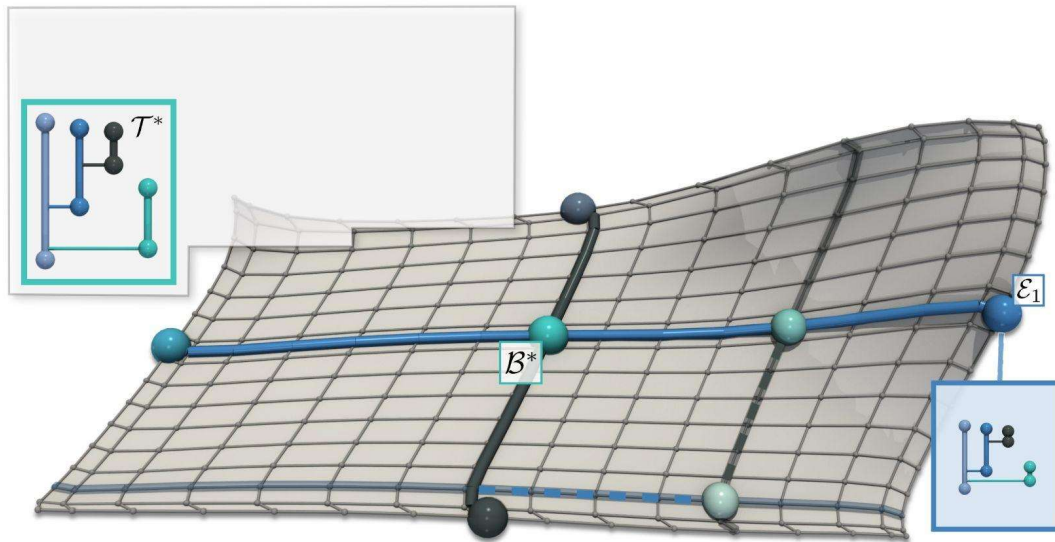
Geodesics



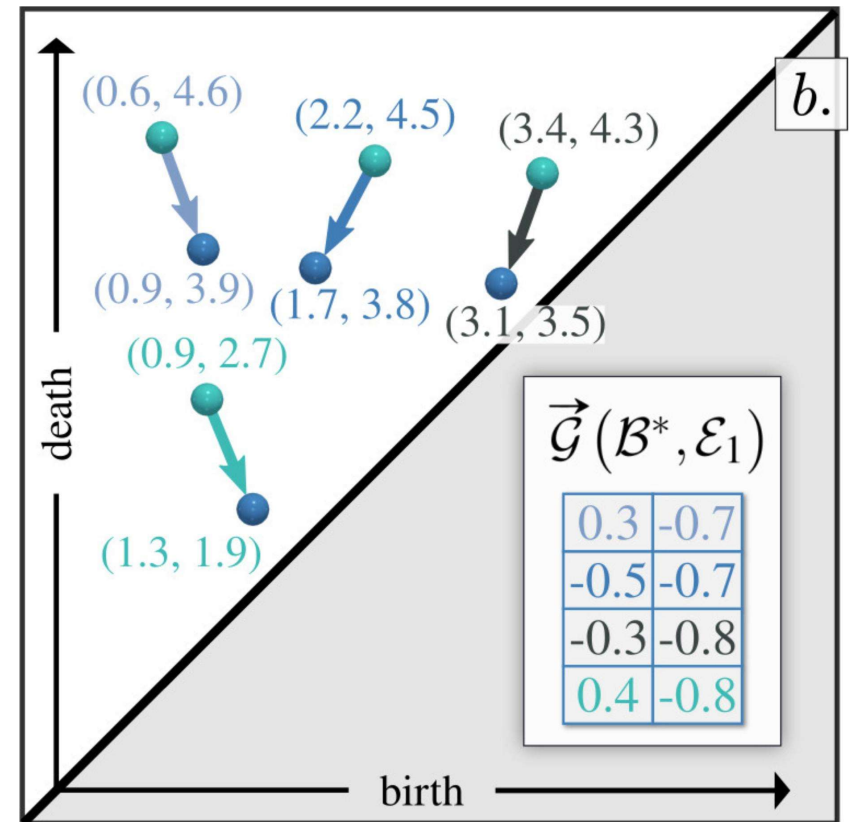
$$\vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}') \in \mathbb{R}^{2 \times |\mathcal{E}|}$$



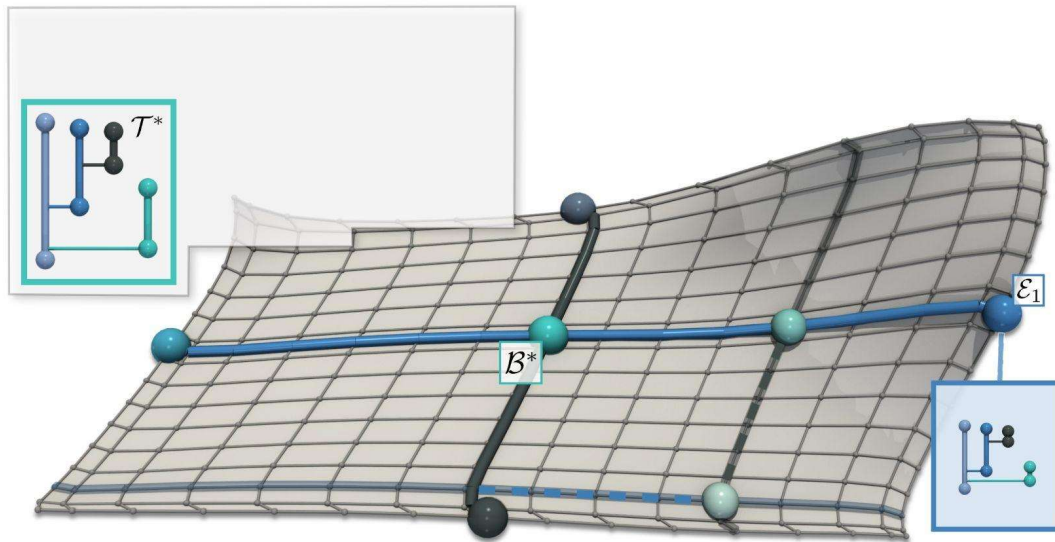
Geodesics



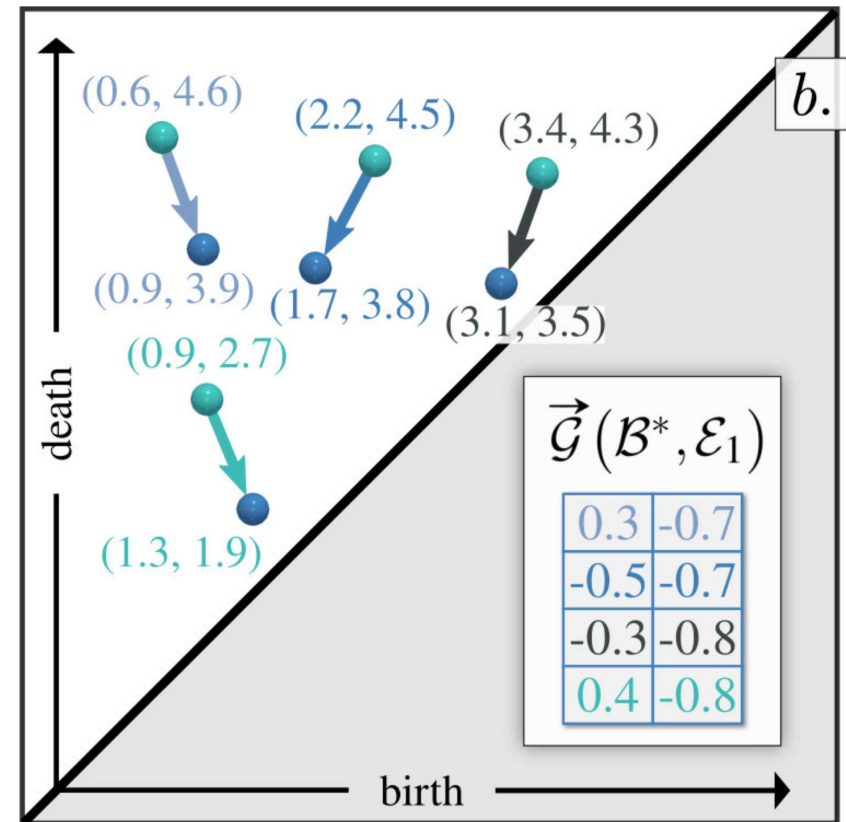
$$\vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}') \in \mathbb{R}^{2 \times |\mathcal{E}|}$$



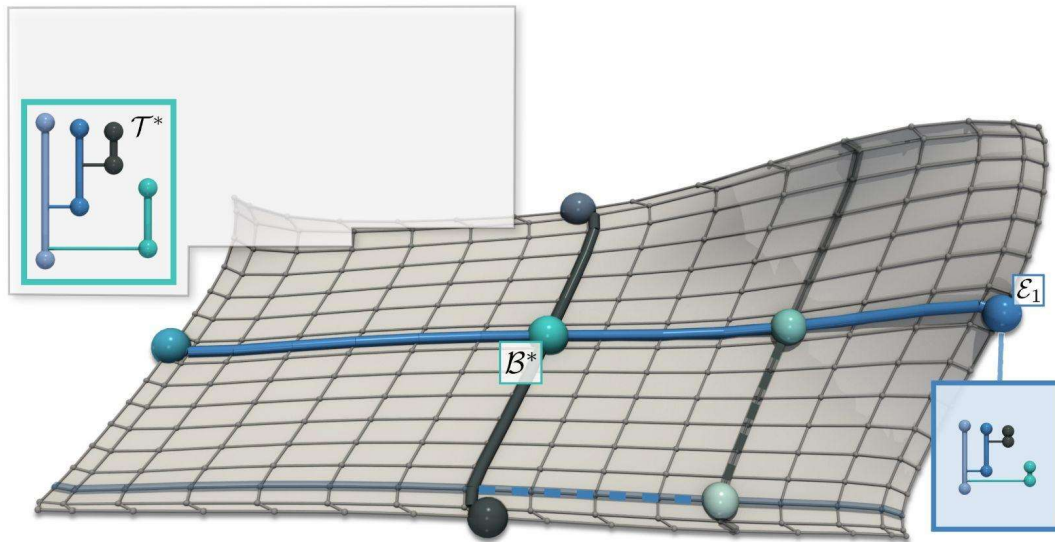
Orthogonal geodesics



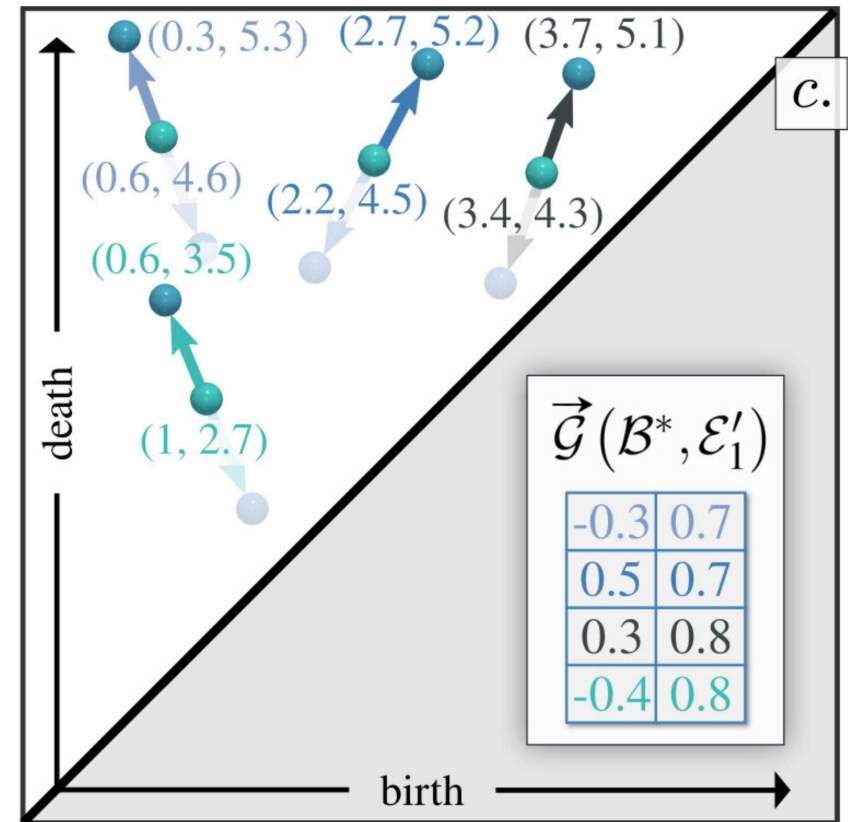
$$\vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}') \cdot \vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}'') = 0$$



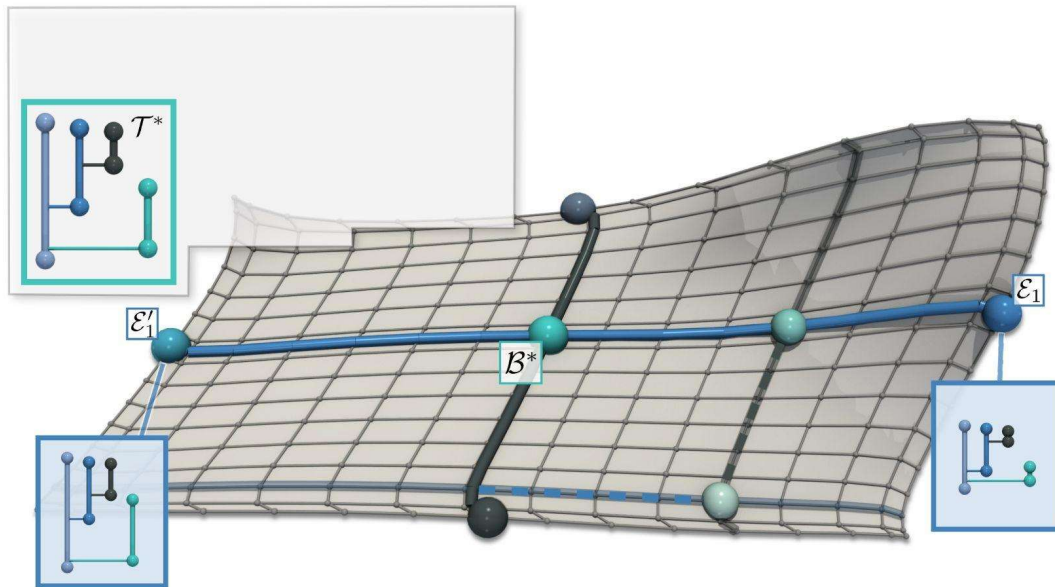
Collinear geodesics



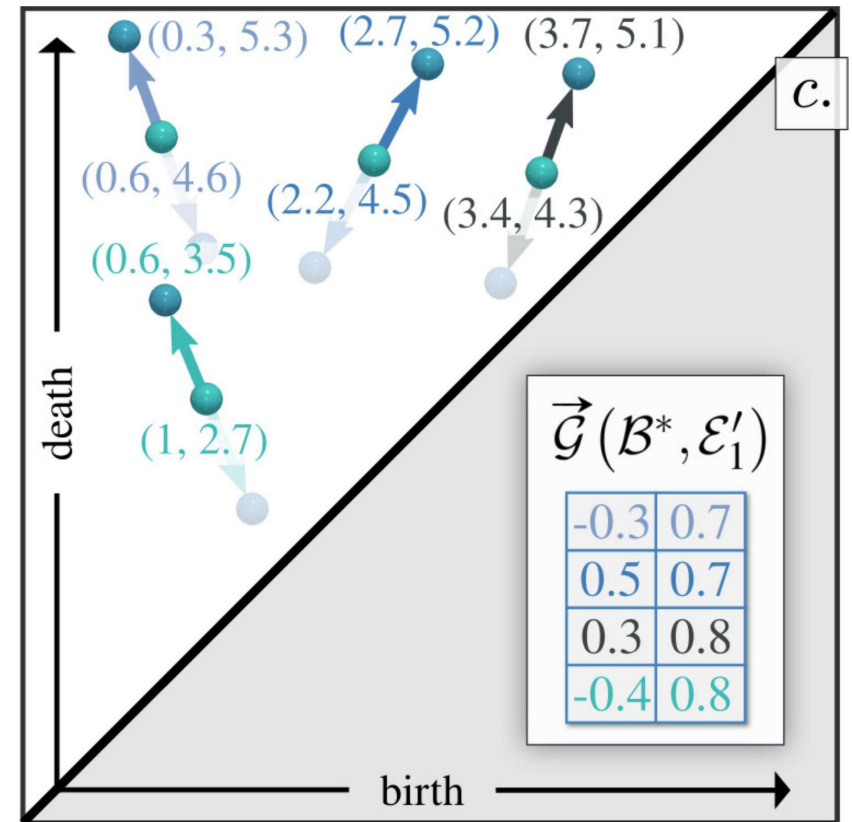
$$\vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}') = \lambda \vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}'')$$



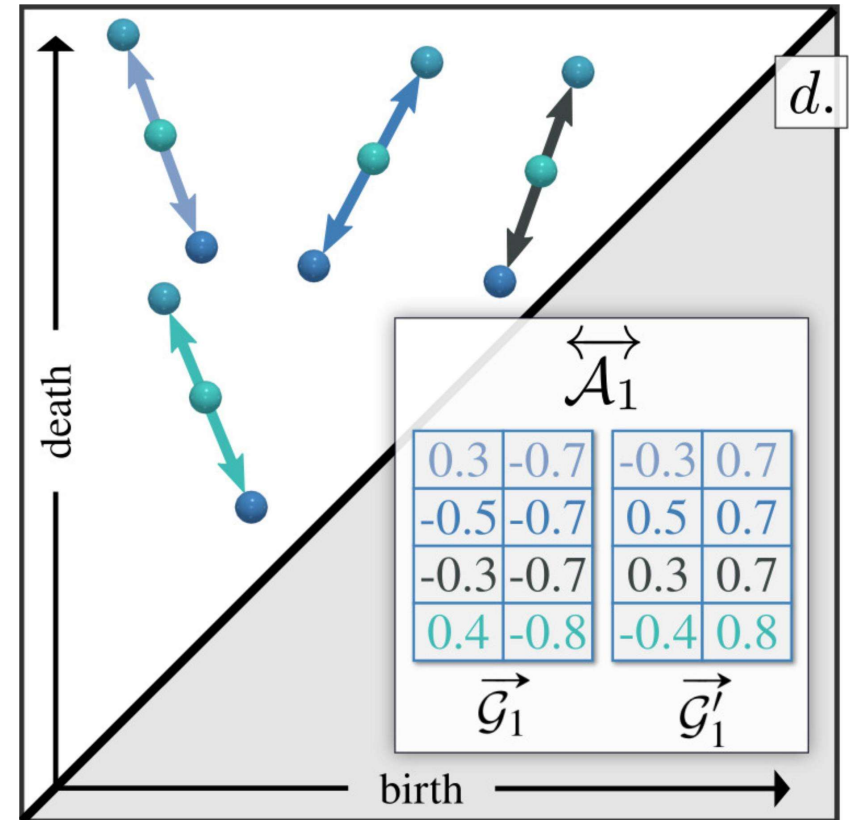
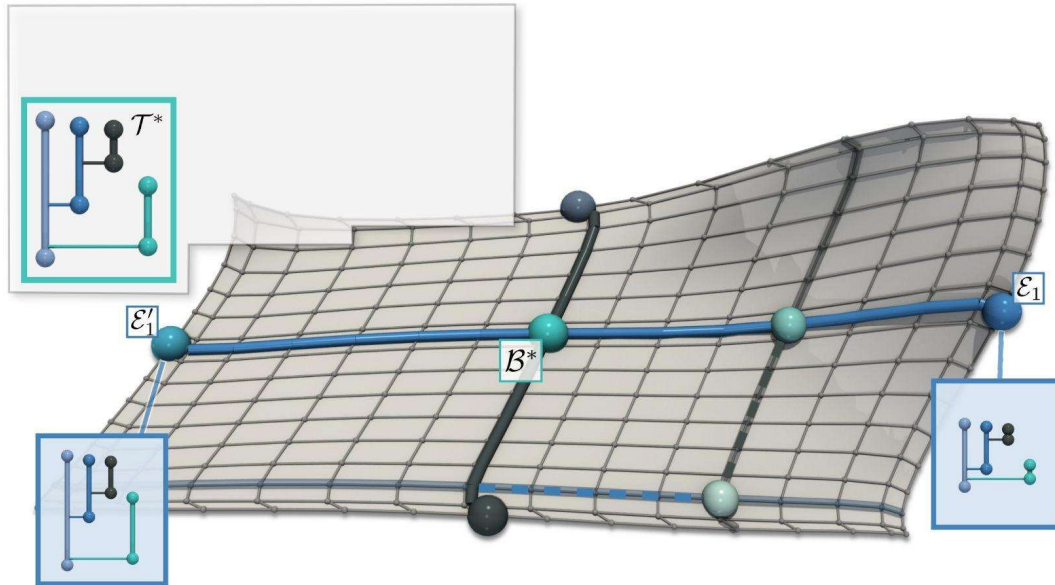
Collinear geodesics



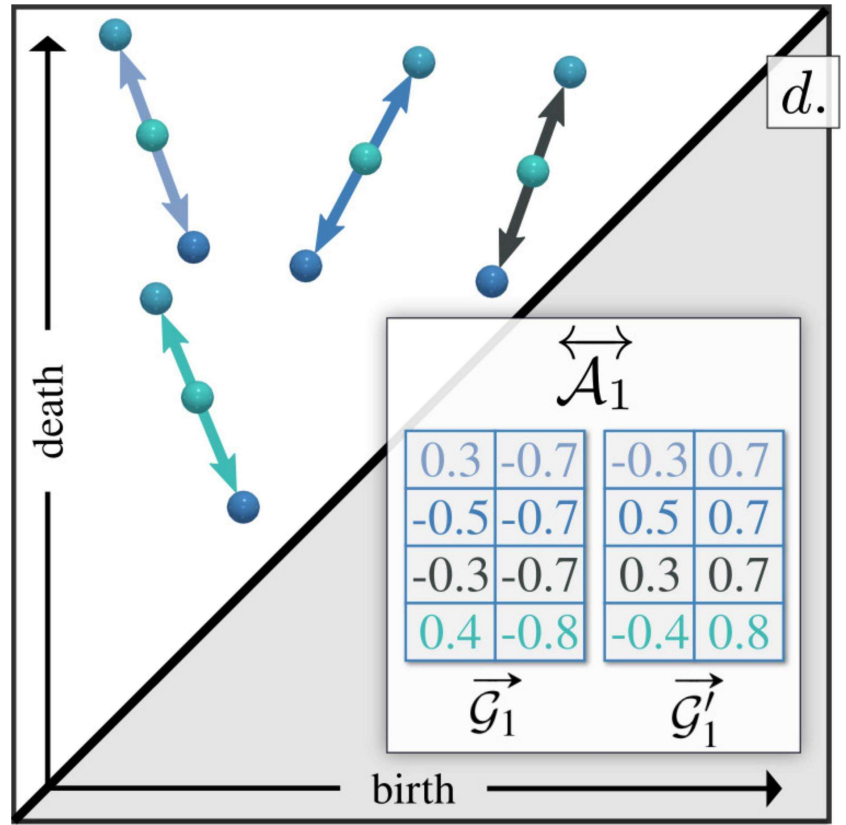
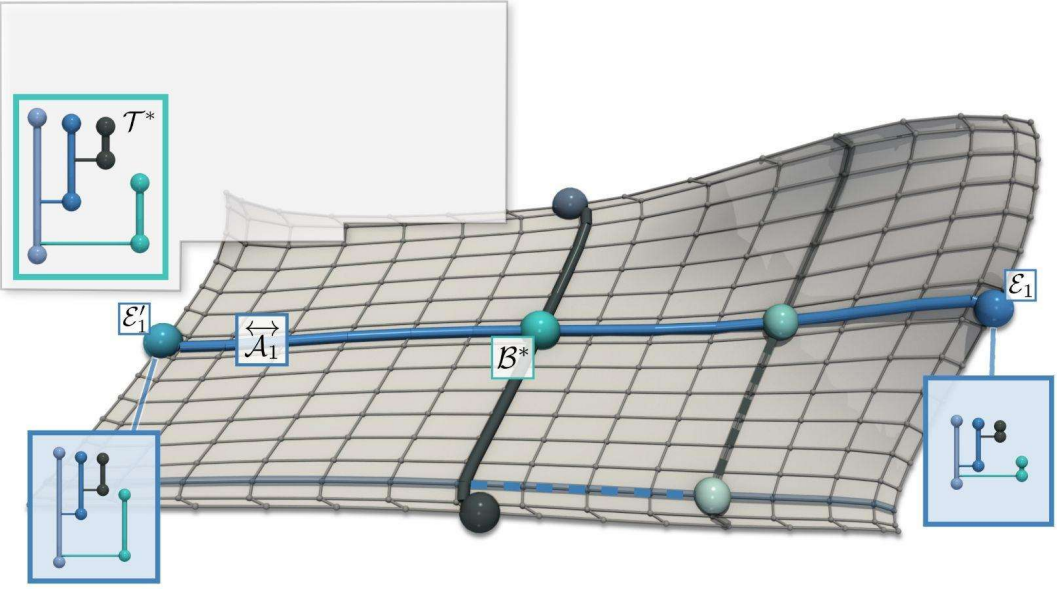
$$\vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}') = \lambda \vec{\mathcal{G}}(\mathcal{E}, \mathcal{E}'')$$



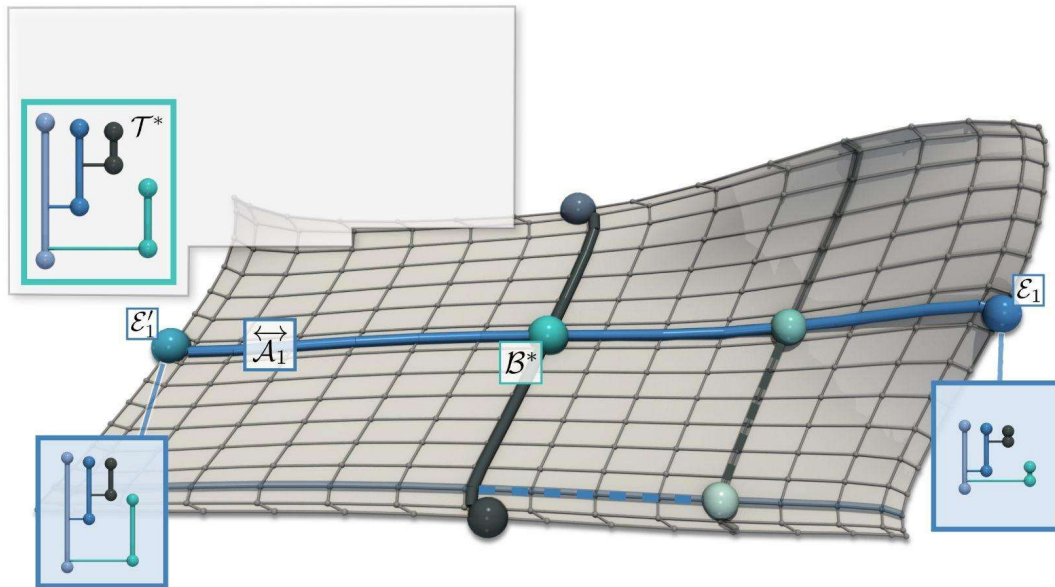
Geodesic axis



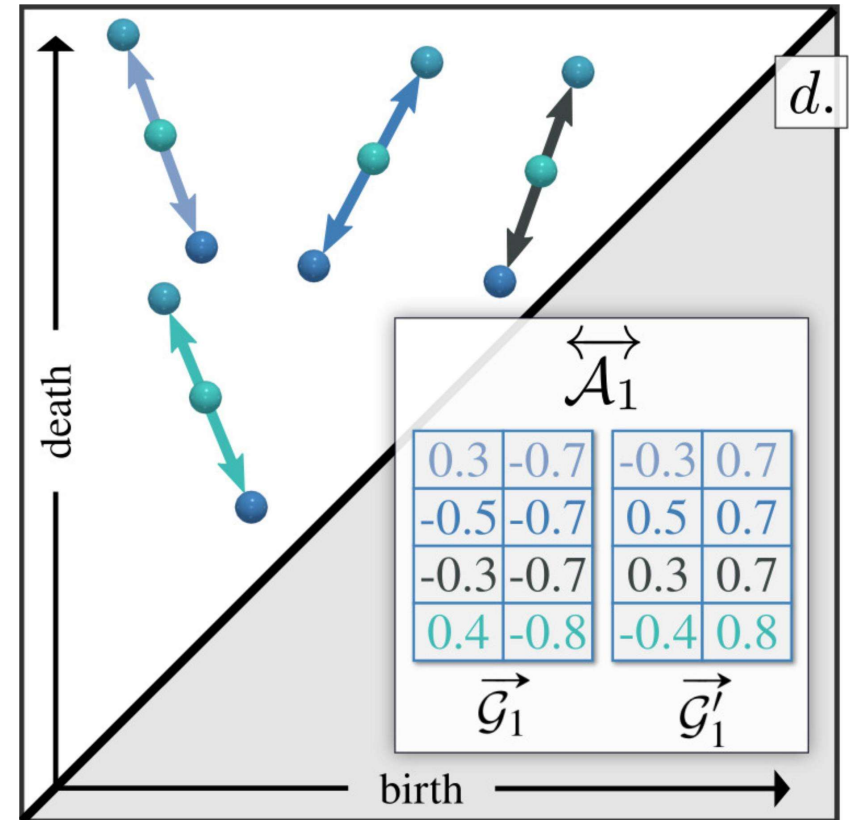
Geodesic axis



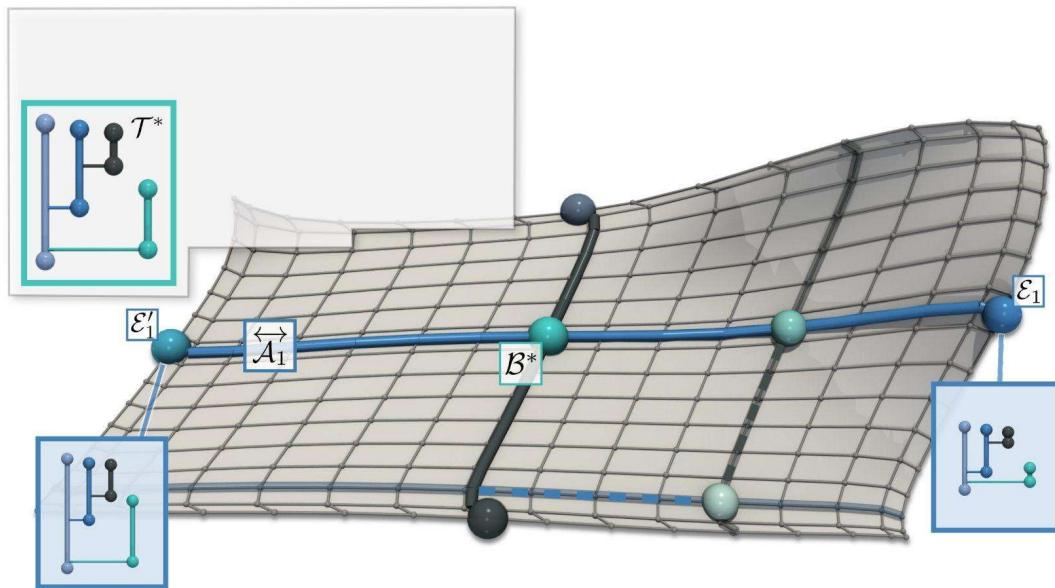
Geodesic axis



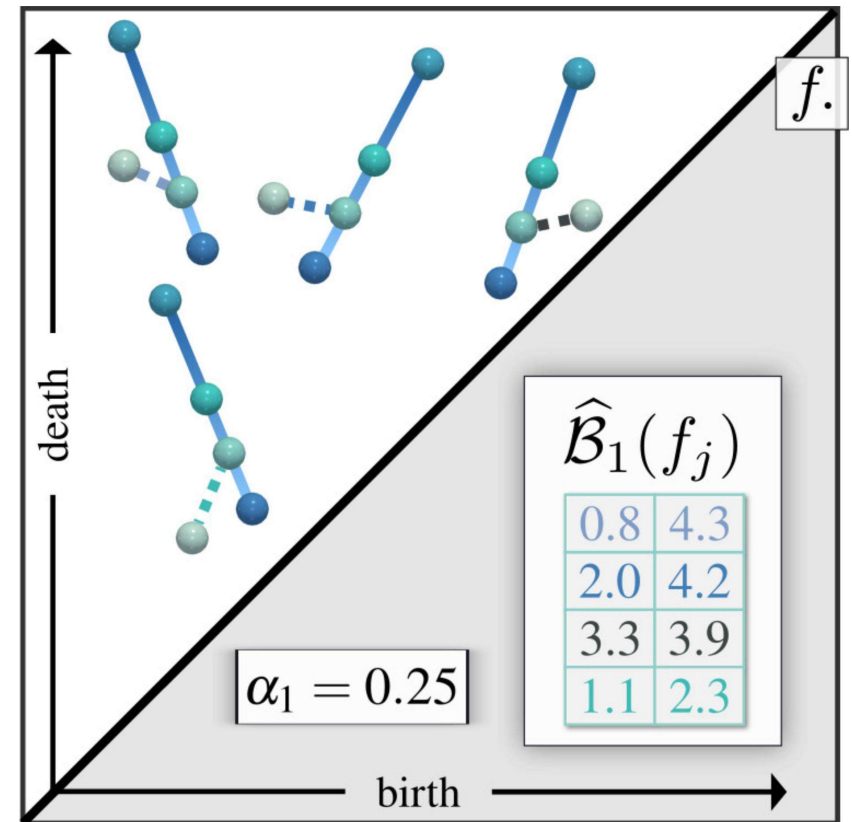
$$\vec{\mathcal{V}}_i = \vec{\mathcal{G}}_i - \vec{\mathcal{G}}'_i = \vec{\mathcal{G}}(\mathcal{E}'_i, \mathcal{E}_i)$$



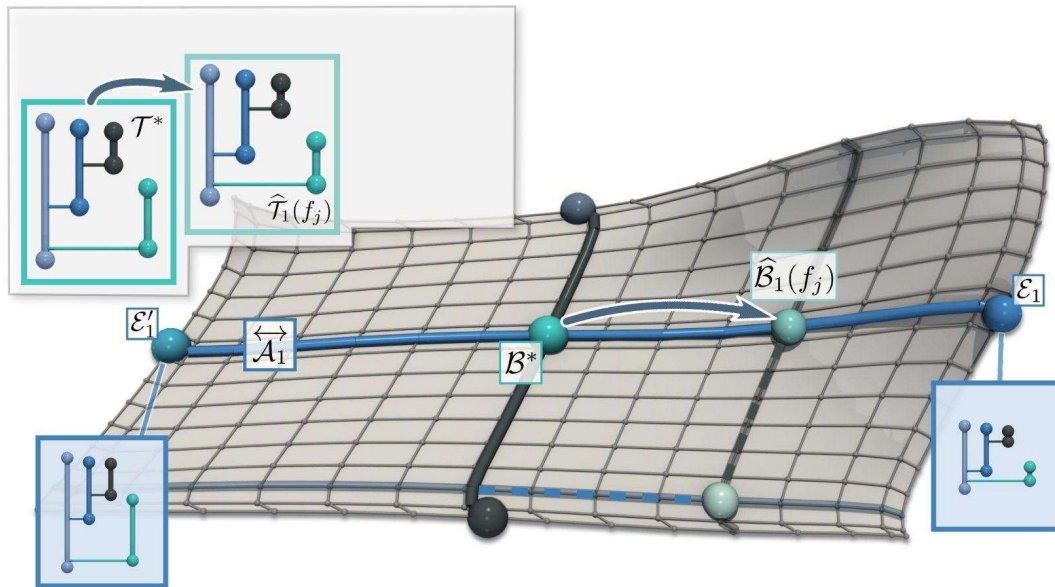
Axis projection



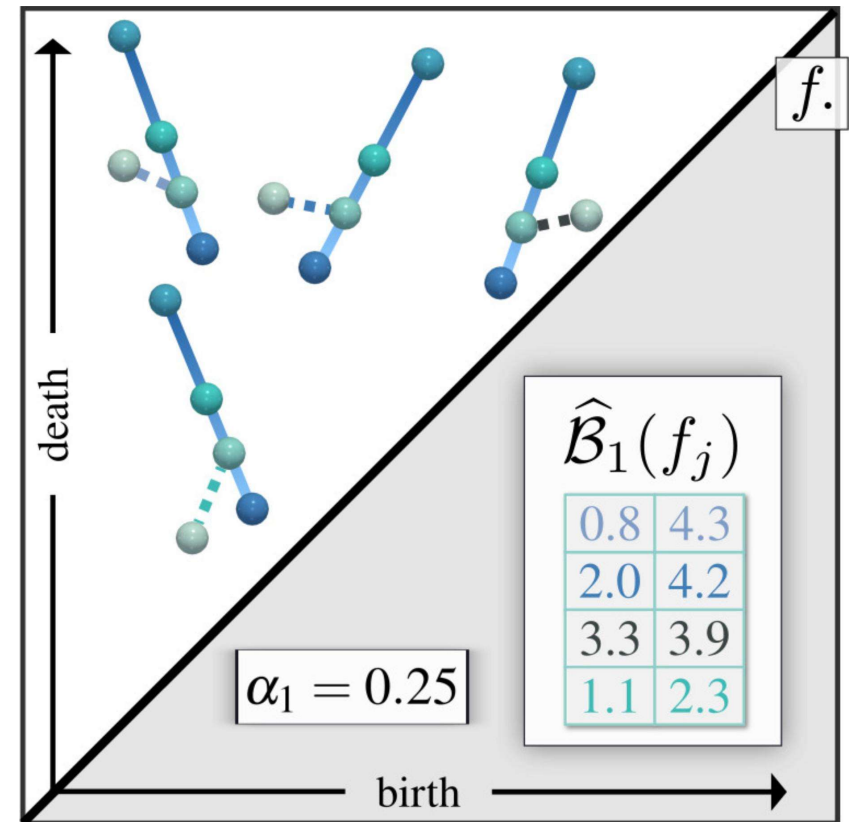
$$\mathcal{B}_{\overleftrightarrow{\mathcal{A}}_i} = \arg \min_{\mathcal{B}' \in \overleftrightarrow{\mathcal{A}}_i} (W_2^T(\mathcal{B}, \mathcal{B}'))$$



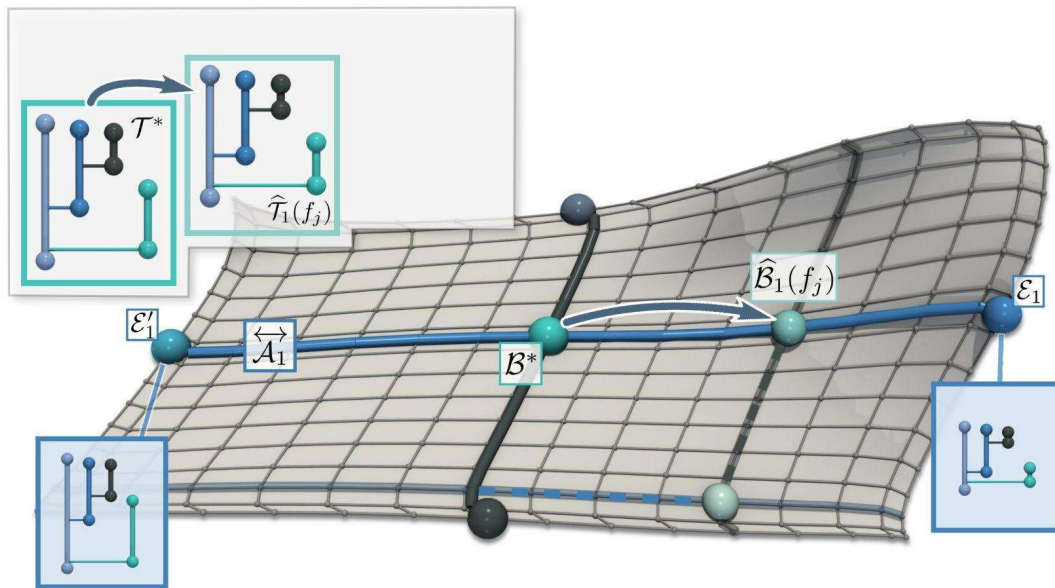
Axis projection



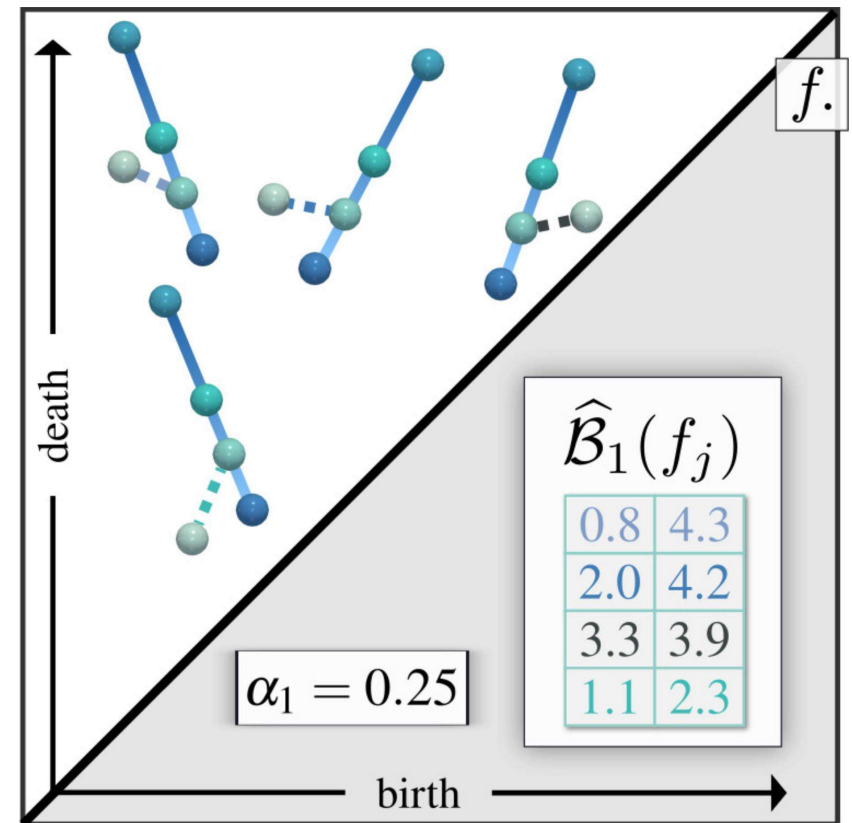
$$\mathcal{B}_{\overleftrightarrow{\mathcal{A}}_i} = \arg \min_{\mathcal{B}' \in \overleftrightarrow{\mathcal{A}}_i} (W_2^T(\mathcal{B}, \mathcal{B}'))$$



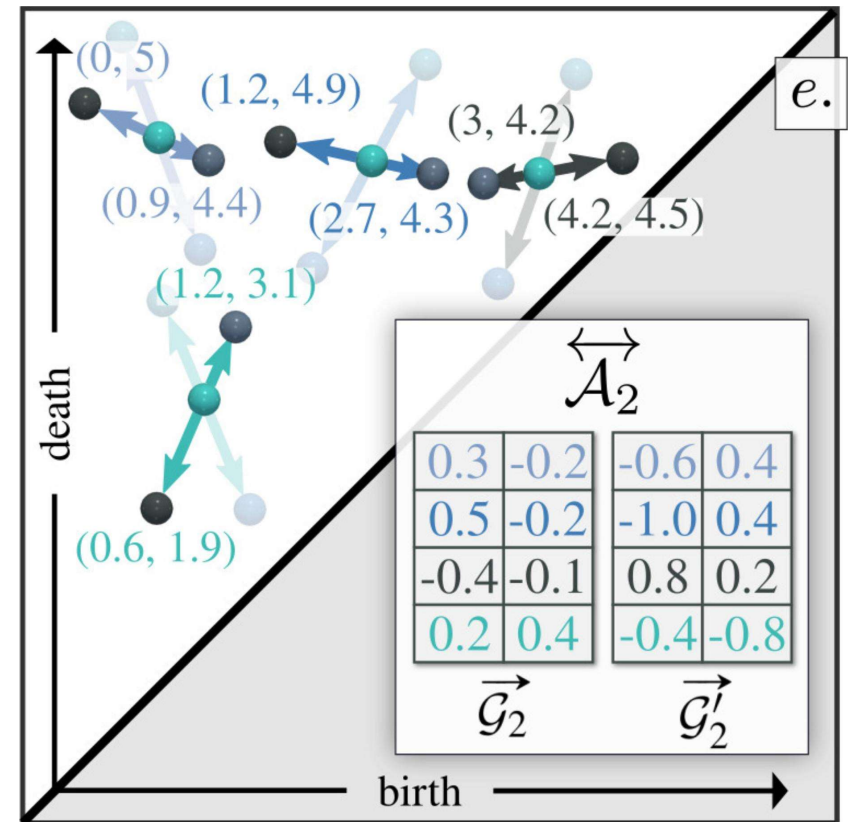
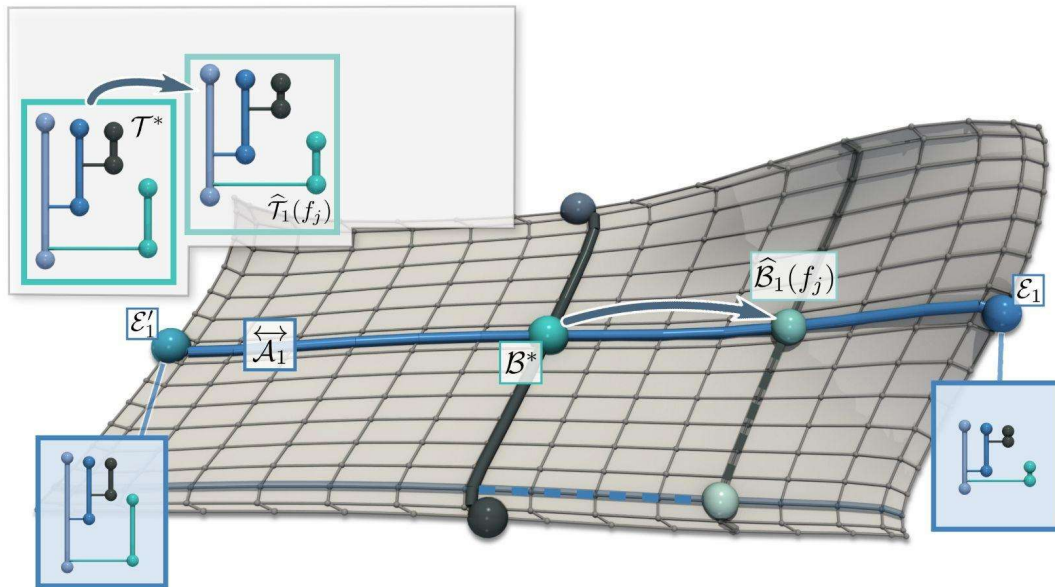
Axis projection



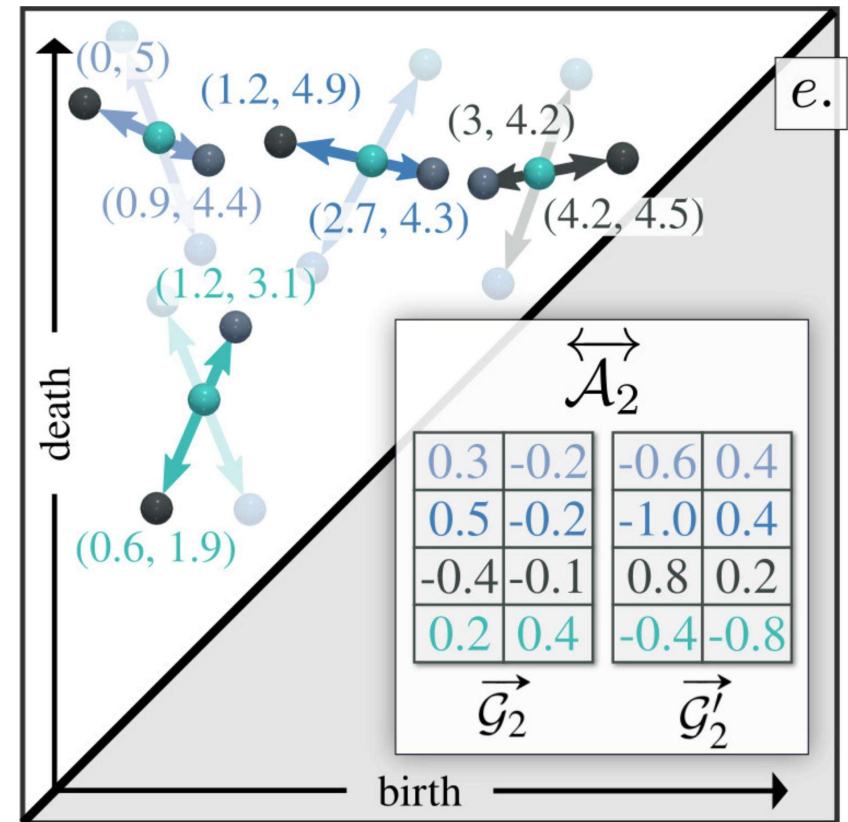
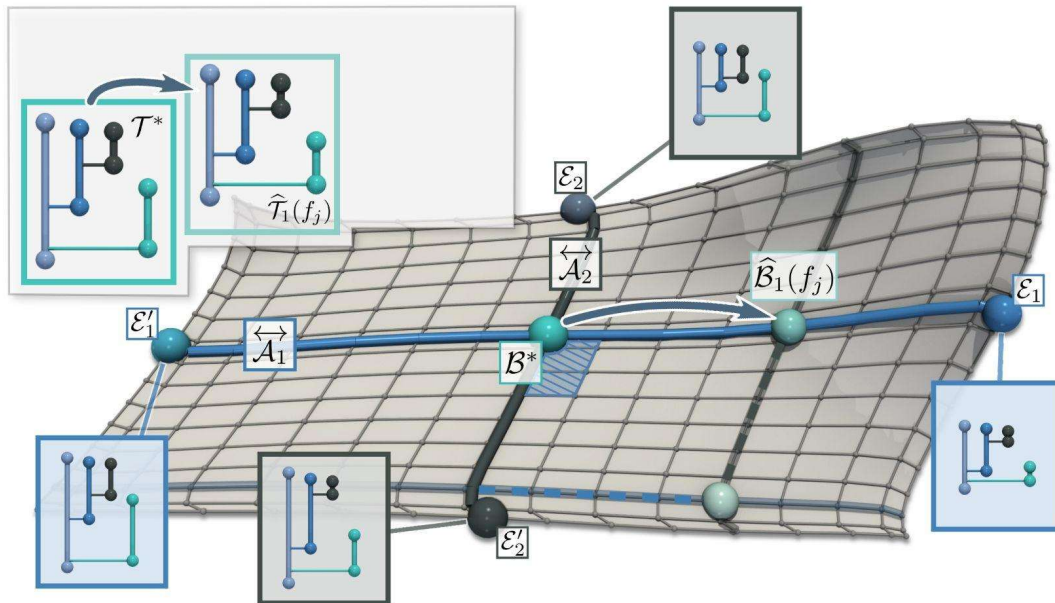
$$\mathcal{B} = \mathcal{E}_O + \vec{\mathcal{A}}_i(\mathcal{B}) = \mathcal{E}_O + \alpha_i \times \vec{\mathcal{G}}_i + (1 - \alpha_i) \times \vec{\mathcal{G}}'_i$$



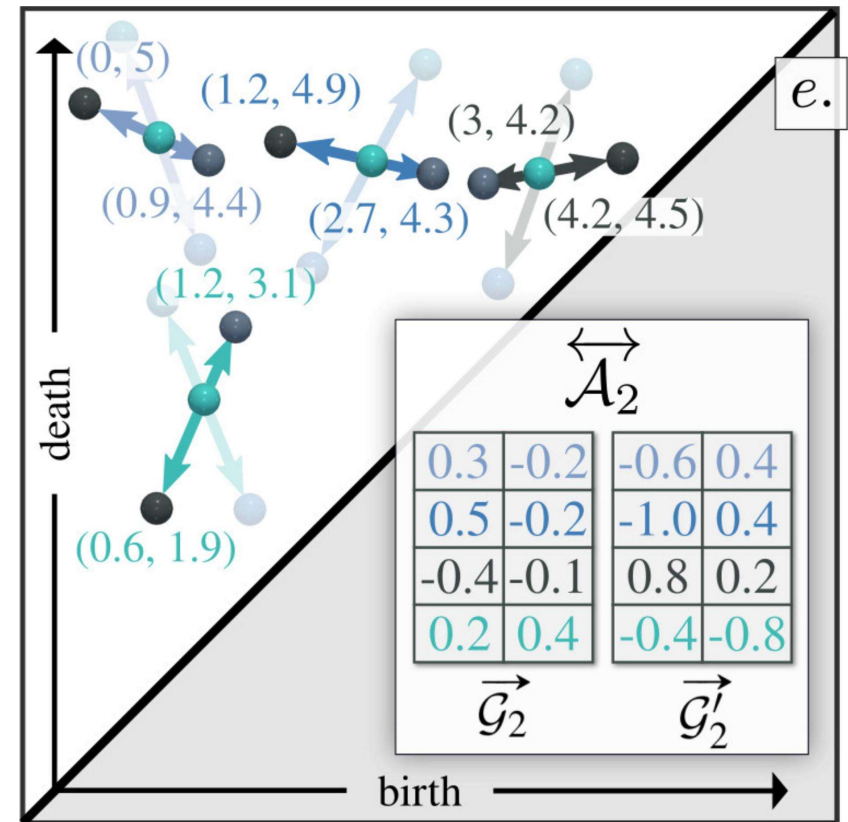
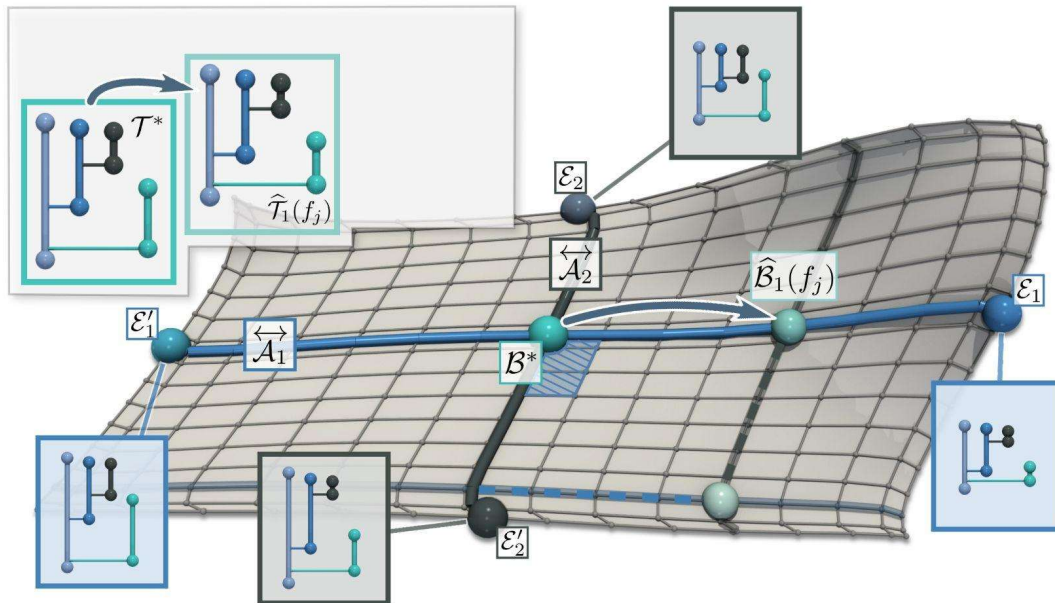
Orthogonal axis



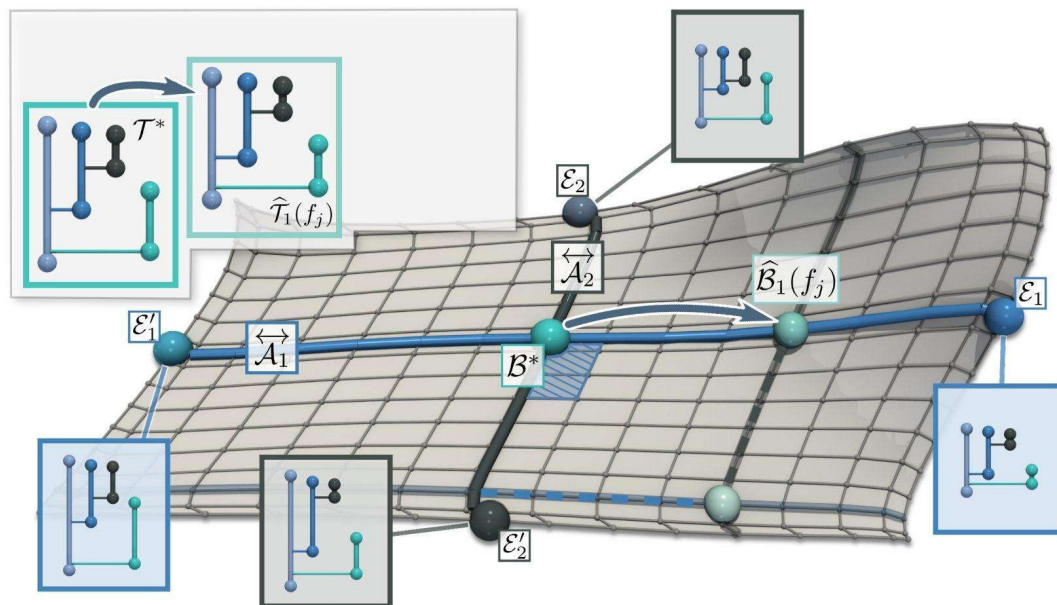
Orthogonal axis



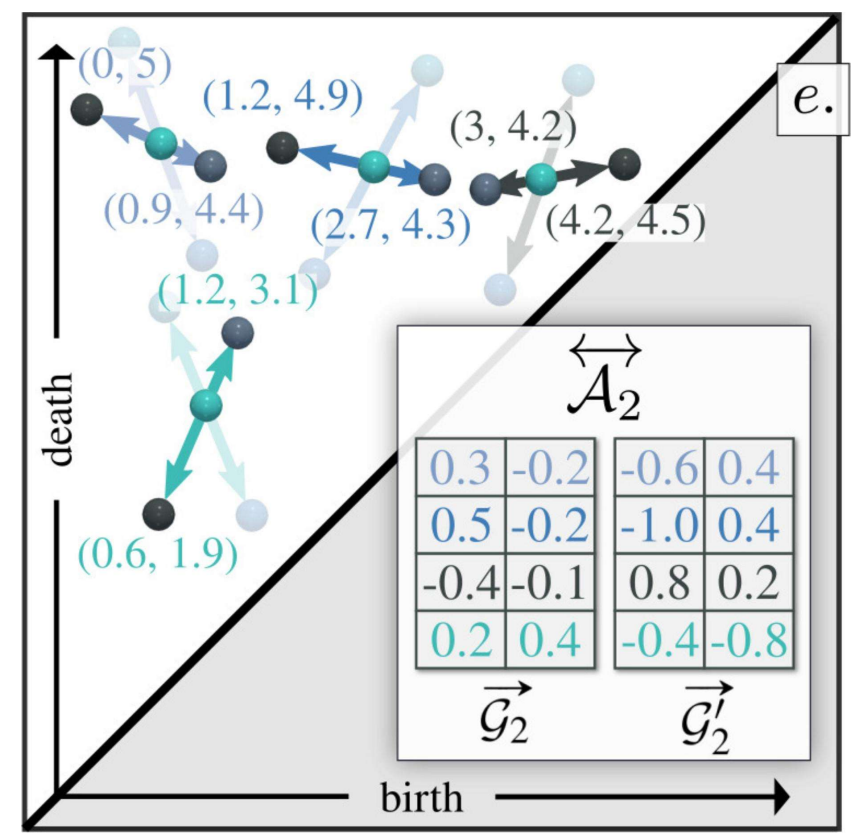
Orthogonal basis



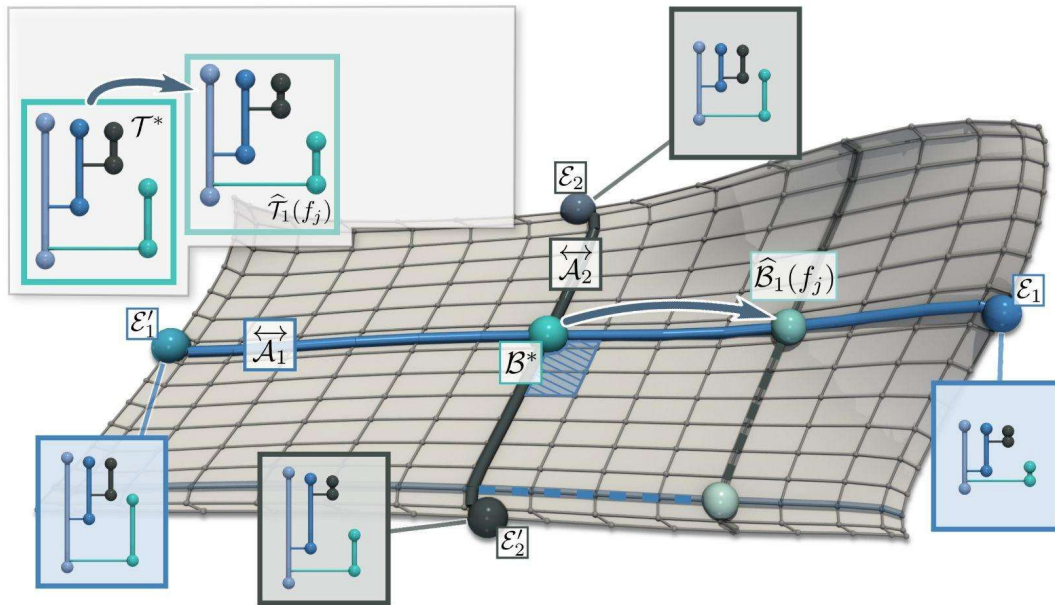
Orthogonal basis



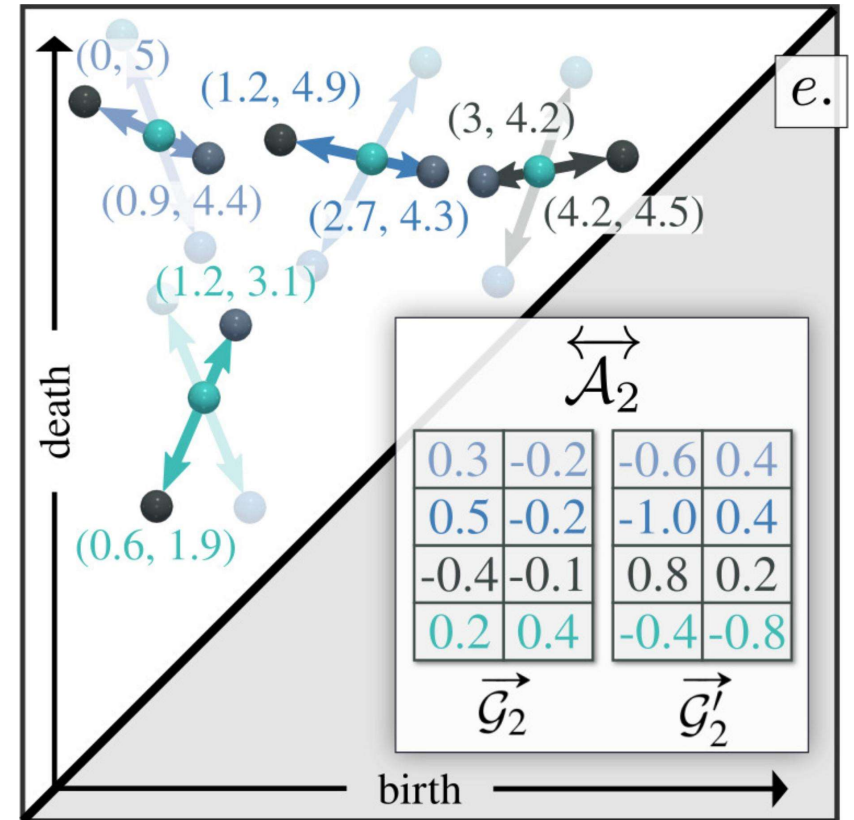
$$B_{\mathbb{B}} = \{ \overleftrightarrow{\mathcal{A}}_1, \overleftrightarrow{\mathcal{A}}_2, \dots, \overleftrightarrow{\mathcal{A}}_{d'} \}$$



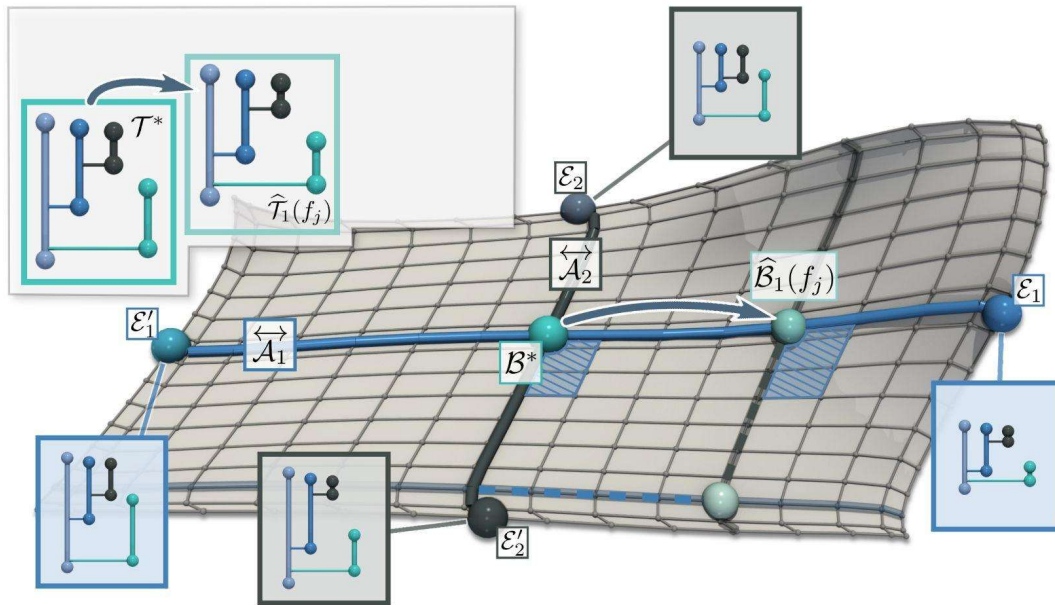
Orthogonal basis



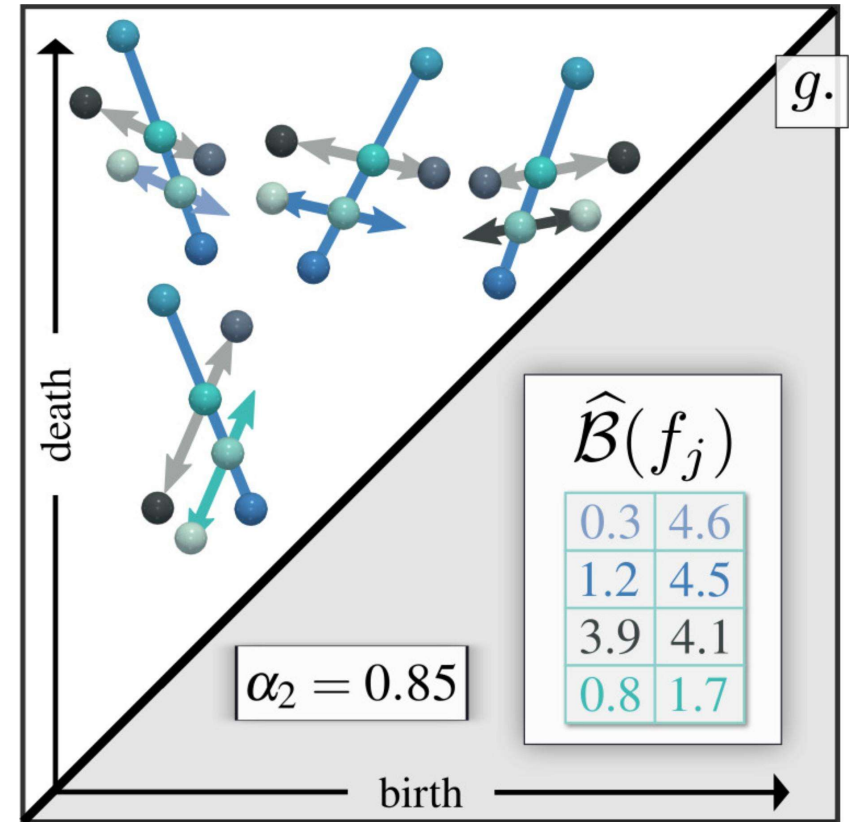
$$B_{\mathbb{B}} = \{ \overleftrightarrow{\mathcal{A}}_1, \overleftrightarrow{\mathcal{A}}_2, \dots, \overleftrightarrow{\mathcal{A}}_{d'} \} \quad \{ \overrightarrow{\mathcal{V}}_1, \overrightarrow{\mathcal{V}}_2, \dots, \overrightarrow{\mathcal{V}}_{d'} \}$$



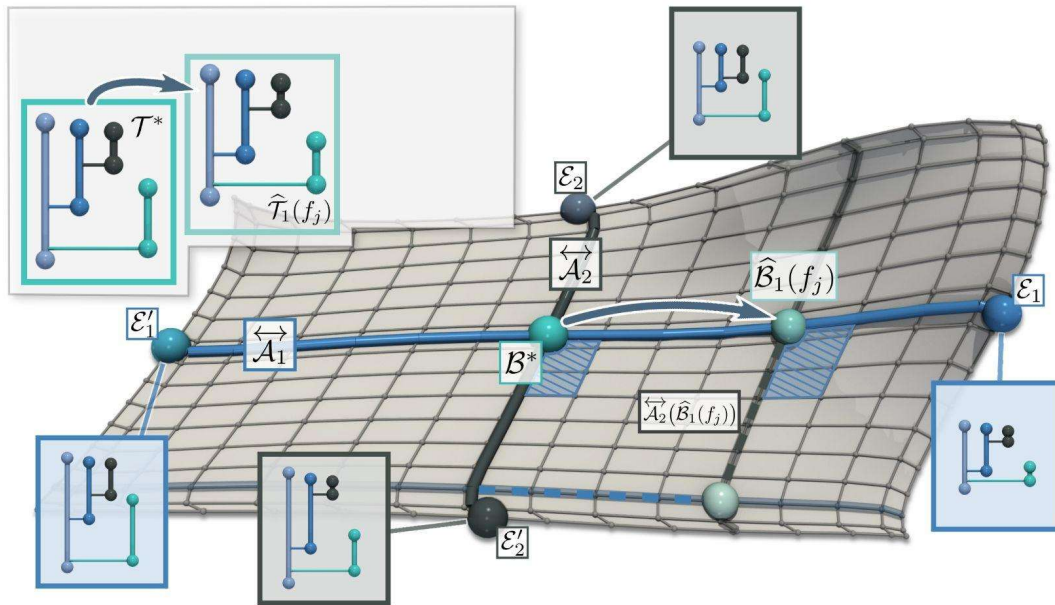
Axis translation



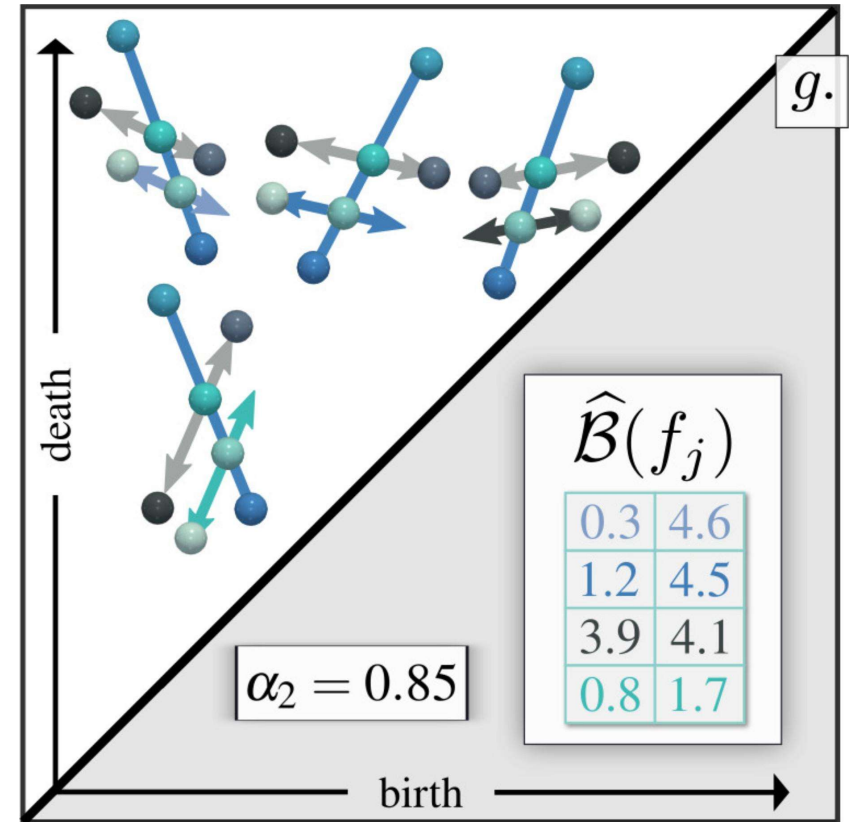
$$\overleftrightarrow{\mathcal{A}}_j(\mathcal{B}) = ((\mathcal{B}, \mathcal{B} + \overrightarrow{\mathcal{G}}_j), (\mathcal{B}, \mathcal{B} + \overrightarrow{\mathcal{G}}'_j))$$



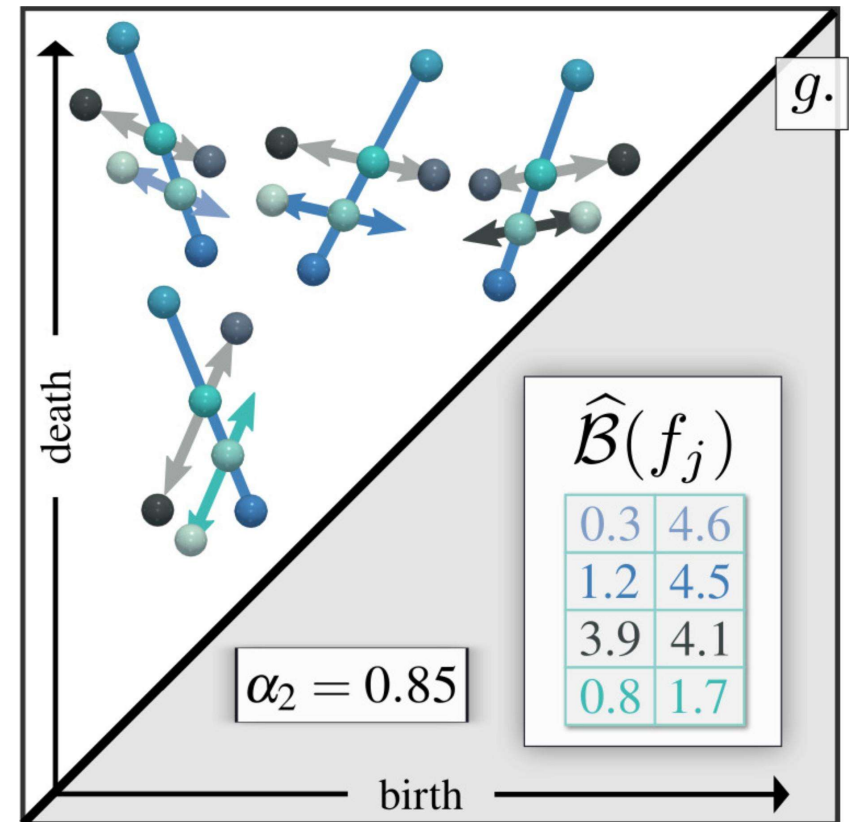
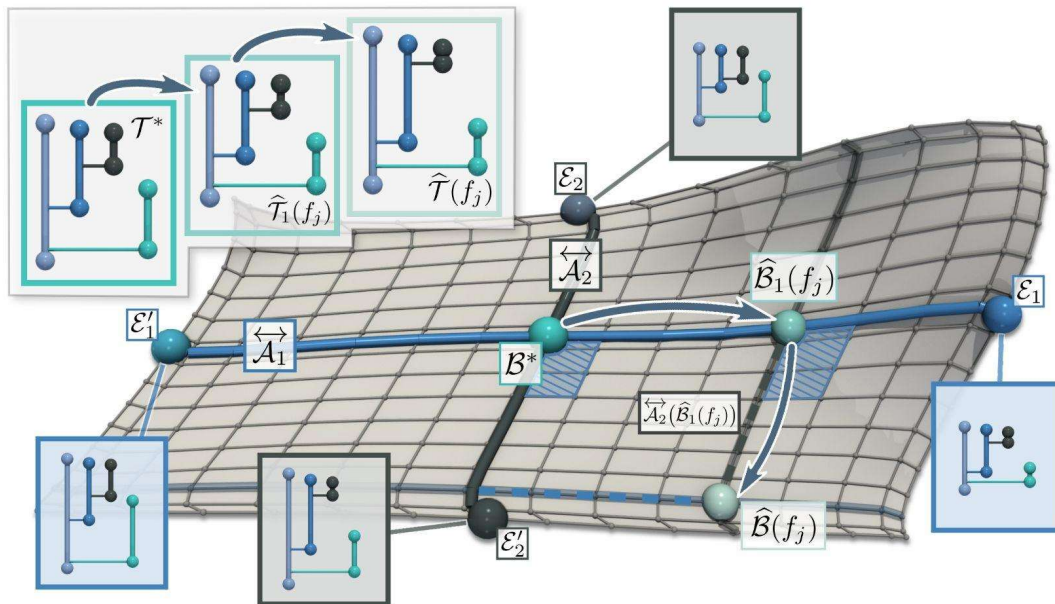
Axis translation



$$\vec{\mathcal{A}}_j(\mathcal{B}) = ((\mathcal{B}, \mathcal{B} + \vec{\mathcal{G}}_j), (\mathcal{B}, \mathcal{B} + \vec{\mathcal{G}}'_j))$$



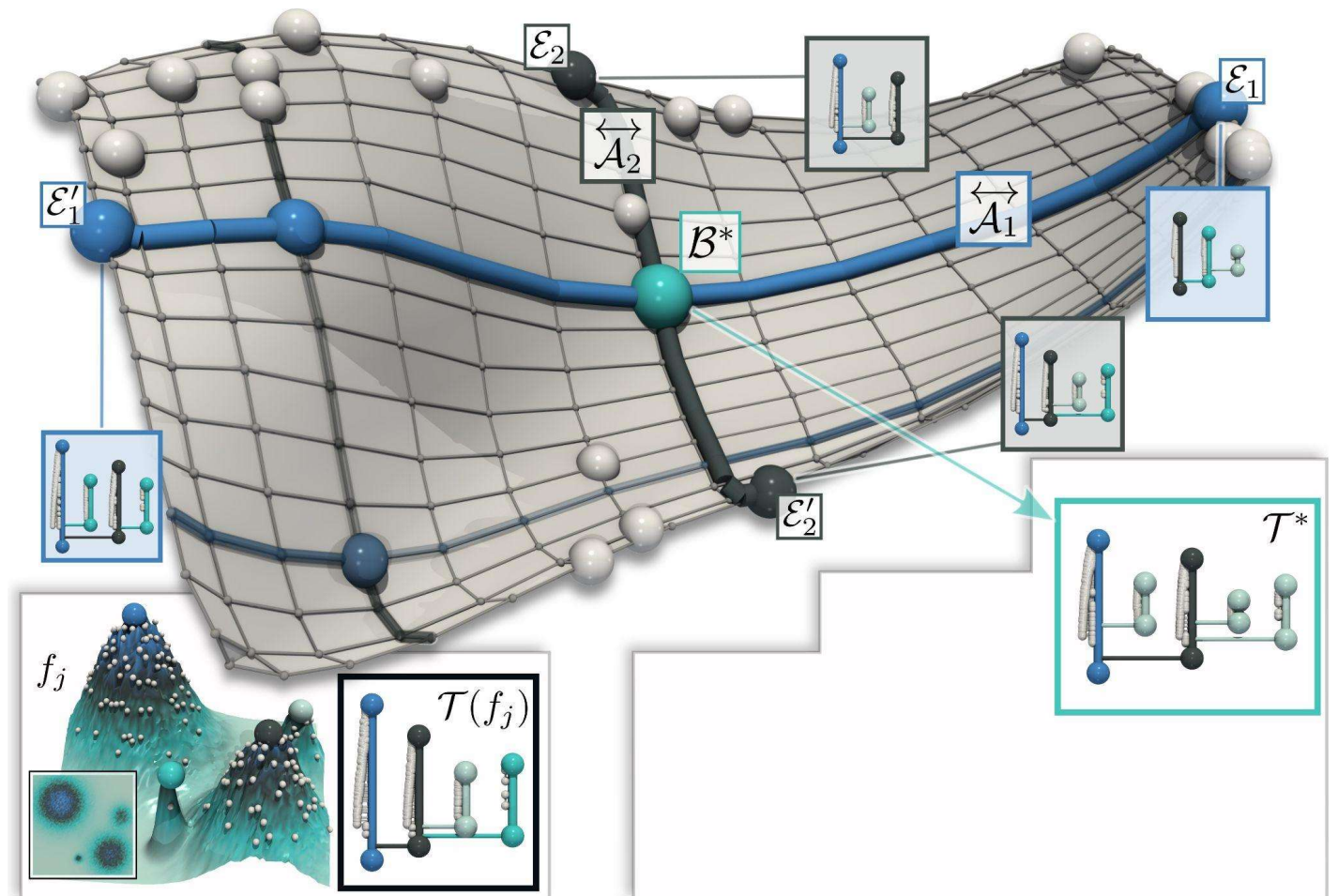
Axis translated projection



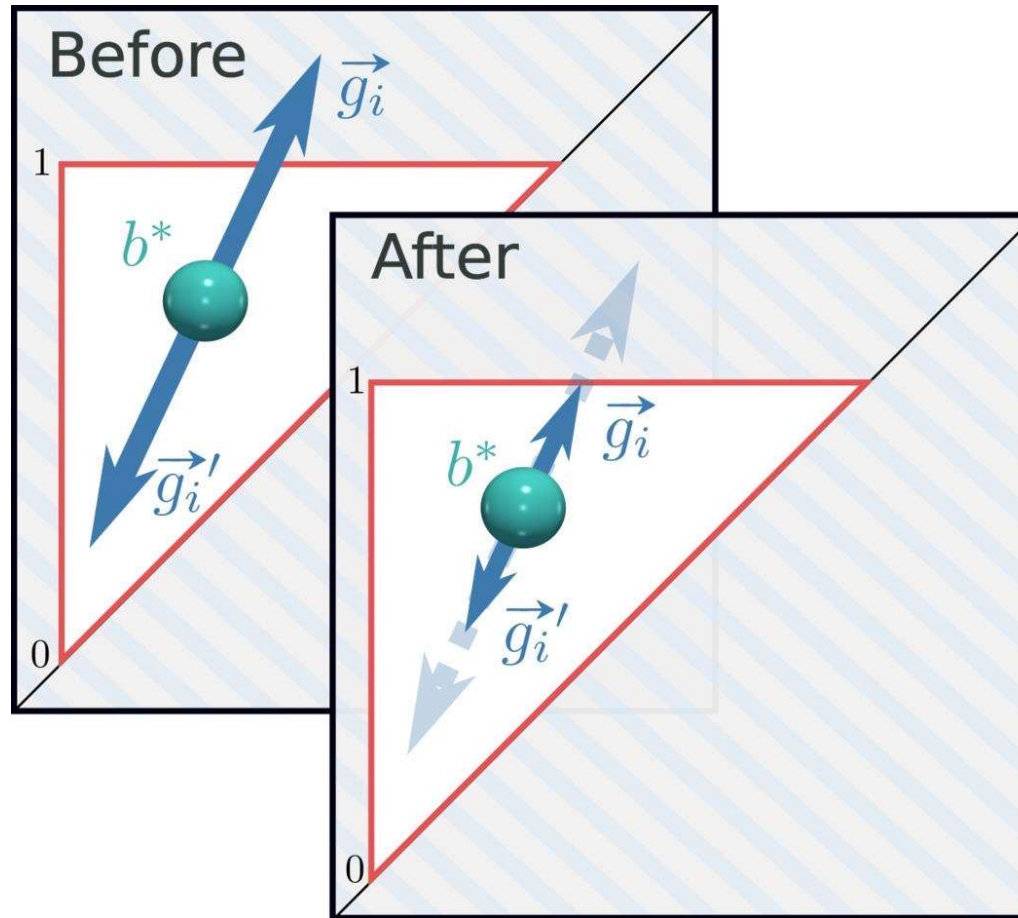
Overview

- **BDT Basis**

- Barycenter
- Geodesic axis
- Orthogonal axes



From BDTs to MTs



Overview

Algorithm 1 Merge Tree Principal Geodesic Analysis (MT-PGA) Algorithm

Input: Set of BDTs $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$.

1:
2:
3:
4:
5:
6:
7:
8:
9:
10:
11:
12:
13:
14:
15:
16:

Overview

Algorithm 1 Merge Tree Principal Geodesic Analysis (MT-PGA) Algorithm

Input: Set of BDTs $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$.

Output1: Basis origin \mathcal{B}^* ;

Output2: Basis geodesic axes $B_{\mathbb{B}} = \{\overleftrightarrow{\mathcal{A}}_1, \overleftrightarrow{\mathcal{A}}_2, \dots, \overleftrightarrow{\mathcal{A}}_{d_{\max}}\}$;

1:
2:
3:
4:
5:
6:
7:
8:
9:
10:
11:
12:
13:
14:
15:
16:

Overview

Algorithm 1 Merge Tree Principal Geodesic Analysis (MT-PGA) Algorithm

Input: Set of BDTs $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$.

Output1: Basis origin \mathcal{B}^* ;

Output2: Basis geodesic axes $B_{\mathbb{B}} = \{\overleftrightarrow{\mathcal{A}}_1, \overleftrightarrow{\mathcal{A}}_2, \dots, \overleftrightarrow{\mathcal{A}}_{d_{\max}}\}$;

Output3: Coordinates $\alpha^j \in [0, 1]^{d_{\max}}$ of the input BDTs in $B_{\mathbb{B}}$ (with $j \in \{1, 2, \dots, N\}$).

1:
2:
3:
4:
5:
6:
7:
8:
9:
10:
11:
12:
13:
14:
15:
16:

Overview

Algorithm 1 Merge Tree Principal Geodesic Analysis (MT-PGA) Algorithm

Input: Set of BDTs $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$.

Output1: Basis origin \mathcal{B}^* ;

Output2: Basis geodesic axes $B_{\mathbb{B}} = \{\overleftarrow{\mathcal{A}}_1, \overleftarrow{\mathcal{A}}_2, \dots, \overleftarrow{\mathcal{A}}_{d_{max}}\}$;

Output3: Coordinates $\alpha^j \in [0, 1]^{d_{max}}$ of the input BDTs in $B_{\mathbb{B}}$ (with $j \in \{1, 2, \dots, N\}$).

1: $\mathcal{B}^* \leftarrow \text{WassersteinBarycenter}(\mathcal{S}_{\mathcal{B}})$;

2:

3:

4:

5:

6:

7:

8:

9:

10:

11:

12:

13:

14:

15:

16:

Overview

Algorithm 1 Merge Tree Principal Geodesic Analysis (MT-PGA) Algorithm

Input: Set of BDTs $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$.

Output1: Basis origin \mathcal{B}^* ;

Output2: Basis geodesic axes $B_{\mathbb{B}} = \{\overleftarrow{\mathcal{A}}_1, \overleftarrow{\mathcal{A}}_2, \dots, \overleftarrow{\mathcal{A}}_{d_{max}}\}$;

Output3: Coordinates $\alpha^j \in [0, 1]^{d_{max}}$ of the input BDTs in $B_{\mathbb{B}}$ (with $j \in \{1, 2, \dots, N\}$).

1: $\mathcal{B}^* \leftarrow \text{WassersteinBarycenter}(\mathcal{S}_{\mathcal{B}})$;

2: **for** $d' \in \{1, 2, \dots, d_{max}\}$ **do**

3:

4:

5:

6:

7:

8:

9:

10:

11:

12:

13:

14:

15:

16: **end for**

Overview

Algorithm 1 Merge Tree Principal Geodesic Analysis (MT-PGA) Algorithm

Input: Set of BDTs $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$.

Output1: Basis origin \mathcal{B}^* ;

Output2: Basis geodesic axes $B_{\mathbb{B}} = \{\overleftrightarrow{\mathcal{A}}_1, \overleftrightarrow{\mathcal{A}}_2, \dots, \overleftrightarrow{\mathcal{A}}_{d_{max}}\}$;

Output3: Coordinates $\alpha^j \in [0, 1]^{d_{max}}$ of the input BDTs in $B_{\mathbb{B}}$ (with $j \in \{1, 2, \dots, N\}$).

- 1: $\mathcal{B}^* \leftarrow \text{WassersteinBarycenter}(\mathcal{S}_{\mathcal{B}})$;
 - 2: **for** $d' \in \{1, 2, \dots, d_{max}\}$ **do**
 - 3: $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{InitializeGeodesicAxis}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}})$;
 - 4:
 - 5:
 - 6:
 - 7:
 - 8:
 - 9:
 - 10:
 - 11:
 - 12:
 - 13:
 - 14:
 - 15:
 - 16: **end for**
-

Overview

Algorithm 1 Merge Tree Principal Geodesic Analysis (MT-PGA) Algorithm

Input: Set of BDTs $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$.

Output1: Basis origin \mathcal{B}^* ;

Output2: Basis geodesic axes $B_{\mathbb{B}} = \{\overleftrightarrow{\mathcal{A}}_1, \overleftrightarrow{\mathcal{A}}_2, \dots, \overleftrightarrow{\mathcal{A}}_{d_{max}}\}$;

Output3: Coordinates $\alpha^j \in [0, 1]^{d_{max}}$ of the input BDTs in $B_{\mathbb{B}}$ (with $j \in \{1, 2, \dots, N\}$).

```
1:  $\mathcal{B}^* \leftarrow \text{WassersteinBarycenter}(\mathcal{S}_{\mathcal{B}})$ ;  
2: for  $d' \in \{1, 2, \dots, d_{max}\}$  do  
3:    $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{InitializeGeodesicAxis}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}})$ ;  
4:   while  $E_{W_2^{\mathcal{T}}}(B_{\mathbb{B}})$  decreases do  
5:     // Optimize the current geodesic axis  $\overleftrightarrow{\mathcal{A}}_{d'}$   
6:  
7:  
8:  
9:  
10:  
11:  
12:  
13:  
14:  
15:   end while  
16: end for
```

Overview

Algorithm 1 Merge Tree Principal Geodesic Analysis (MT-PGA) Algorithm

Input: Set of BDTs $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$.

Output1: Basis origin \mathcal{B}^* ;

Output2: Basis geodesic axes $B_{\mathbb{B}} = \{\overleftrightarrow{\mathcal{A}}_1, \overleftrightarrow{\mathcal{A}}_2, \dots, \overleftrightarrow{\mathcal{A}}_{d_{max}}\}$;

Output3: Coordinates $\alpha^j \in [0, 1]^{d_{max}}$ of the input BDTs in $B_{\mathbb{B}}$ (with $j \in \{1, 2, \dots, N\}$).

```
1:  $\mathcal{B}^* \leftarrow \text{WassersteinBarycenter}(\mathcal{S}_{\mathcal{B}})$ ;  
2: for  $d' \in \{1, 2, \dots, d_{max}\}$  do  
3:    $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{InitializeGeodesicAxis}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}})$ ;  
4:   while  $E_{W_2^{\mathcal{T}}}(B_{\mathbb{B}})$  decreases do  
5:     // Optimize the current geodesic axis  $\overleftrightarrow{\mathcal{A}}_{d'}$   
6:      $\alpha_{d'}^{j \in \{1, 2, \dots, N\}} \leftarrow \text{ProjectTrees}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$ ;  
7:  
8:  
9:  
10:  
11:  
12:  
13:  
14:  
15:   end while  
16: end for
```

Overview

Algorithm 1 Merge Tree Principal Geodesic Analysis (MT-PGA) Algorithm

Input: Set of BDTs $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$.

Output1: Basis origin \mathcal{B}^* ;

Output2: Basis geodesic axes $B_{\mathbb{B}} = \{\overleftrightarrow{\mathcal{A}}_1, \overleftrightarrow{\mathcal{A}}_2, \dots, \overleftrightarrow{\mathcal{A}}_{d_{max}}\}$;

Output3: Coordinates $\alpha^j \in [0, 1]^{d_{max}}$ of the input BDTs in $B_{\mathbb{B}}$ (with $j \in \{1, 2, \dots, N\}$).

- 1: $\mathcal{B}^* \leftarrow \text{WassersteinBarycenter}(\mathcal{S}_{\mathcal{B}})$;
 - 2: **for** $d' \in \{1, 2, \dots, d_{max}\}$ **do**
 - 3: $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{InitializeGeodesicAxis}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}})$;
 - 4: **while** $E_{W_2^{\mathcal{T}}}(B_{\mathbb{B}})$ decreases **do**
 - 5: // Optimize the current geodesic axis $\overleftrightarrow{\mathcal{A}}_{d'}$
 - 6: $\alpha^{j \in \{1, 2, \dots, N\}} \leftarrow \text{ProjectTrees}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$;
 - 7: $E_{W_2^{\mathcal{T}}}(B_{\mathbb{B}}) \leftarrow \text{EvaluateFittingEnergy}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$;
 - 8:
 - 9:
 - 10:
 - 11:
 - 12:
 - 13:
 - 14:
 - 15: **end while**
 - 16: **end for**
-

Overview

Algorithm 1 Merge Tree Principal Geodesic Analysis (MT-PGA) Algorithm

Input: Set of BDTs $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$.

Output1: Basis origin \mathcal{B}^* ;

Output2: Basis geodesic axes $B_{\mathbb{B}} = \{\overleftrightarrow{\mathcal{A}}_1, \overleftrightarrow{\mathcal{A}}_2, \dots, \overleftrightarrow{\mathcal{A}}_{d_{max}}\}$;

Output3: Coordinates $\alpha^j \in [0, 1]^{d_{max}}$ of the input BDTs in $B_{\mathbb{B}}$ (with $j \in \{1, 2, \dots, N\}$).

```
1:  $\mathcal{B}^* \leftarrow \text{WassersteinBarycenter}(\mathcal{S}_{\mathcal{B}})$ ;  
2: for  $d' \in \{1, 2, \dots, d_{max}\}$  do  
3:    $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{InitializeGeodesicAxis}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}})$ ;  
4:   while  $E_{W_2^{\mathcal{T}}}(B_{\mathbb{B}})$  decreases do  
5:     // Optimize the current geodesic axis  $\overleftrightarrow{\mathcal{A}}_{d'}$   
6:      $\alpha^{j \in \{1, 2, \dots, N\}} \leftarrow \text{ProjectTrees}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$ ;  
7:      $E_{W_2^{\mathcal{T}}}(B_{\mathbb{B}}) \leftarrow \text{EvaluateFittingEnergy}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$ ;  
8:      $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{OptimizeFittingEnergy}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$ ;  
9:  
10:  
11:  
12:  
13:  
14:  
15:   end while  
16: end for
```

Overview

Algorithm 1 Merge Tree Principal Geodesic Analysis (MT-PGA) Algorithm

Input: Set of BDTs $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$.

Output1: Basis origin \mathcal{B}^* ;

Output2: Basis geodesic axes $B_{\mathbb{B}} = \{\overleftrightarrow{\mathcal{A}}_1, \overleftrightarrow{\mathcal{A}}_2, \dots, \overleftrightarrow{\mathcal{A}}_{d_{max}}\}$;

Output3: Coordinates $\alpha^j \in [0, 1]^{d_{max}}$ of the input BDTs in $B_{\mathbb{B}}$ (with $j \in \{1, 2, \dots, N\}$).

```
1:  $\mathcal{B}^* \leftarrow \text{WassersteinBarycenter}(\mathcal{S}_{\mathcal{B}})$ ;  
2: for  $d' \in \{1, 2, \dots, d_{max}\}$  do  
3:    $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{InitializeGeodesicAxis}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}})$ ;  
4:   while  $E_{W_2^{\mathcal{T}}}(B_{\mathbb{B}})$  decreases do  
5:     // Optimize the current geodesic axis  $\overleftrightarrow{\mathcal{A}}_{d'}$   
6:      $\alpha^{j \in \{1, 2, \dots, N\}} \leftarrow \text{ProjectTrees}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$ ;  
7:      $E_{W_2^{\mathcal{T}}}(B_{\mathbb{B}}) \leftarrow \text{EvaluateFittingEnergy}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$ ;  
8:      $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{OptimizeFittingEnergy}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$ ;  
9:     // Enforce the constraints  
10:    while  $\overleftrightarrow{\mathcal{A}}_{d'}$  evolves do  
11:        
12:        
13:        
14:    end while  
15:  end while  
16: end for
```

Overview

Algorithm 1 Merge Tree Principal Geodesic Analysis (MT-PGA) Algorithm

Input: Set of BDTs $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$.

Output1: Basis origin \mathcal{B}^* ;

Output2: Basis geodesic axes $B_{\mathbb{B}} = \{\overleftrightarrow{\mathcal{A}}_1, \overleftrightarrow{\mathcal{A}}_2, \dots, \overleftrightarrow{\mathcal{A}}_{d_{\max}}\}$;

Output3: Coordinates $\alpha^j \in [0, 1]^{d_{\max}}$ of the input BDTs in $B_{\mathbb{B}}$ (with $j \in \{1, 2, \dots, N\}$).

```
1:  $\mathcal{B}^* \leftarrow \text{WassersteinBarycenter}(\mathcal{S}_{\mathcal{B}})$ ;  
2: for  $d' \in \{1, 2, \dots, d_{\max}\}$  do  
3:    $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{InitializeGeodesicAxis}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}})$ ;  
4:   while  $E_{W_2\mathcal{T}}(B_{\mathbb{B}})$  decreases do  
5:     // Optimize the current geodesic axis  $\overleftrightarrow{\mathcal{A}}_{d'}$   
6:      $\alpha^{j \in \{1, 2, \dots, N\}} \leftarrow \text{ProjectTrees}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$ ;  
7:      $E_{W_2\mathcal{T}}(B_{\mathbb{B}}) \leftarrow \text{EvaluateFittingEnergy}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$ ;  
8:      $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{OptimizeFittingEnergy}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$ ;  
9:     // Enforce the constraints  
10:    while  $\overleftrightarrow{\mathcal{A}}_{d'}$  evolves do  
11:       $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{EnforceGeodesics}(\mathcal{B}^*, \overleftrightarrow{\mathcal{A}}_{d'})$ ;  
12:      
13:      
14:    end while  
15:  end while  
16: end for
```

Overview

Algorithm 1 Merge Tree Principal Geodesic Analysis (MT-PGA) Algorithm

Input: Set of BDTs $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$.

Output1: Basis origin \mathcal{B}^* ;

Output2: Basis geodesic axes $B_{\mathbb{B}} = \{\overleftrightarrow{\mathcal{A}}_1, \overleftrightarrow{\mathcal{A}}_2, \dots, \overleftrightarrow{\mathcal{A}}_{d_{max}}\}$;

Output3: Coordinates $\alpha^j \in [0, 1]^{d_{max}}$ of the input BDTs in $B_{\mathbb{B}}$ (with $j \in \{1, 2, \dots, N\}$).

- 1: $\mathcal{B}^* \leftarrow \text{WassersteinBarycenter}(\mathcal{S}_{\mathcal{B}})$;
 - 2: **for** $d' \in \{1, 2, \dots, d_{max}\}$ **do**
 - 3: $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{InitializeGeodesicAxis}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}})$;
 - 4: **while** $E_{W_2\mathcal{T}}(B_{\mathbb{B}})$ decreases **do**
 - 5: // Optimize the current geodesic axis $\overleftrightarrow{\mathcal{A}}_{d'}$
 - 6: $\alpha^{j \in \{1, 2, \dots, N\}} \leftarrow \text{ProjectTrees}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$;
 - 7: $E_{W_2\mathcal{T}}(B_{\mathbb{B}}) \leftarrow \text{EvaluateFittingEnergy}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$;
 - 8: $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{OptimizeFittingEnergy}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$;
 - 9: // Enforce the constraints
 - 10: **while** $\overleftrightarrow{\mathcal{A}}_{d'}$ evolves **do**
 - 11: $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{EnforceGeodesics}(\mathcal{B}^*, \overleftrightarrow{\mathcal{A}}_{d'})$;
 - 12: $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{EnforceNegativeCollinearity}(\overleftrightarrow{\mathcal{A}}_{d'})$;
 - 13:
 - 14: **end while**
 - 15: **end while**
 - 16: **end for**
-

Overview

Algorithm 1 Merge Tree Principal Geodesic Analysis (MT-PGA) Algorithm

Input: Set of BDTs $\mathcal{S}_{\mathcal{B}} = \{\mathcal{B}(f_1), \dots, \mathcal{B}(f_N)\}$.

Output1: Basis origin \mathcal{B}^* ;

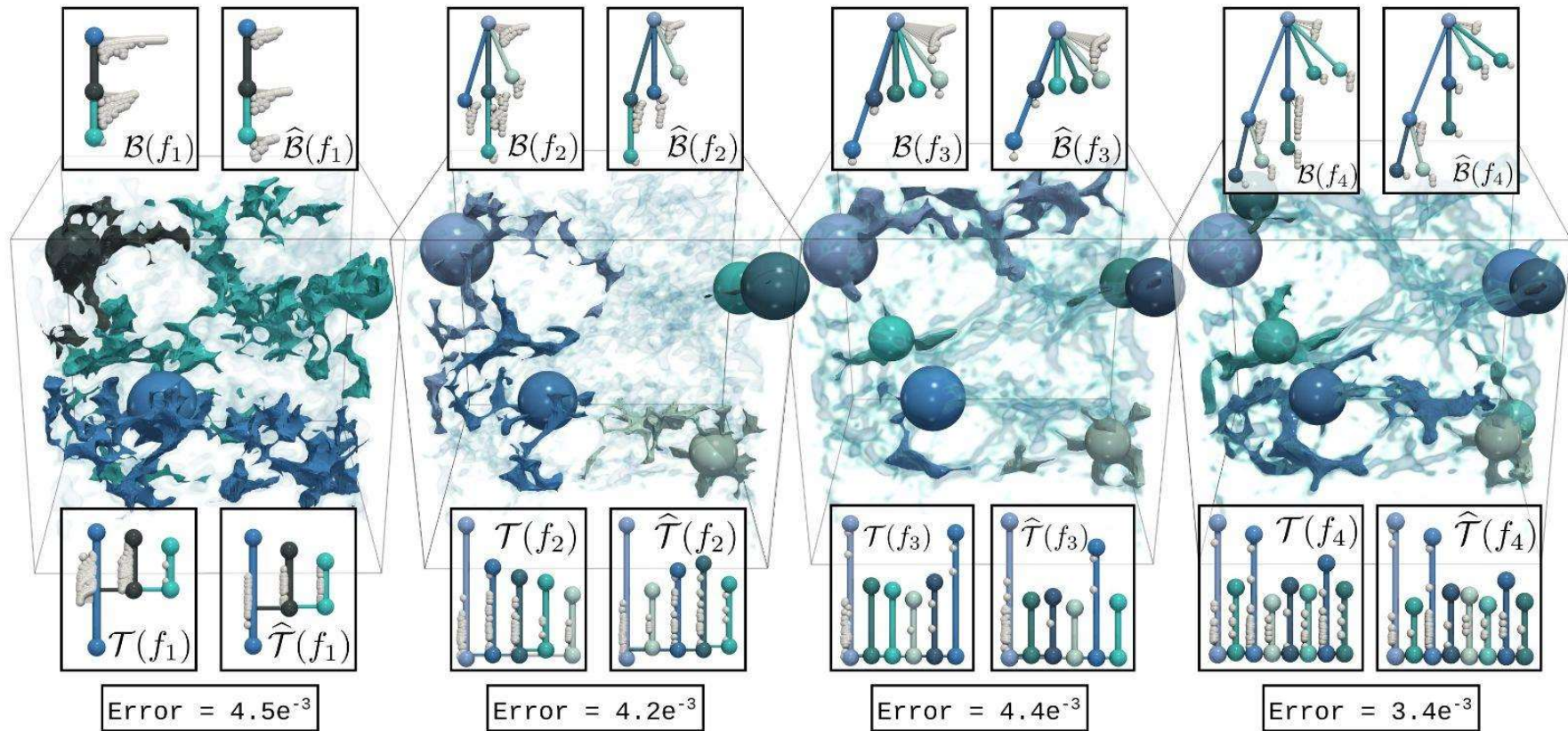
Output2: Basis geodesic axes $B_{\mathbb{B}} = \{\overleftrightarrow{\mathcal{A}}_1, \overleftrightarrow{\mathcal{A}}_2, \dots, \overleftrightarrow{\mathcal{A}}_{d_{\max}}\}$;

Output3: Coordinates $\alpha^j \in [0, 1]^{d_{\max}}$ of the input BDTs in $B_{\mathbb{B}}$ (with $j \in \{1, 2, \dots, N\}$).

- 1: $\mathcal{B}^* \leftarrow \text{WassersteinBarycenter}(\mathcal{S}_{\mathcal{B}})$;
 - 2: **for** $d' \in \{1, 2, \dots, d_{\max}\}$ **do**
 - 3: $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{InitializeGeodesicAxis}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}})$;
 - 4: **while** $E_{W_2\mathcal{T}}(B_{\mathbb{B}})$ decreases **do**
 - 5: // Optimize the current geodesic axis $\overleftrightarrow{\mathcal{A}}_{d'}$
 - 6: $\alpha^{j \in \{1, 2, \dots, N\}} \leftarrow \text{ProjectTrees}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$;
 - 7: $E_{W_2\mathcal{T}}(B_{\mathbb{B}}) \leftarrow \text{EvaluateFittingEnergy}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$;
 - 8: $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{OptimizeFittingEnergy}(\mathcal{S}_{\mathcal{B}}, \mathcal{B}^*, B_{\mathbb{B}}, \alpha^{j \in \{1, 2, \dots, N\}})$;
 - 9: // Enforce the constraints
 - 10: **while** $\overleftrightarrow{\mathcal{A}}_{d'}$ evolves **do**
 - 11: $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{EnforceGeodesics}(\mathcal{B}^*, \overleftrightarrow{\mathcal{A}}_{d'})$;
 - 12: $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{EnforceNegativeCollinearity}(\overleftrightarrow{\mathcal{A}}_{d'})$;
 - 13: $\overleftrightarrow{\mathcal{A}}_{d'} \leftarrow \text{EnforceOrthogonality}(B_{\mathbb{B}}, \overleftrightarrow{\mathcal{A}}_{d'})$;
 - 14: **end while**
 - 15: **end while**
 - 16: **end for**
-

Data reduction

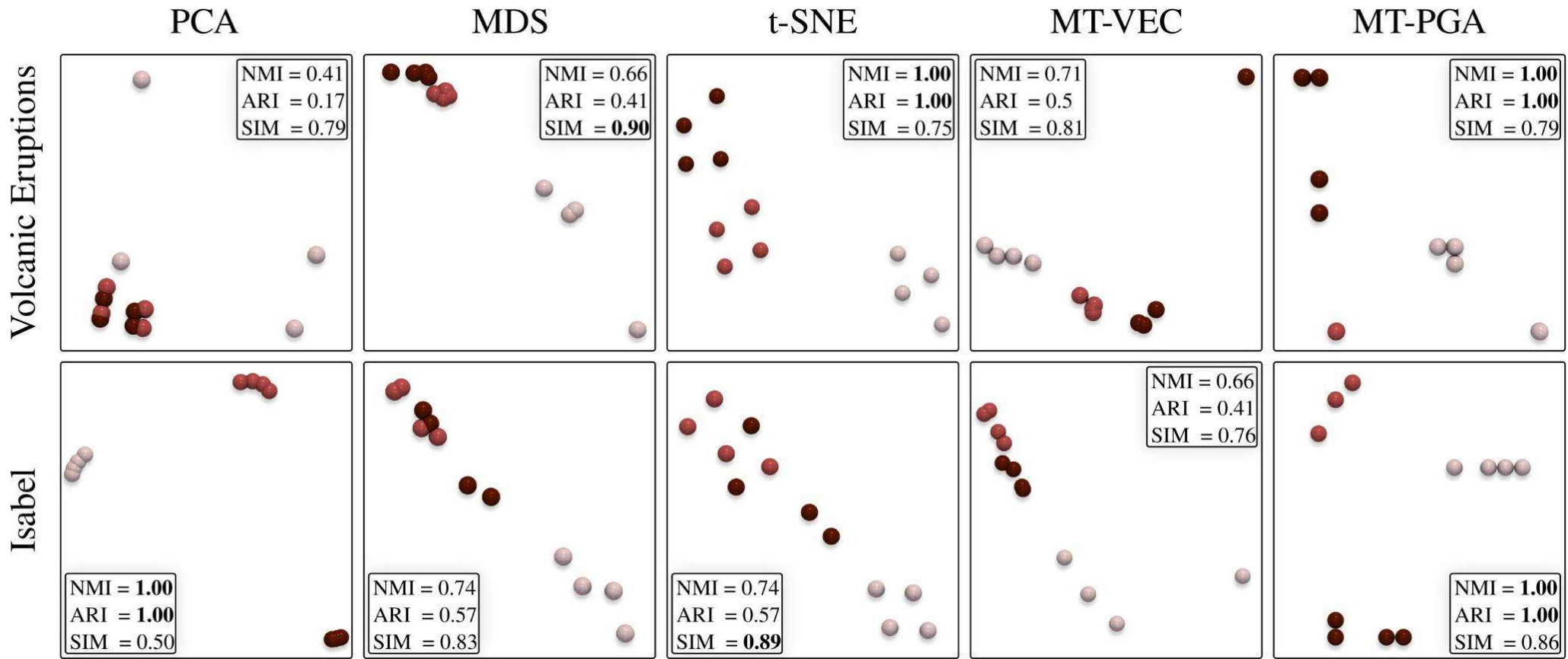
Compression factor: 19.27



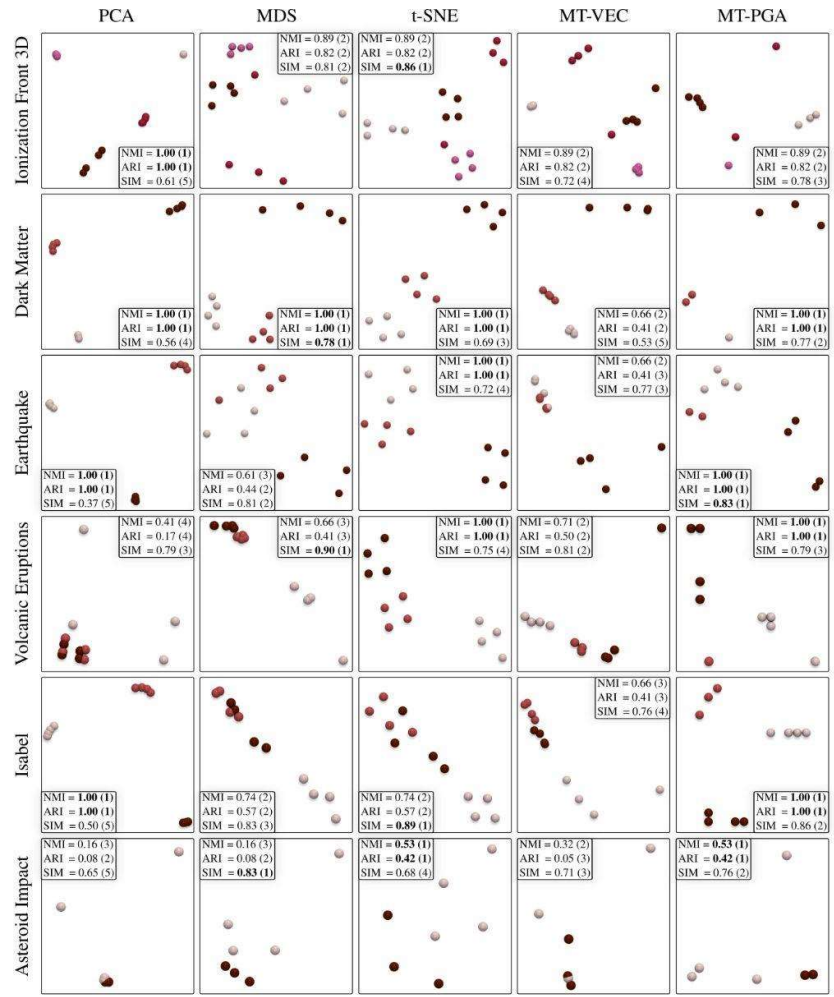
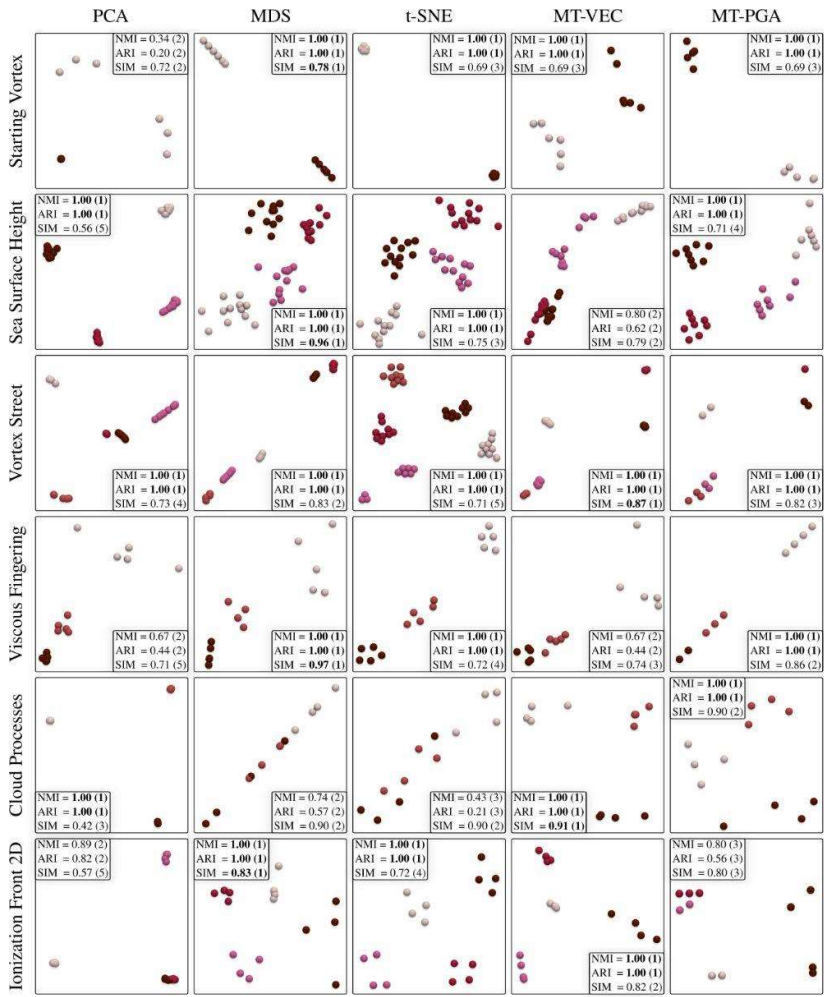
Data reduction

Dataset	N	$ \mathcal{B} $	PD-PGA		MT-PGA	
			Factor	Error	Factor	Error
Asteroid Impact (3D)	7	1,295	2.97	0.07	4.84	0.22
Cloud processes (2D)	12	1,209	5.94	0.19	7.39	0.01
Viscous fingering (3D)	15	118	2.23	0.13	4.71	0.02
Dark matter (3D)	40	2,592	10.00	0.04	19.27	0.04
Volcanic eruptions (2D)	12	811	9.99	0.12	4.83	0.04
Ionization front (2D)	16	135	2.56	0.14	5.12	0.40
Ionization front (3D)	16	763	3.27	0.17	4.85	0.46
Earthquake (3D)	12	1,203	1.42	0.18	2.19	0.33
Isabel (3D)	12	1,338	5.49	0.27	9.25	0.05
Starting Vortex (2D)	12	124	1.76	0.07	4.42	0.01
Sea Surface Height (2D)	48	1,787	19.59	0.18	9.48	0.48
Vortex Street (2D)	45	23	1.86	0.04	11.84	0.02

Dimensionality reduction



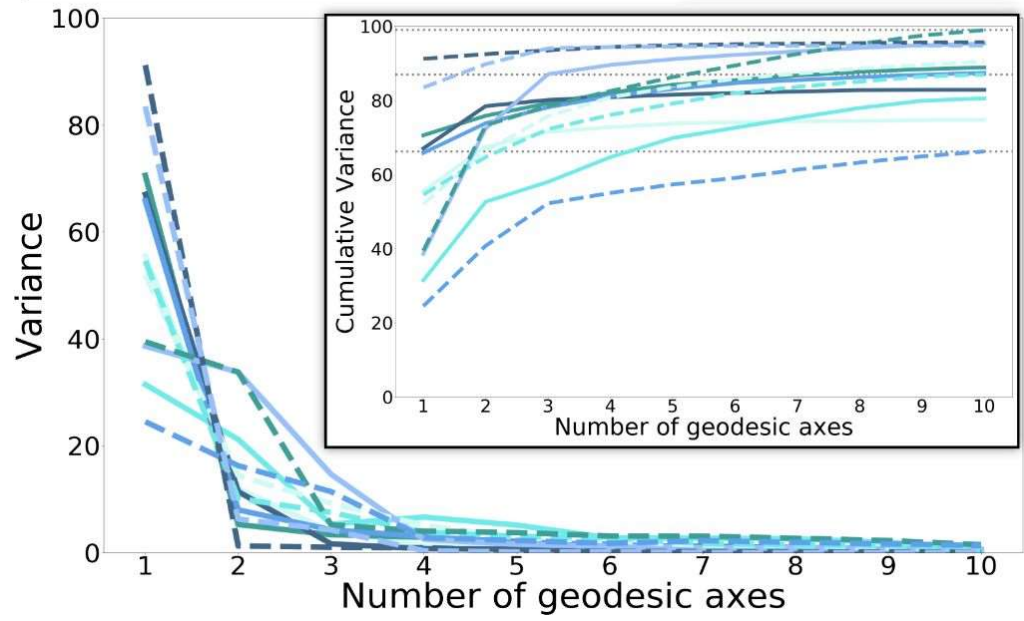
Dimensionality reduction



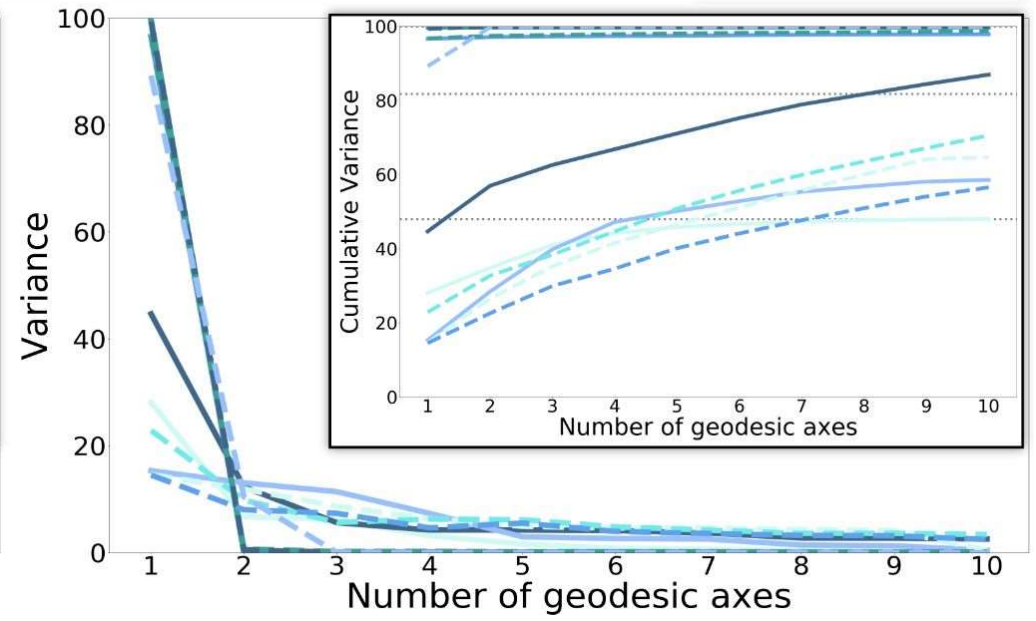
Dimensionality reduction

Indicator	PCA	MDS (W_2^T) [66]	t-SNE (W_2^T) [118]	MT-VEC	MT-PGA
NMI	0.79	0.82	0.88	0.78	0.94
ARI	0.71	0.73	0.84	0.64	0.90
SIM	0.60	0.85	0.75	0.76	0.80

Variance



PD-PGA



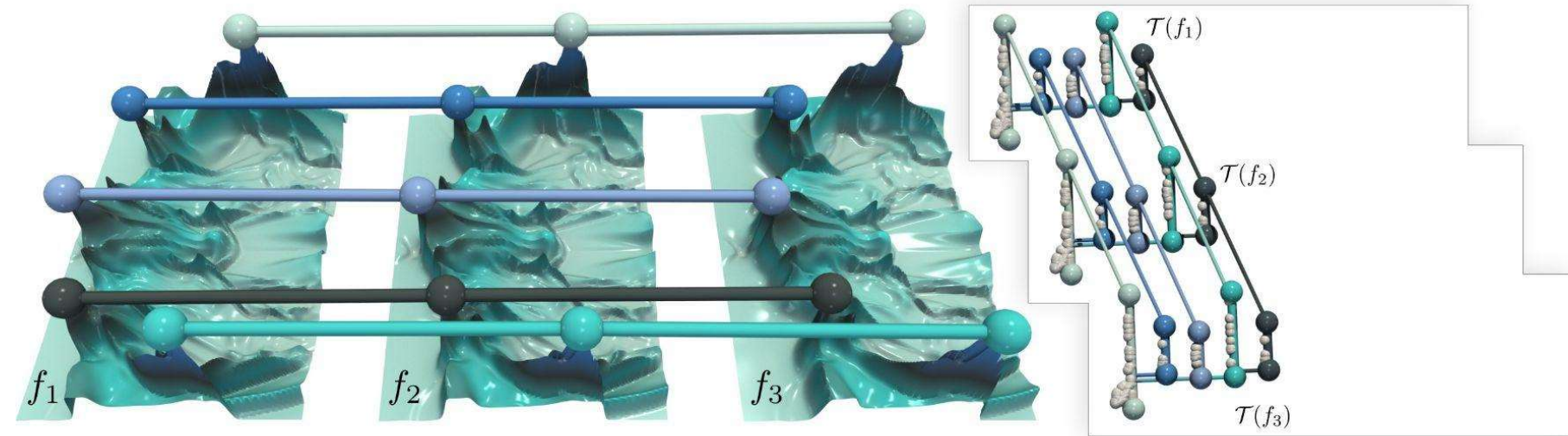
MT-PGA

Data reduction

- **Only store**
 - The basis
 - The coordinates

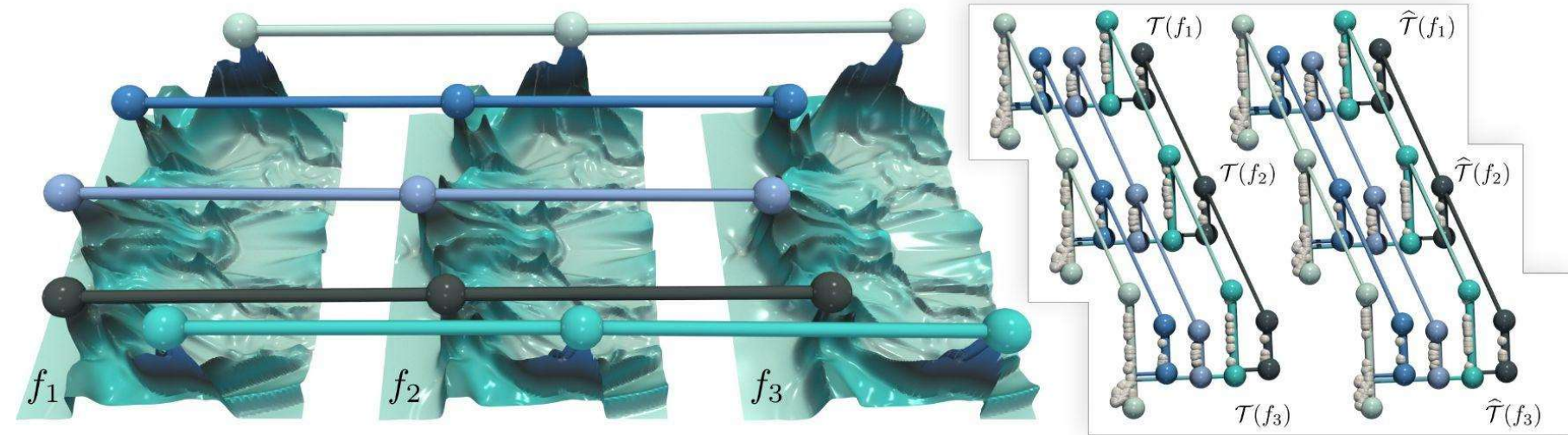
Data reduction

- **Only store**
 - The basis
 - The coordinates



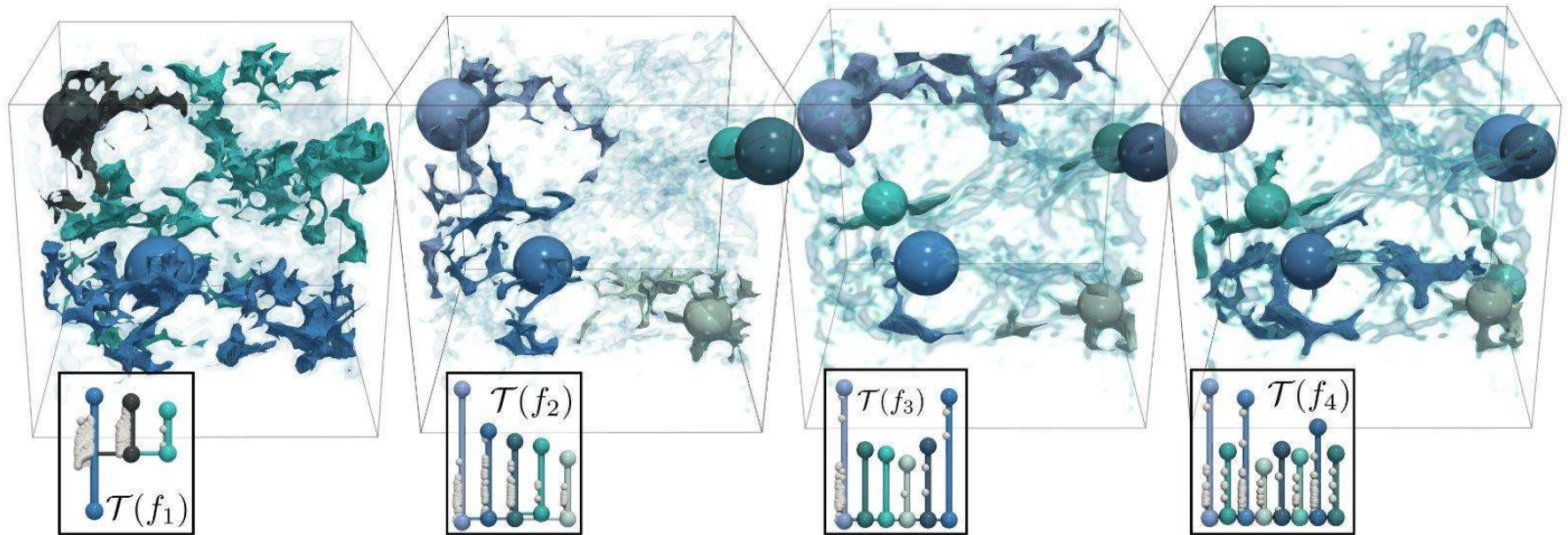
Data reduction

- Only store
 - The basis
 - The coordinates



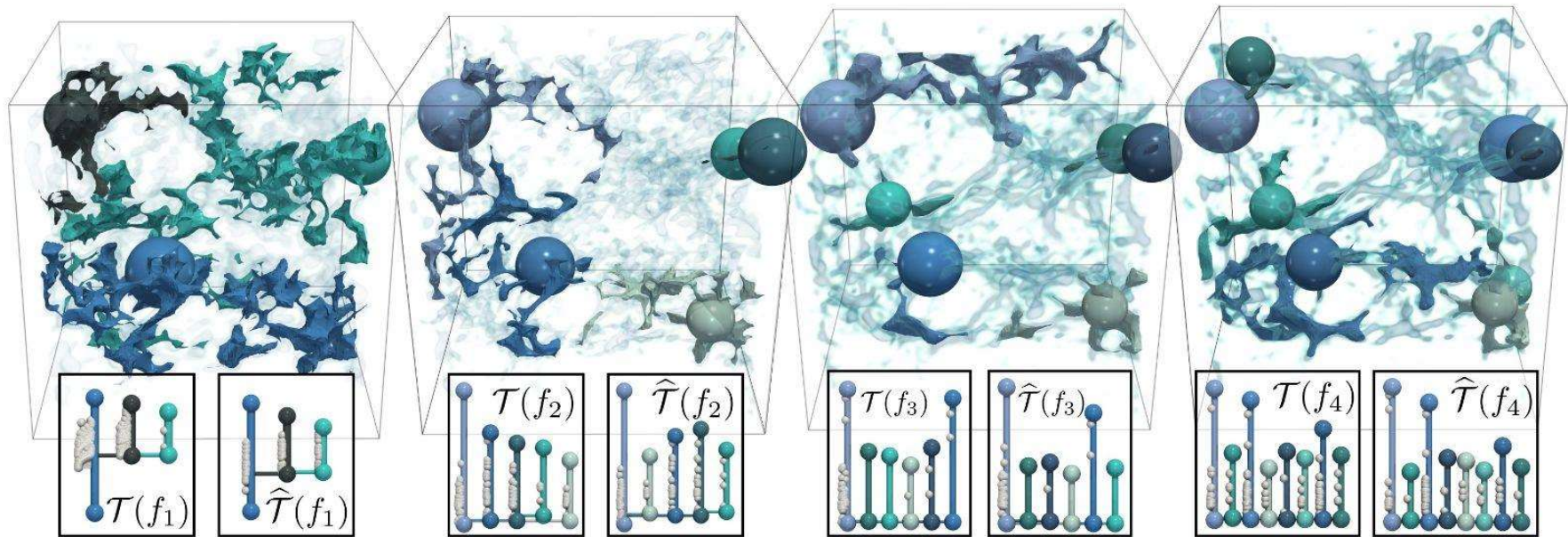
Compression factor: 5.12

Data reduction

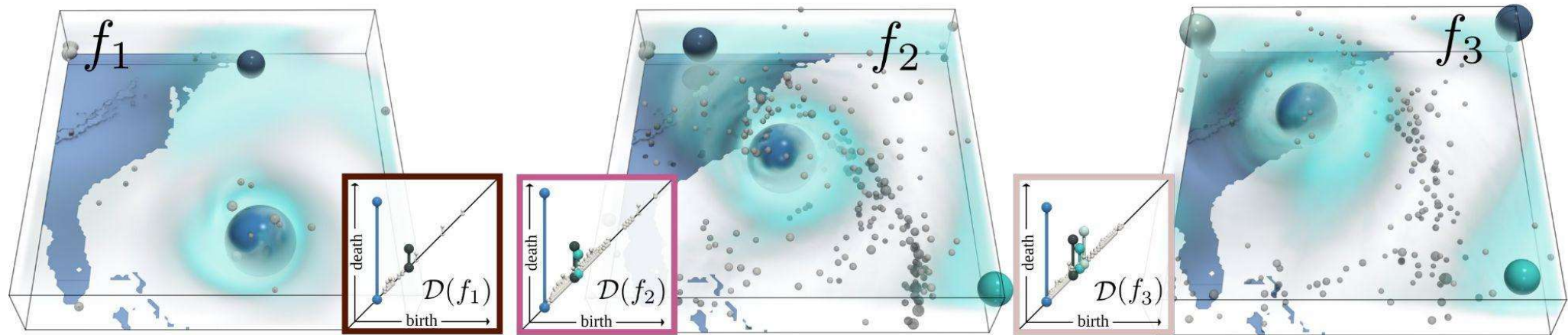


Data reduction

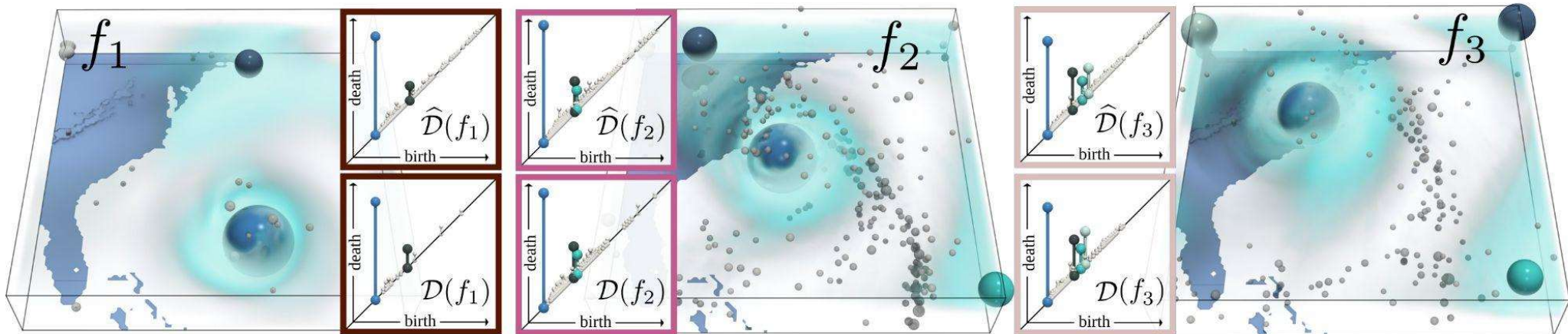
Compression factor: 19.27



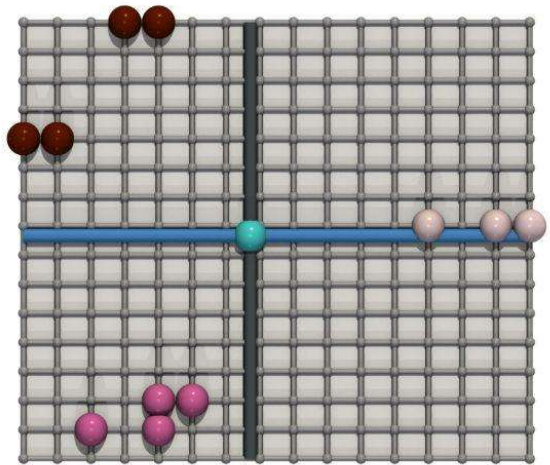
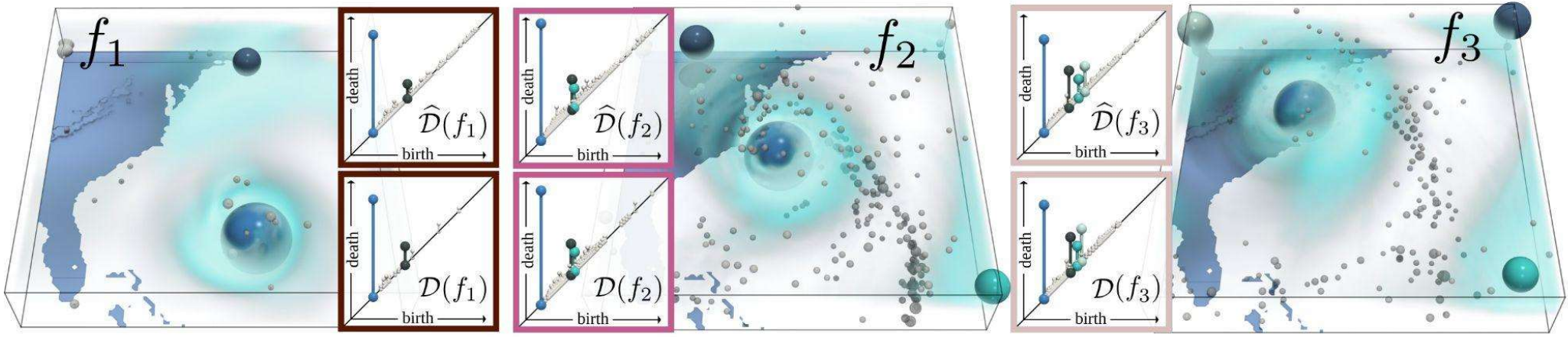
PD-PGA



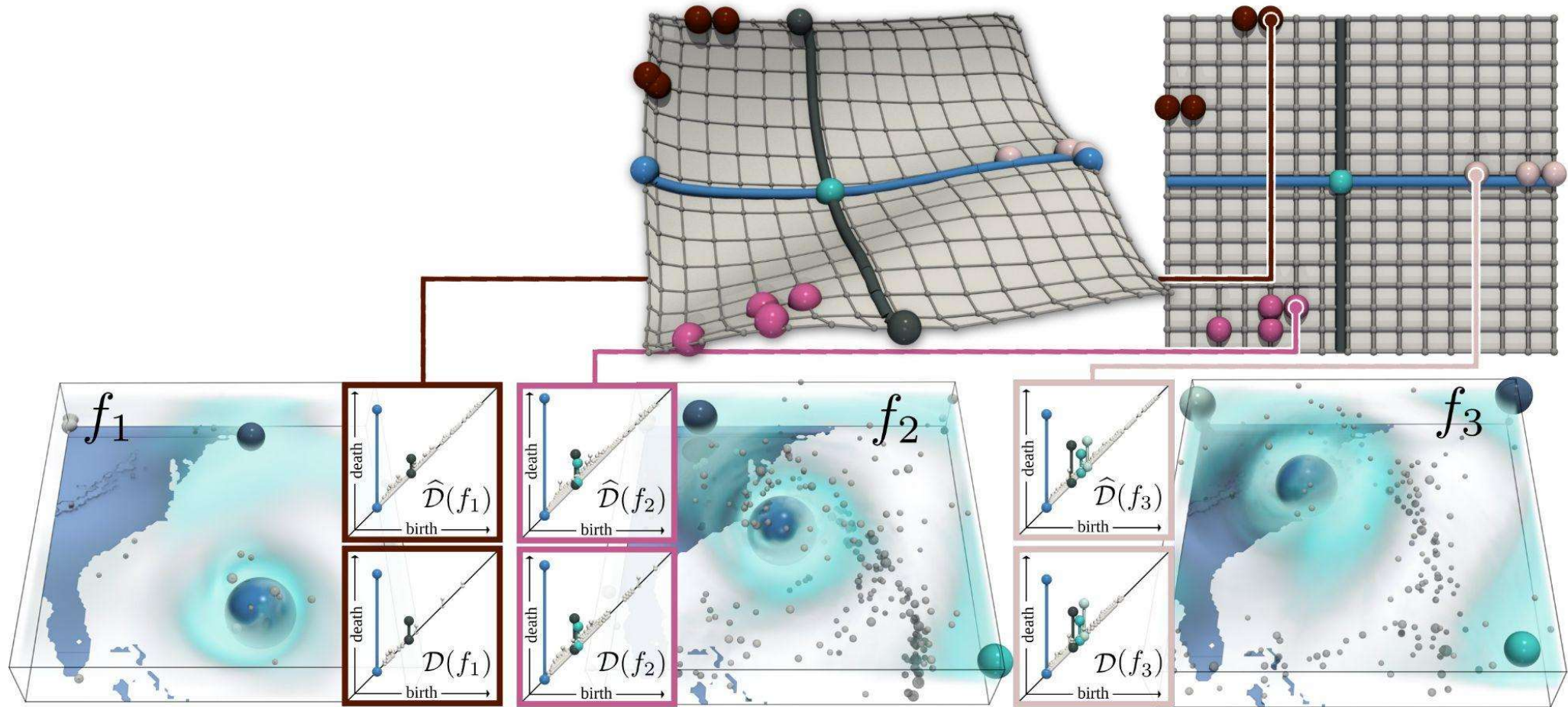
PD-PGA



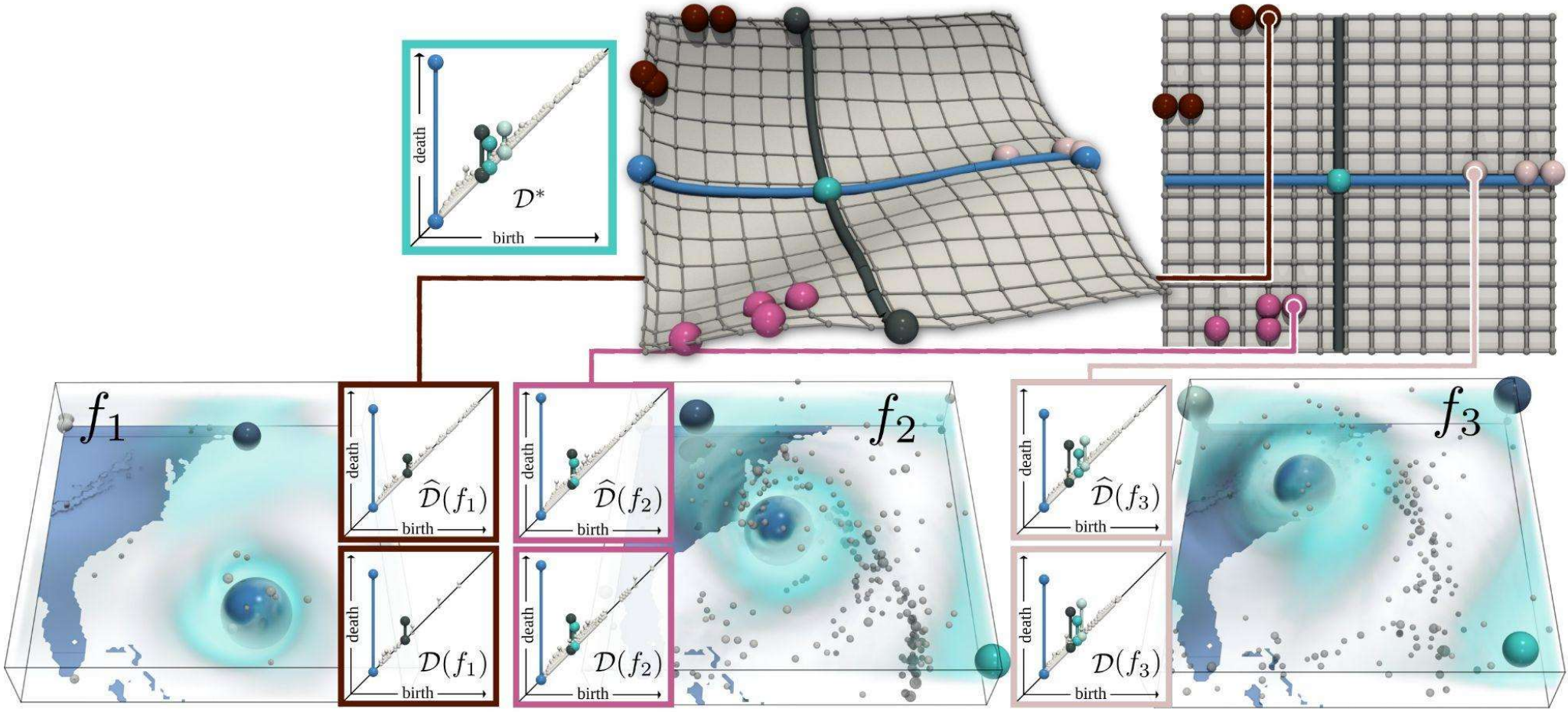
PD-PGA



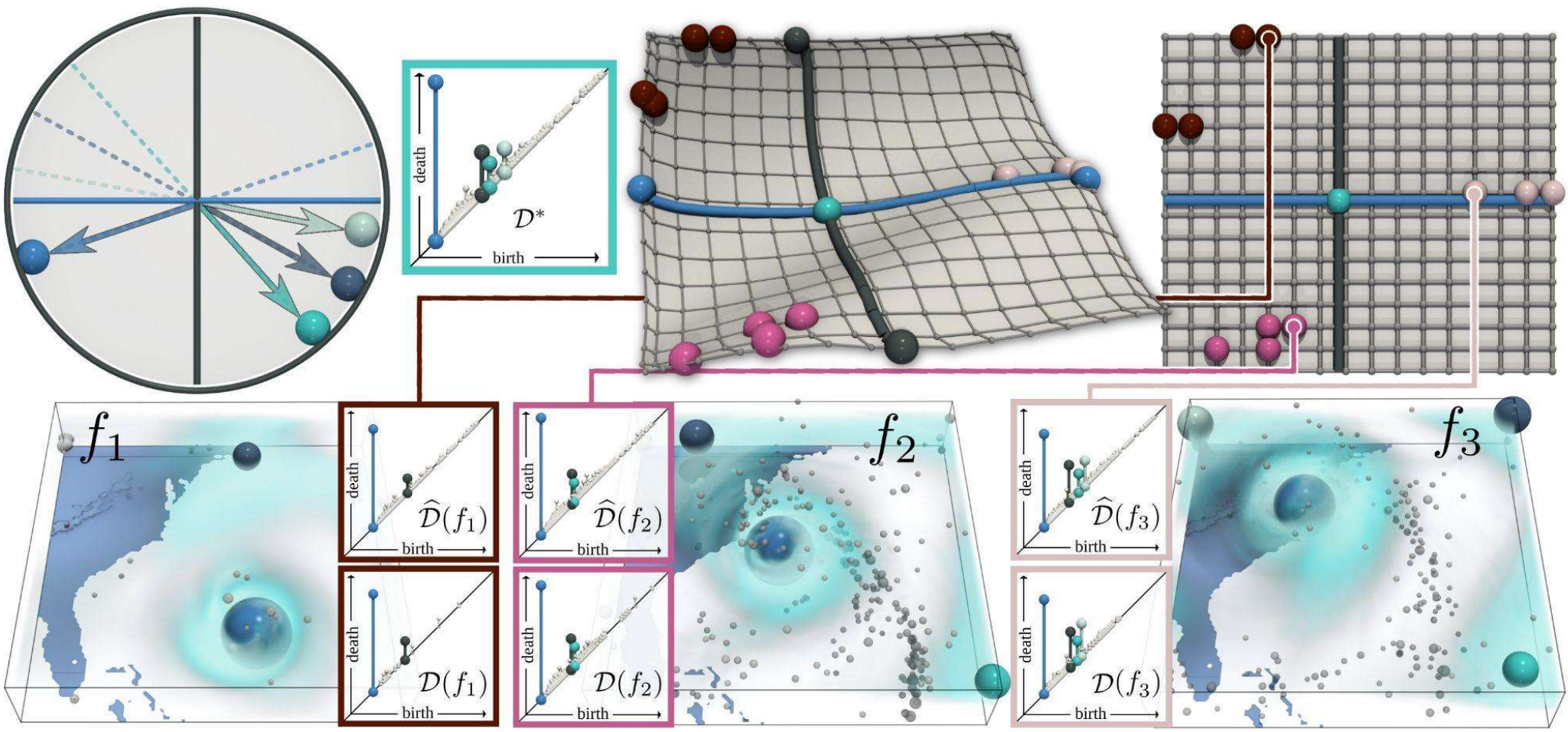
PD-PGA



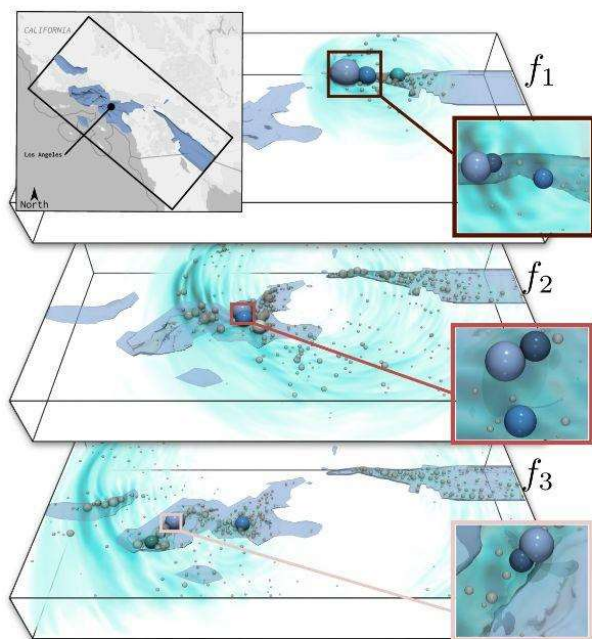
PD-PGA



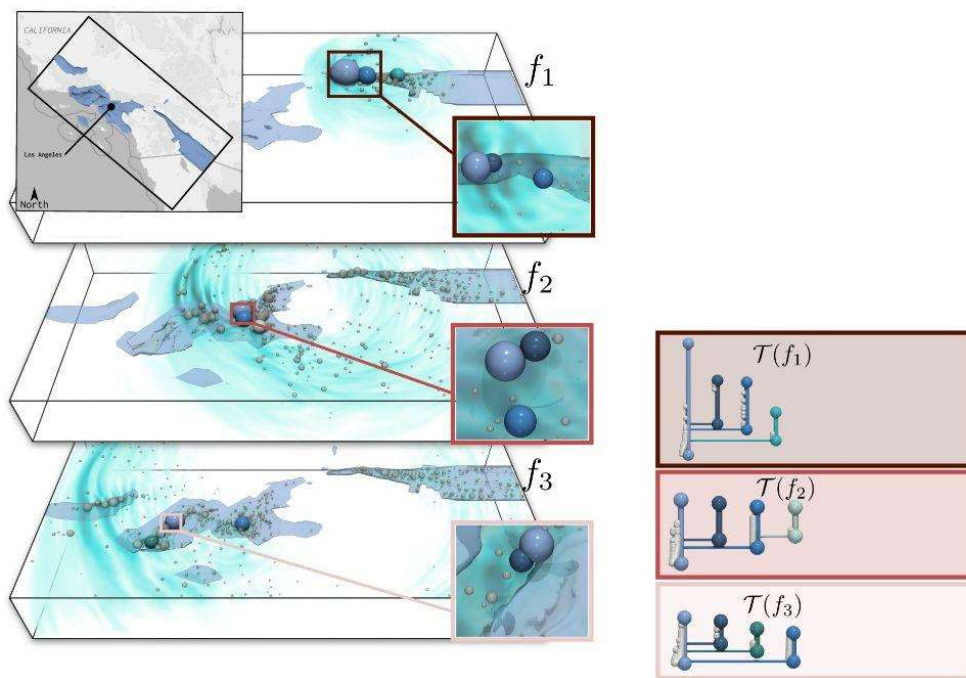
PD-PGA



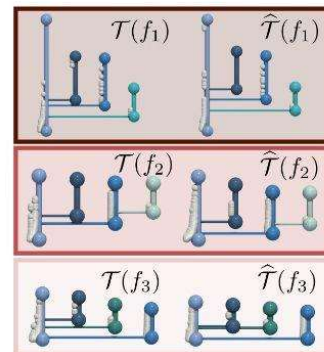
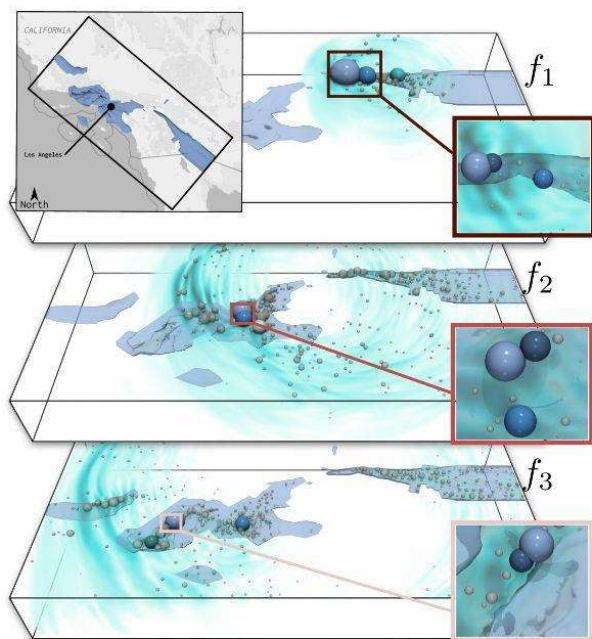
MT-PGA



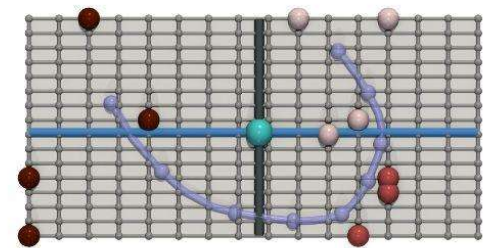
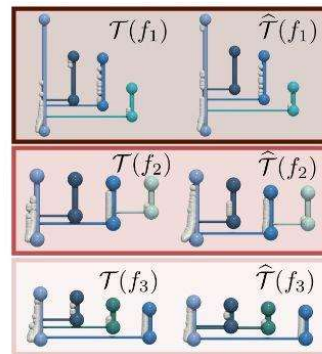
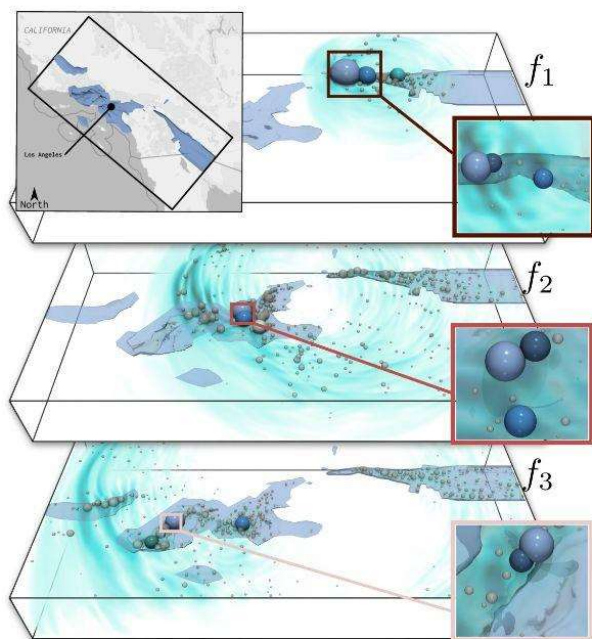
MT-PGA



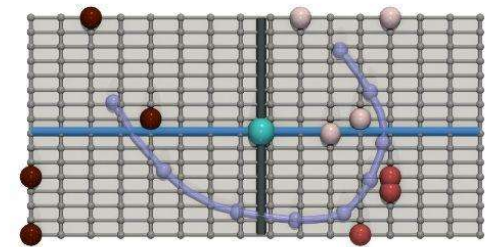
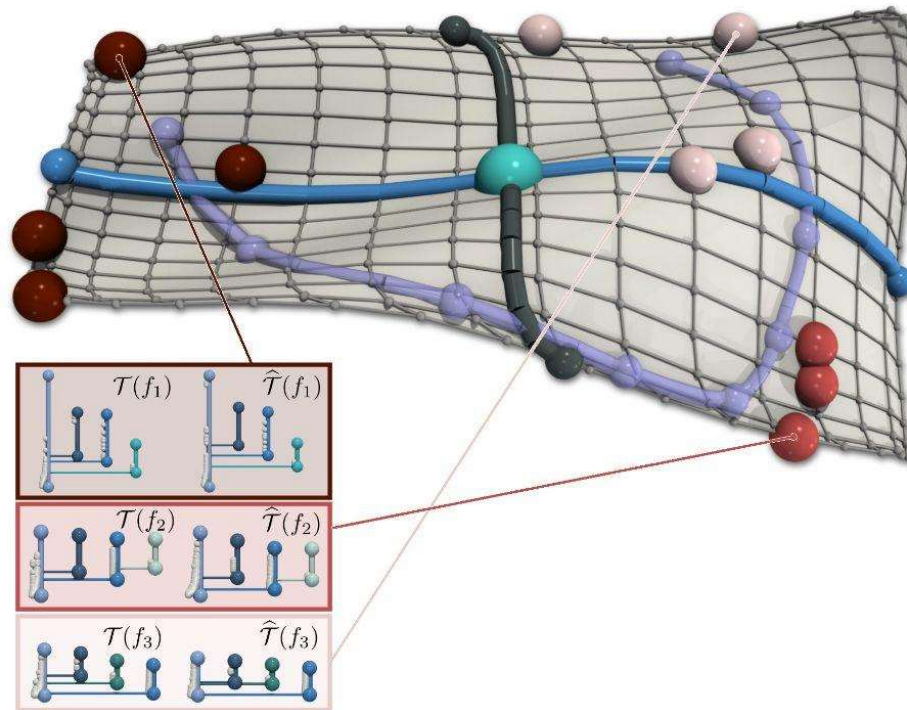
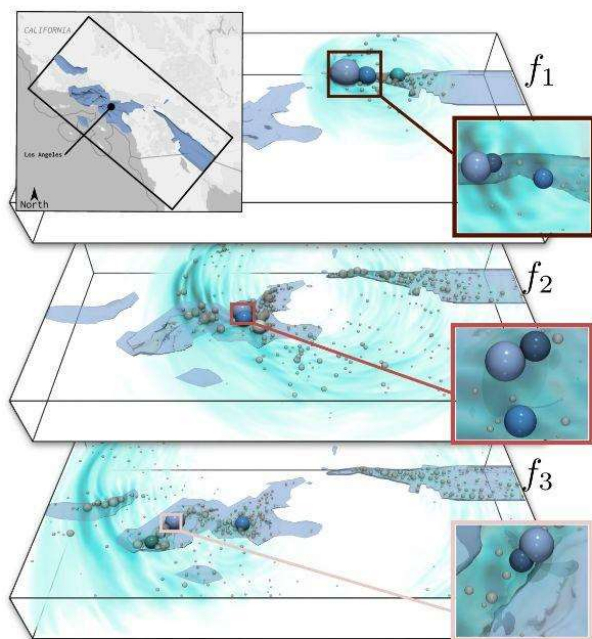
MT-PGA



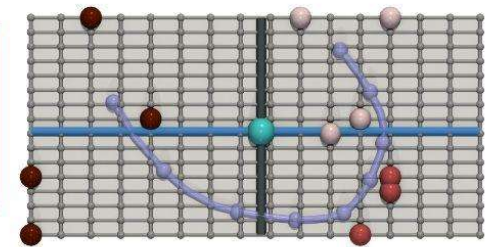
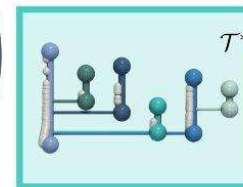
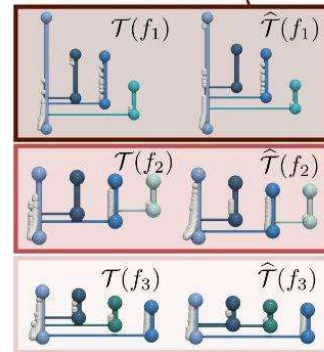
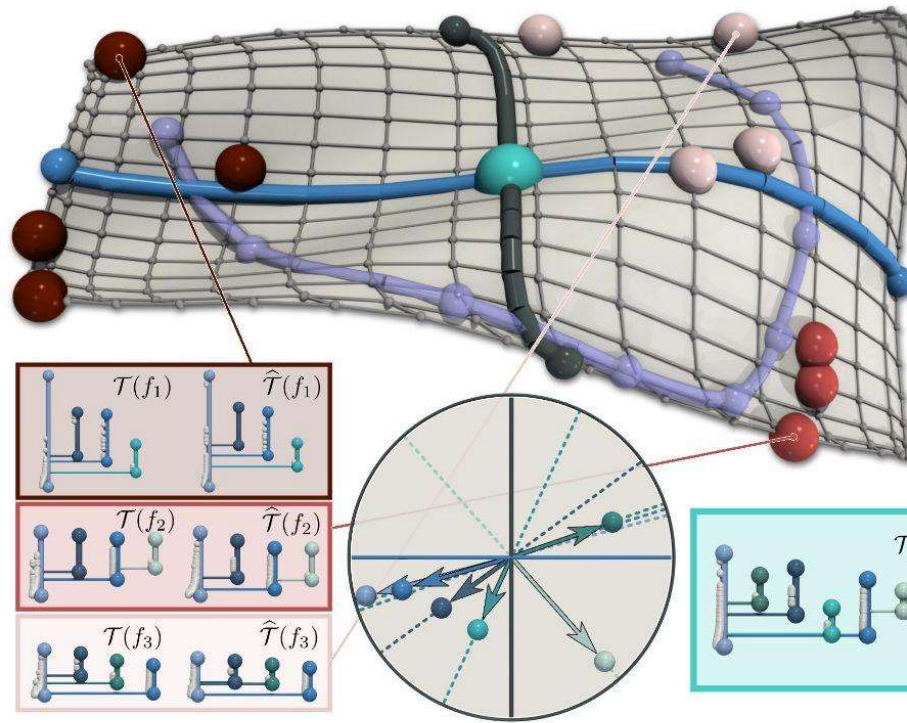
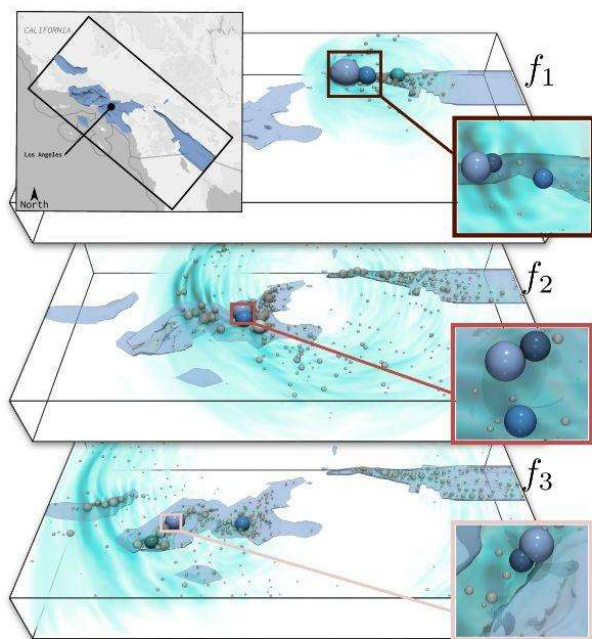
MT-PGA



MT-PGA



MT-PGA



MT-PGA

