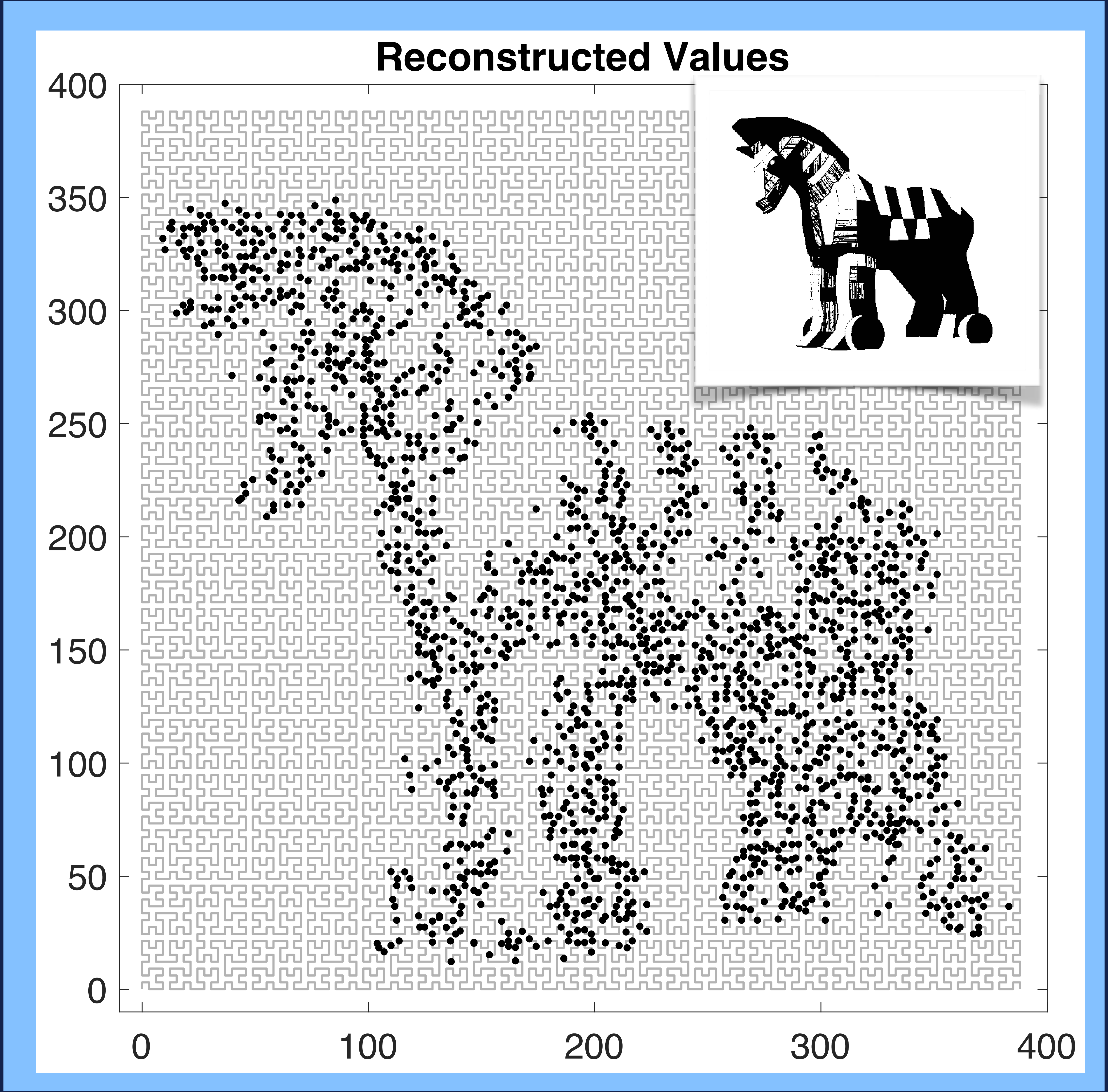
DATA RECOVERY ON ENCRYPTED DATABASES WITH k-NEAREST NEIGHBOR QUERY LEAKAGE

EVGENIOS M. KORNAROPOULOS
CHARALAMPOS PAPAMANTHOU
ROBERTO TAMASSIA

BROWN
UNIVERSITY OF MARYLAND



Full version: https://eprint.iacr.org/2018/719

INTRO
WHO CARES ABOUT k-NN?

COLUMN-ORIENTED DBMS

18.2.8. KNearestNeighborProcess

GeoMesa Documentation

geomesa

The `KNearestNeighborProcess` performs a K Nearest Neighbor search on a Geomesa feature collection using another feature collection as input. Return k neighbors for each point in the input data set. If a point is the nearest neighbor of multiple points of the input data set, it is returned only once.

Parameters	Description
inputFeatures	Input feature collection that defines the KNN search.
dataFeatures	The data set to query for matching features.
numDesired	K : number of nearest neighbors to return.
estimatedDistance	Estimate of Search Distance in meters for K neighbors—used to set the granularity of the search.
maxSearchDistance	Maximum search distance in meters—used to prevent runaway queries of the entire table.

18.2.8.1. K-Nearest-Neighbor example (XML)

KNNProcess_wps.xml is a geoserver WPS call to the GeoMesa KNearestNeighborProcess. It is here chained with a Query process (see Chaining Processes) in order to avoid points related to the same Id to be matched by the request. It can be run with the following curl call:

```
curl -v -u admin:geoserver -H "Content-Type: text/xml" -d@KNNProcess_wps.xml localhost:8080/geoserver/wp
```

INTRO
WHO CARES ABOUT k-NN?

COLUMN-ORIENTED DBMS

…rocess

…s a K Nearest Neighbor search on a Geomesa feature collection

GeoMesa Documentation

OBJECT-RELATIONAL DBMS

PostGIS — Spatial and Geographic objects for PostgreSQL

Home | Download | Documentation | Development | Support | OSGeo

27.2. Index-based KNN

"KNN" stands for "K nearest neighbours", where "K" is the number of neighbours you are looking for.

KNN is a pure index based nearest neighbour search. By walking up and down the index, the search can find the nearest candidate geometries without using any magical search radius numbers, so the technique is suitable and high performance even for very large tables with highly variable data densities.

> **Note**
>
> The KNN feature is only available on PostGIS 2.0 with PostgreSQL 9.1 or greater.

The KNN system works by evaluating distances between bounding boxes inside the PostGIS R-Tree index.

Because the index is built using the bounding boxes of geometries, the distances between any geometries that are not points will be inexact: they will be the distances between the bounding boxes of geometries.

The syntax of the index-based KNN query places a special "index-based distance operator" in the ORDER BY clause of the query, in this case "<->". There are two index-based distance operators,

- <-> means "distance between box centers"
- <#> means "distance between box edges"

One side of the index-based distance operator must be a literal geometry value. We can get away with a subquery that returns as single geometry, or we could include a *WKT* geometry instead.

```
-- Closest 10 streets to Broad Street station are ?
SELECT
  streets.gid,
  streets.name
FROM
```

INTRO
WHO CARES ABOUT k-NN?

COLUMN-ORIENTED DBMS

18.2. Processors
18.2.1. ArrowConversionProcess
18.2.2. BinConversionProcess

rocess

a K Nearest Neighbor search on a Geomesa feature collection

GeoMesa Documentation

OBJECT-RELATIONAL DBMS

PostGIS
Spatial and Geographic objects for PostgreSQL
Home | Download | Documentation | Development | Support | OSGeo

s", where "K" is the number of neighbours you are looking for.

ghbour search. By walking up and down the index, the
geometries without using any magical search radius
nd high performance even for very large tables with highly

PostGIS 2.0 with PostgreSQL 9.1 or greater.

distances between bounding boxes inside the PostGIS R-

unding boxes of geometries, the distances between any
exact: they will be the distances between the bounding

ery places a special "index-based distance operator" in the
case "<->". There are two index-based distance operators,

centers"
edges"

One side of the index-based distance operator must be a literal geometry value. We can get away with a subquery that returns as single geometry, or we could include a *WKT* geometry instead.

```
-- Closest 10 streets to Broad Street station are ?
SELECT
  streets.gid,
  streets.name
FROM
```

CLOUD SERVICES

IBM Cloud | Catalog | Docs

IBM Cloudant

LEARN
- Getting started tutorial
- Overview
- IBM Cloud Public
- Pricing
- Security and Compliance
- Release information
- Other offerings

HOW TO
- Tutorials
- Recovery and backup

Nearest neighbor search

IBM Cloudant Geo supports Nearest Neighbor search, which is known as NN search. If provided, the `nearest=true` search returns all results by sorting their distances to the center of the query geometry. This geometric relation `nearest=true` can be used either with all the geometric relations described earlier, or alone.

For example, one police officer might search five crimes that occurred near a specific location by typing the query in the following example.

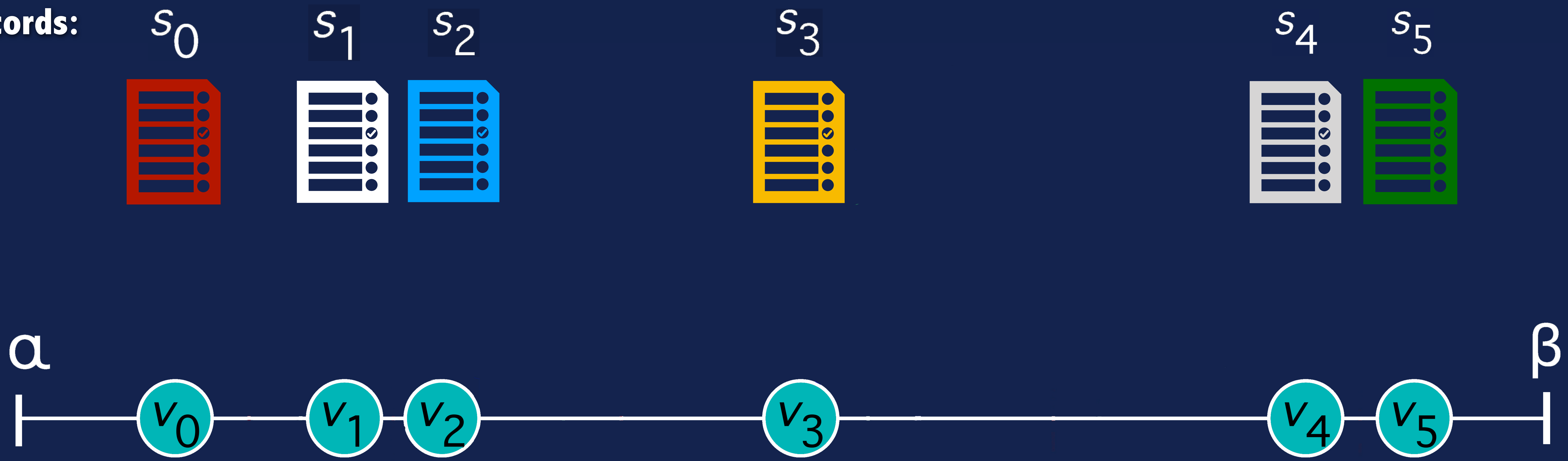
Example query to find nearest five crimes against a specific location:

```
https://education.cloudant.com/crimes/_design/geodd/_geo/geoidx?g=POINT(-71.053712
```

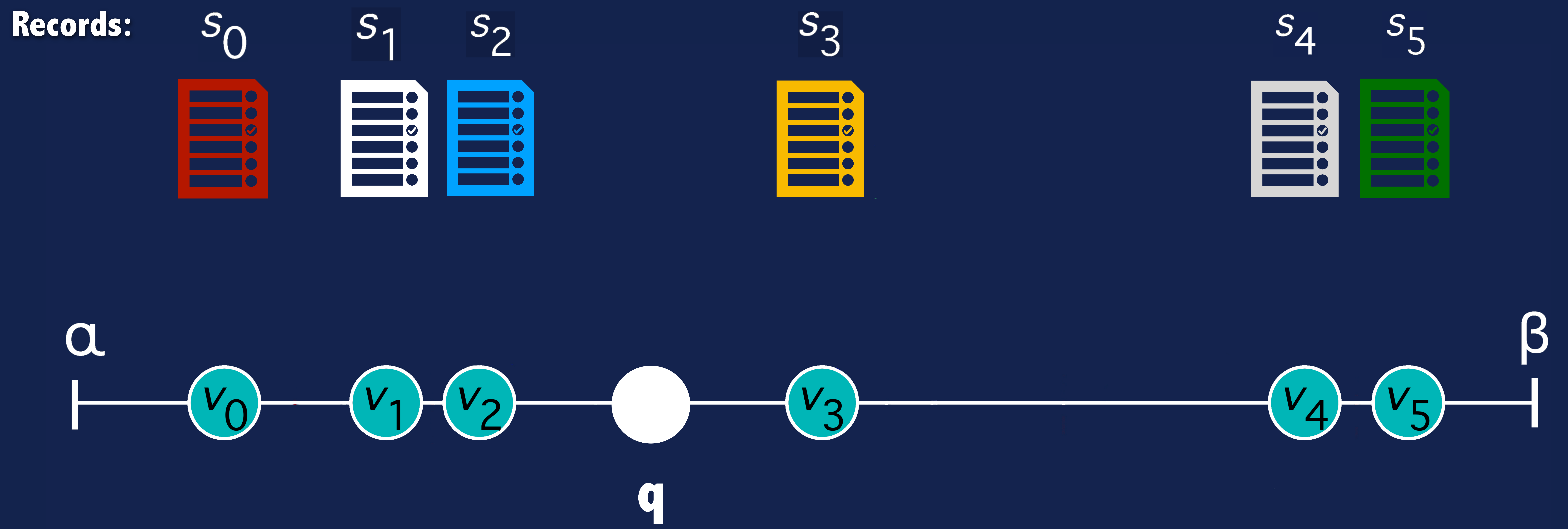
Tip: The `nearest=true` search can change the semantics of an IBM Cloudant Geo search. For example, without `nearest=true` in the example query, the results include only GeoJSON documents that have coordinates equal to the query point `(-71.0537124 42.3681995)` *or* an empty results set. However, by using the `nearest=true` search, the results include all GeoJSON documents in the database whose order is measured by the distance to the query point.

SETUP
k-NEAREST NEIGHBORS

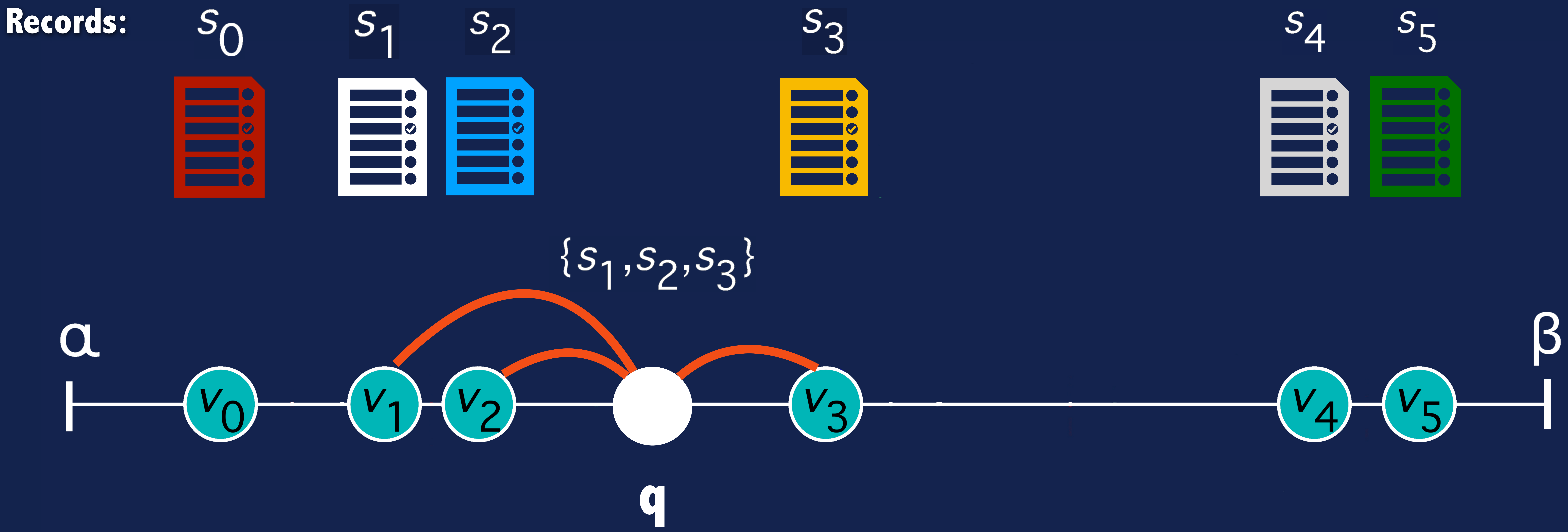
Records:



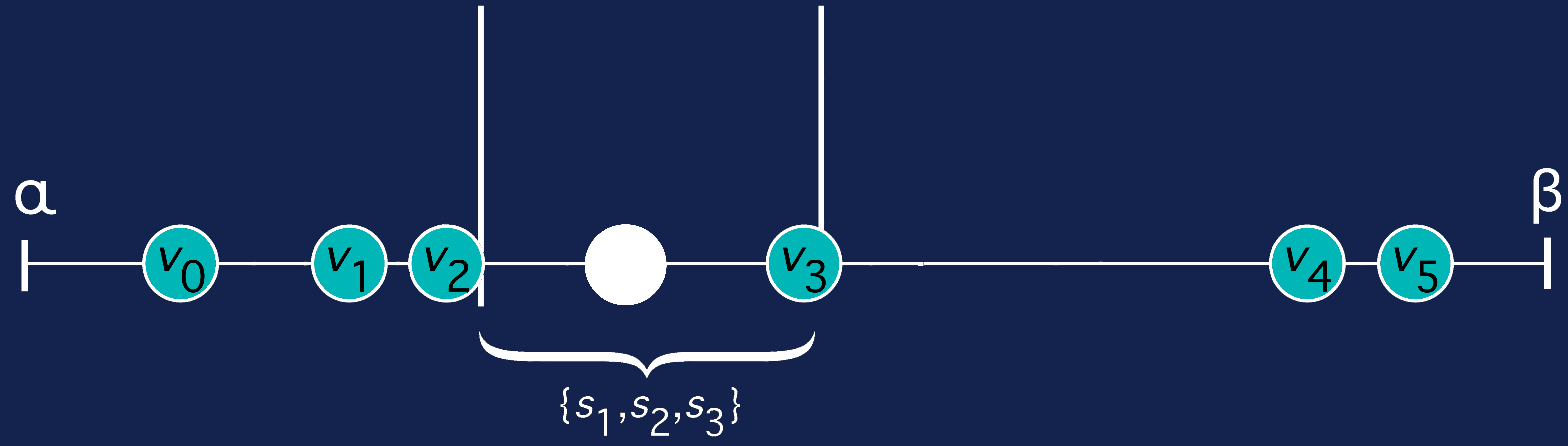
SETUP
k-NEAREST NEIGHBORS



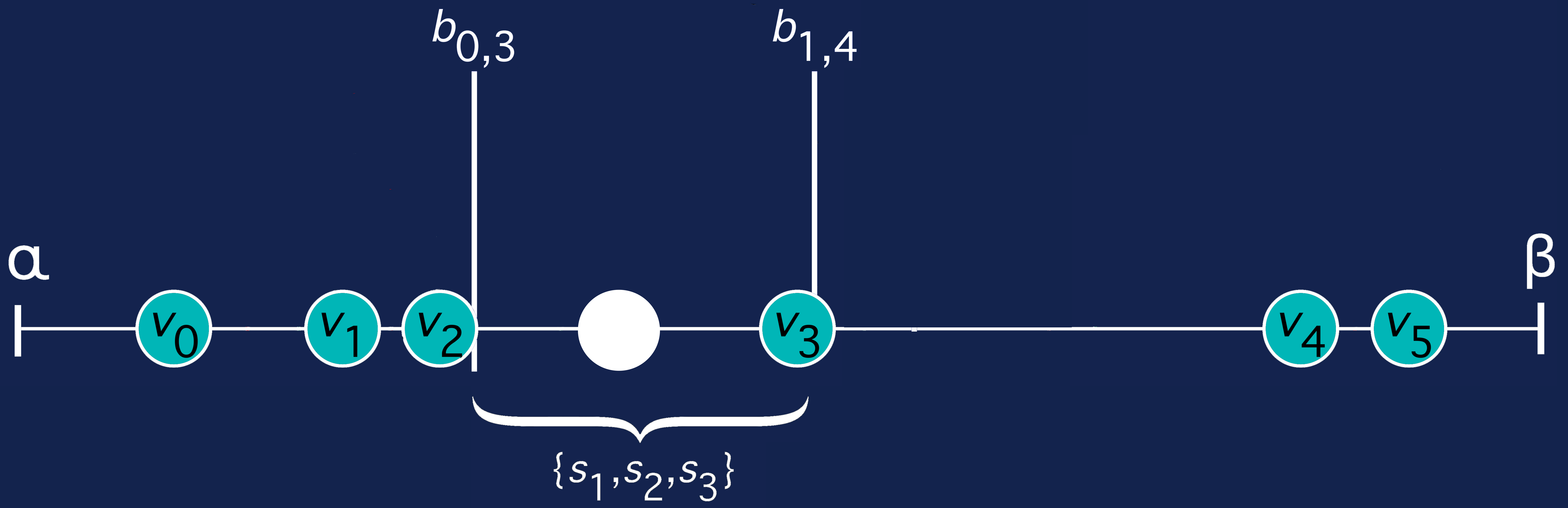
SETUP
k-NEAREST NEIGHBORS



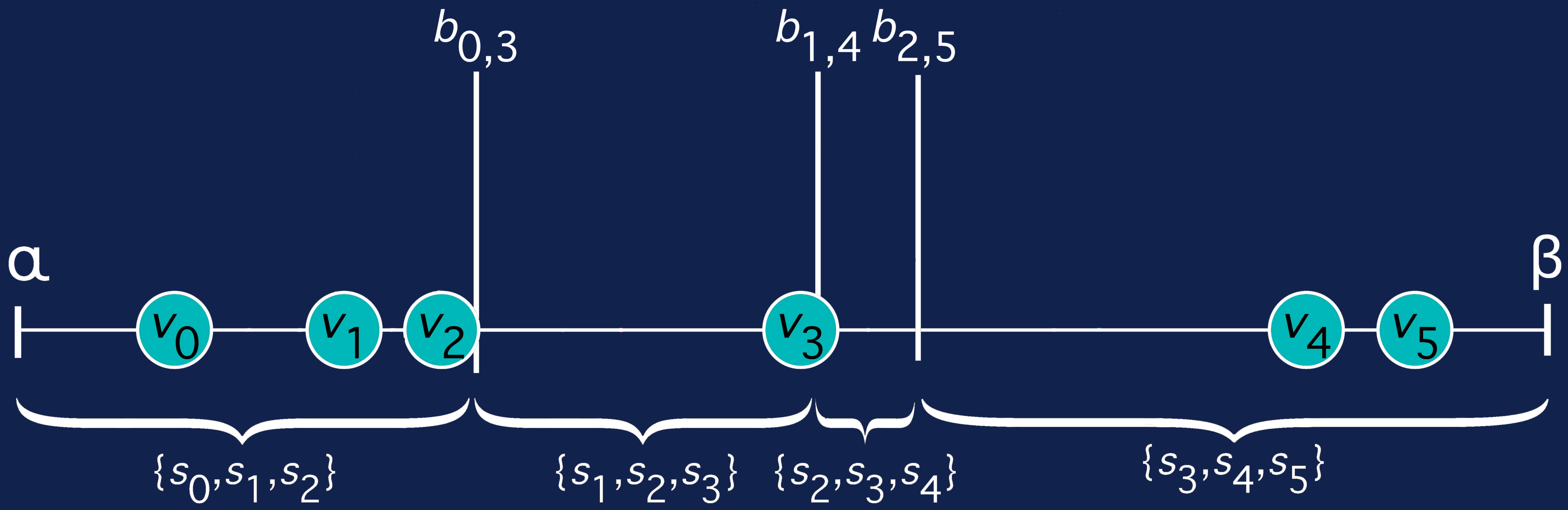
SETUP
VORONOI DIAGRAMS



SETUP
VORONOI DIAGRAMS



SETUP
VORONOI DIAGRAMS



Voronoi Diagram

SETUP
ENCRYPTED SEARCH

Client

Server

Tokens | **Responses**

$\mathrm{PRF}_K(\bullet) = t$

$\mathrm{PRF}_K(\bullet) = t'$

$\mathrm{PRF}_K(\bullet) = t''$

$\mathrm{PRF}_K(\bullet) = t$

SETUP

ENCRYPTED SEARCH

Client

Server

Tokens

$\mathrm{PRF}_K(\,\bullet\,) = t$

$\mathrm{PRF}_K(\,\bullet\,) = t'$

$\mathrm{PRF}_K(\,\bullet\,) = t''$

$\mathrm{PRF}_K(\,\bullet\,) = t$

Search Pattern Leakage

Responses

Access Pattern Leakage

k-NN EXACT RECONSTRUCTION

ORDERED RESPONSES: Possible when all encrypted queries are issued

UNORDERED RESPONSES: Impossible due to many reconstructions

k-NN APPROXIMATE RECONSTRUCTION

ORDERED RESPONSES: Approximate reconstruction when not all encrypted queries are issued

UNORDERED RESPONSES: Even with many reconstructions approximate with bounded error

k-NN EXACT RECONSTRUCTION

ORDERED RESPONSES: Possible when all encrypted queries are issued

UNORDERED RESPONSES: Impossible due to many reconstructions

k-NN APPROXIMATE RECONSTRUCTION

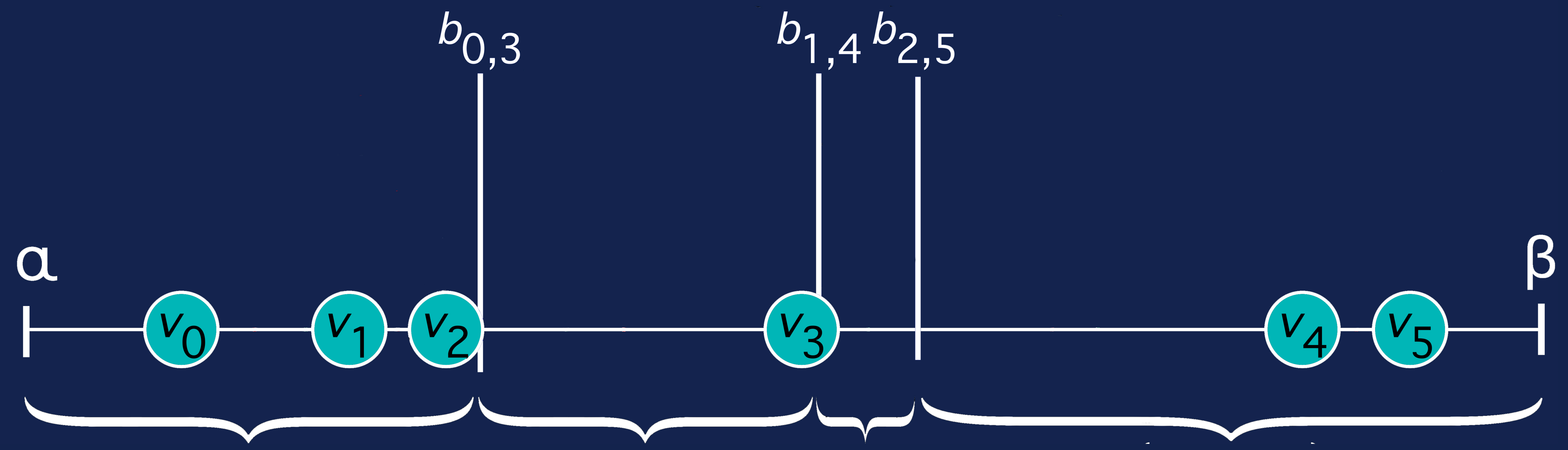
ORDERED RESPONSES: Approximate reconstruction when not all encrypted queries are issued

UNORDERED RESPONSES: Even with many reconstructions approximate with bounded error

UNORDERED RESPONSES
ASSUMPTIONS OF THE ATTACK

BOUNDARIES:

Known boundaries α and β



STATIC:

No updates in the database

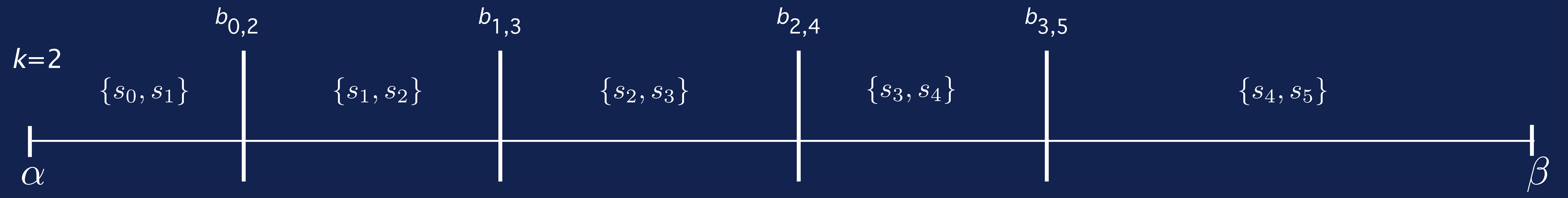
UNIFORMITY:

Queries are generated uniformly at random from [α,β]

UNORDERED RESPONSES
EXACT RECONSTRUCTION

Impossible to achieve Exact Reconstruction

Best Case Scenario for the Adversary

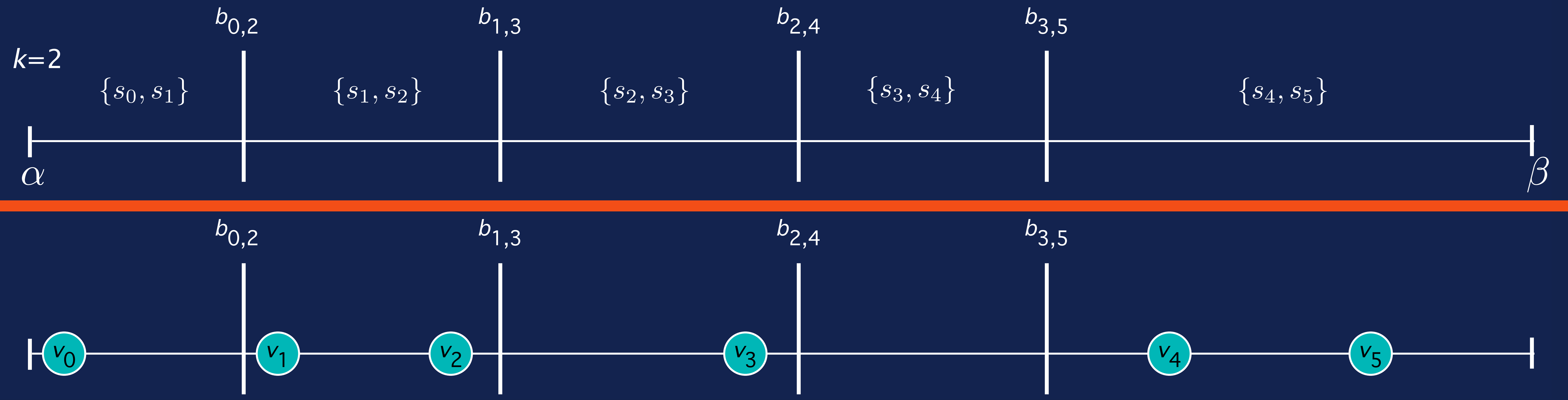


UNORDERED RESPONSES
EXACT RECONSTRUCTION

Impossible to achieve Exact Reconstruction

Best Case Scenario for the Adversary

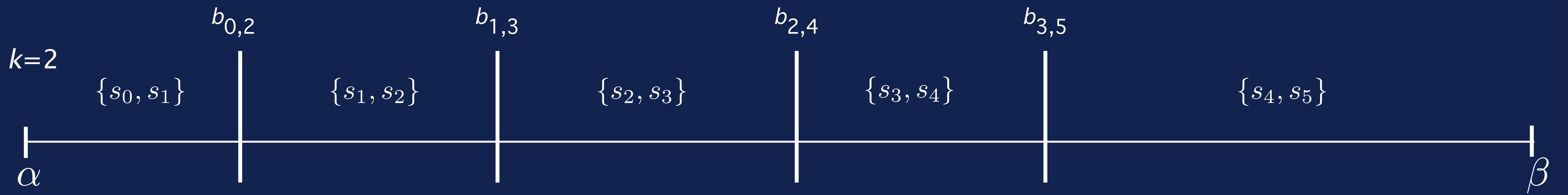
Valid Reconstruction DB_1



UNORDERED RESPONSES
EXACT RECONSTRUCTION

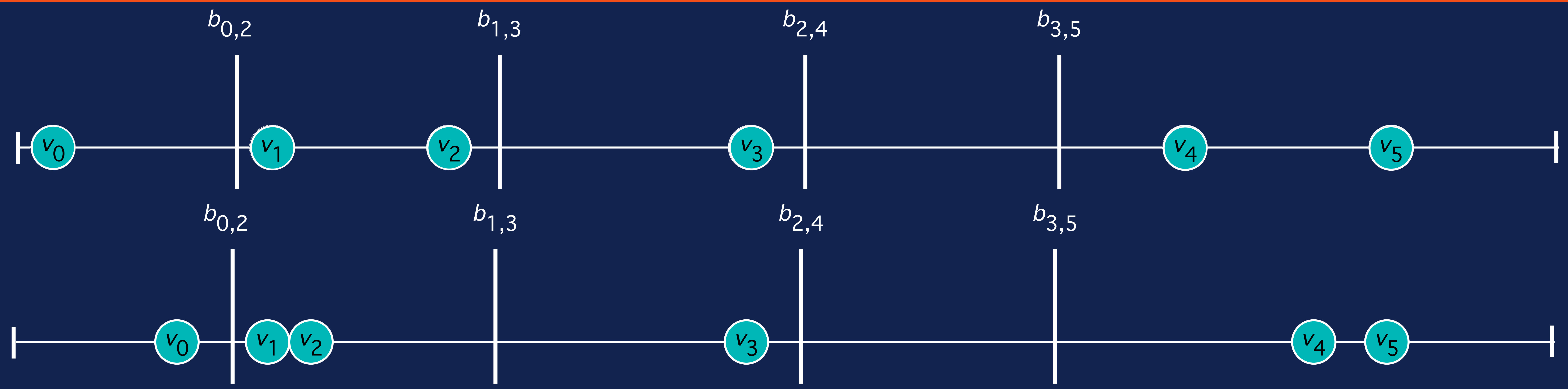
Impossible to achieve Exact Reconstruction

Best Case Scenario for the Adversary



Valid Reconstruction DB_1

Valid Reconstruction DB_2



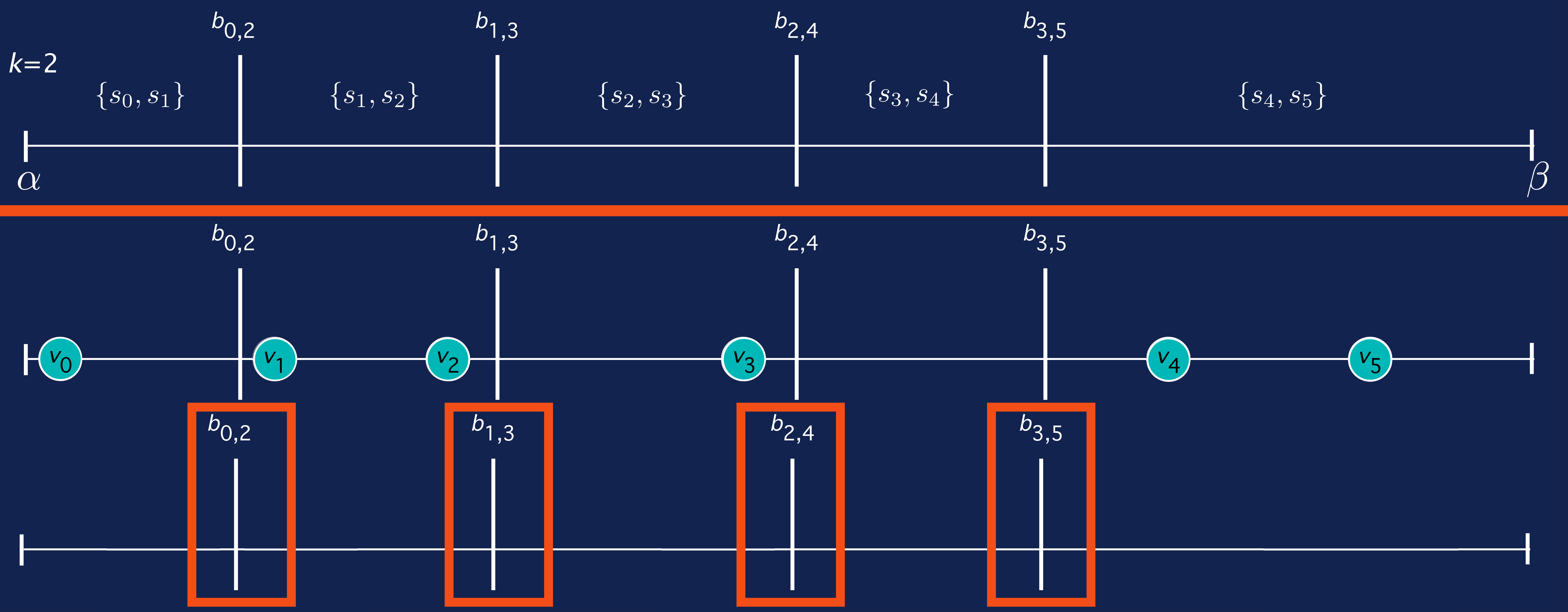
UNORDERED RESPONSES
EXACT RECONSTRUCTION

Impossible to achieve Exact Reconstruction

Best Case Scenario for the Adversary

Valid Reconstruction DB_1

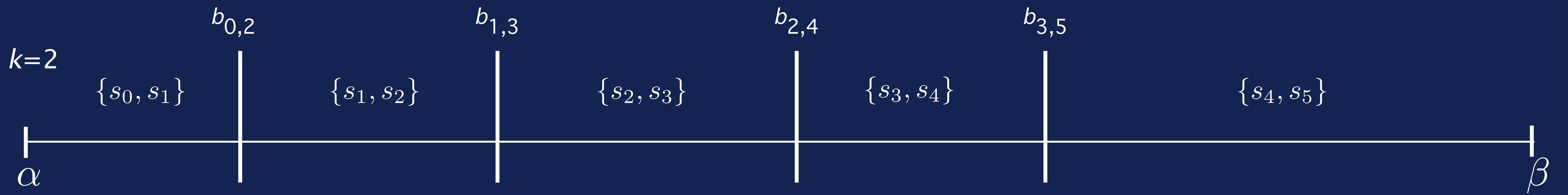
Valid Reconstruction DB_2



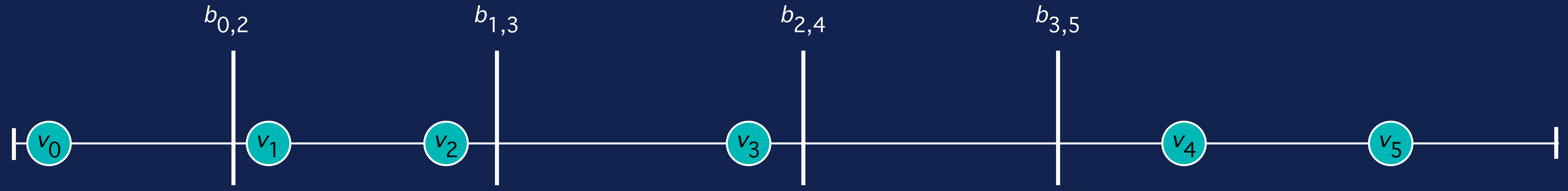
UNORDERED RESPONSES
EXACT RECONSTRUCTION

Impossible to achieve Exact Reconstruction

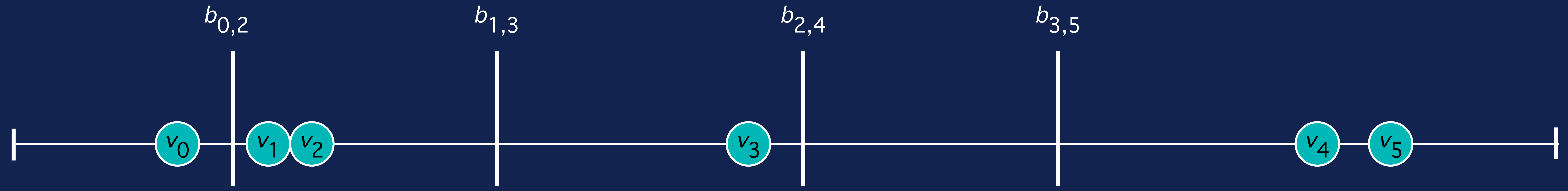
Best Case Scenario for the Adversary



Valid Reconstruction DB_1

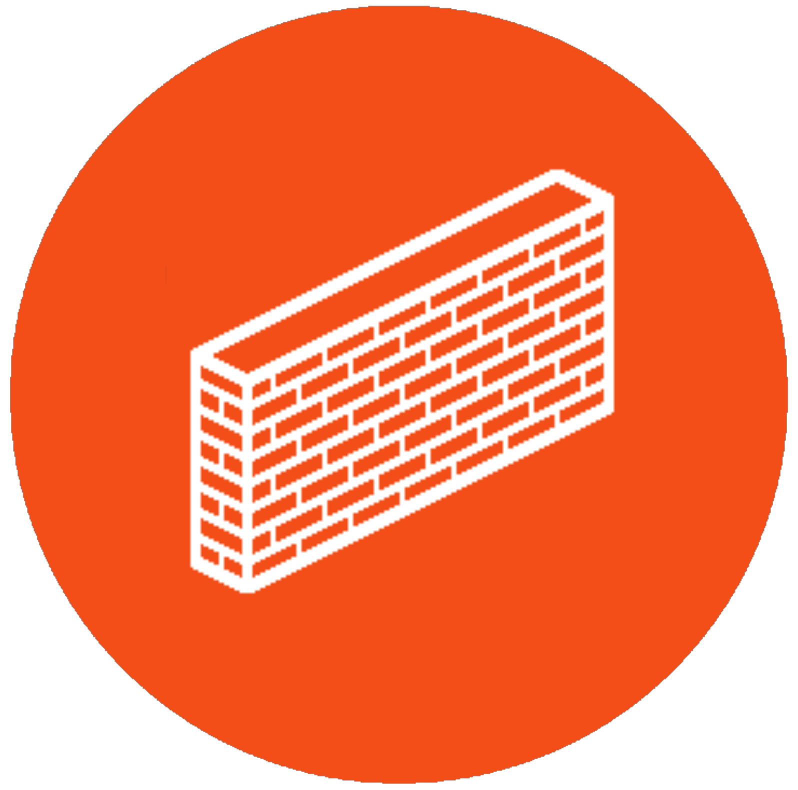


Valid Reconstruction DB_2

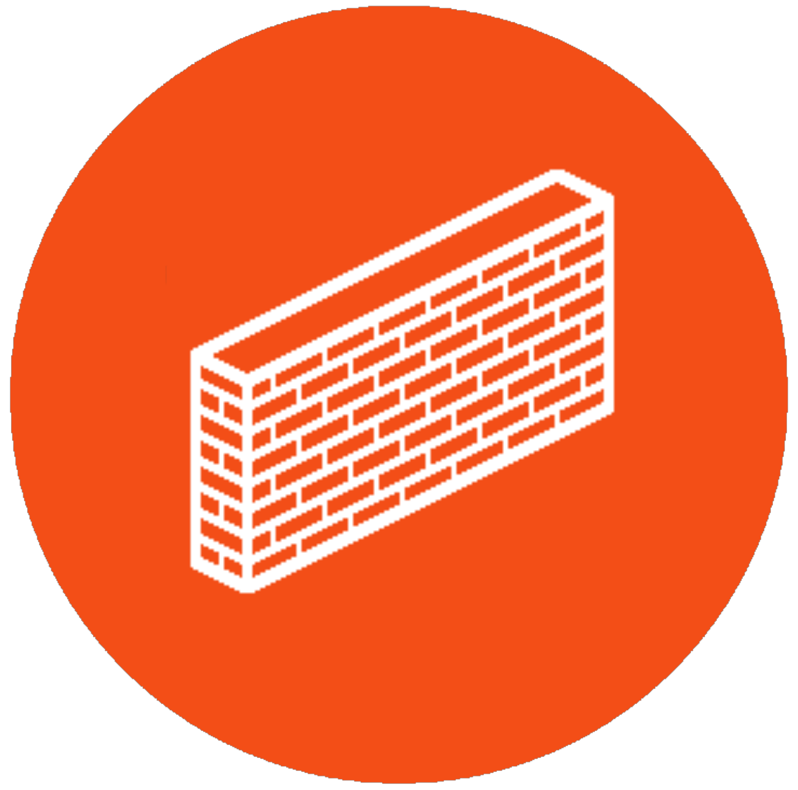


$Vor(DB_1) = Vor(DB_2) = \ldots$

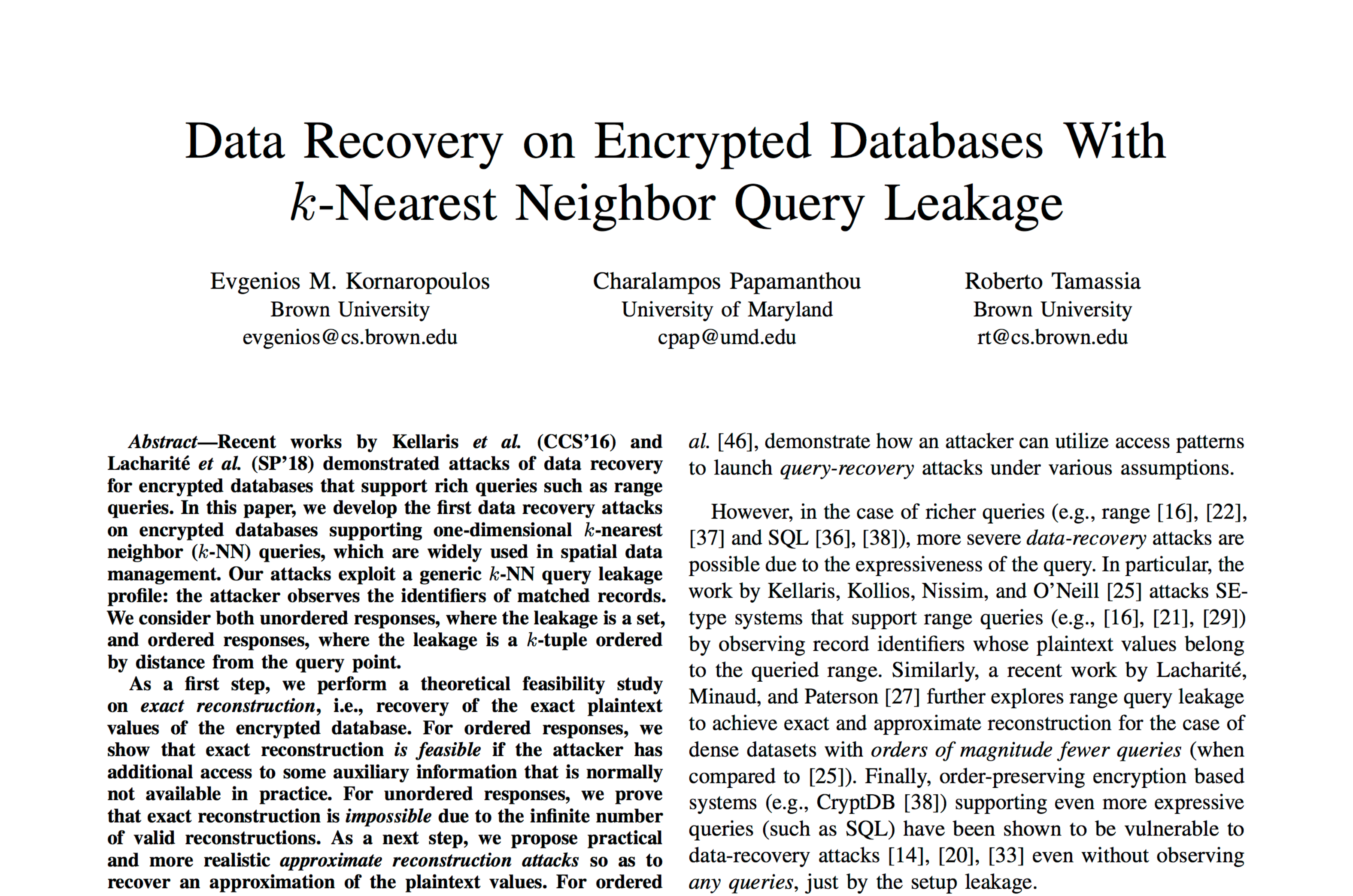
Many reconstructions that explain the Voronoi Diagram



Since there are **MANY** reconstructions and the exact recovery is **IMPOSSIBLE**, the encrypted values must be safe…



Since there are **MANY** reconstructions and the exact recovery is **IMPOSSIBLE**, the encrypted values must be safe…



Data Recovery on Encrypted Databases With k-Nearest Neighbor Query Leakage

Evgenios M. Kornaropoulos
Brown University
evgenios@cs.brown.edu

Charalampos Papamanthou
University of Maryland
cpap@umd.edu

Roberto Tamassia
Brown University
rt@cs.brown.edu

***Abstract*—Recent works by Kellaris *et al.* (CCS'16) and Lacharité *et al.* (SP'18) demonstrated attacks of data recovery for encrypted databases that support rich queries such as range queries. In this paper, we develop the first data recovery attacks on encrypted databases supporting one-dimensional k-nearest neighbor (k-NN) queries, which are widely used in spatial data management. Our attacks exploit a generic k-NN query leakage profile: the attacker observes the identifiers of matched records. We consider both unordered responses, where the leakage is a set, and ordered responses, where the leakage is a k-tuple ordered by distance from the query point.**

As a first step, we perform a theoretical feasibility study on *exact reconstruction*, i.e., recovery of the exact plaintext values of the encrypted database. For ordered responses, we show that exact reconstruction *is feasible* if the attacker has additional access to some auxiliary information that is normally not available in practice. For unordered responses, we prove that exact reconstruction is *impossible* due to the infinite number of valid reconstructions. As a next step, we propose practical and more realistic *approximate reconstruction attacks* so as to recover an approximation of the plaintext values. For ordered

al. [46], demonstrate how an attacker can utilize access patterns to launch *query-recovery* attacks under various assumptions.

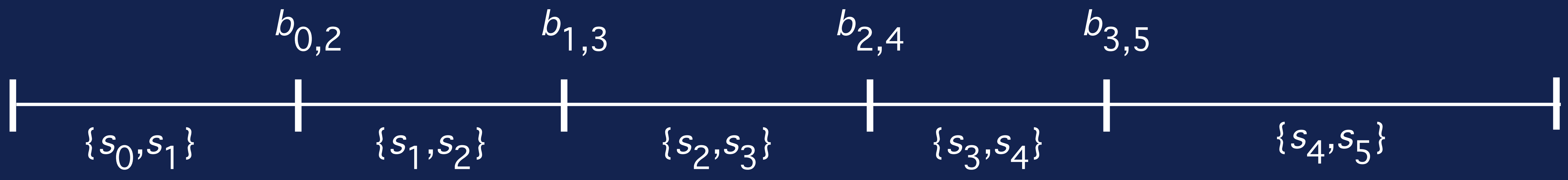
However, in the case of richer queries (e.g., range [16], [22], [37] and SQL [36], [38]), more severe *data-recovery* attacks are possible due to the expressiveness of the query. In particular, the work by Kellaris, Kollios, Nissim, and O'Neill [25] attacks SE-type systems that support range queries (e.g., [16], [21], [29]) by observing record identifiers whose plaintext values belong to the queried range. Similarly, a recent work by Lacharité, Minaud, and Paterson [27] further explores range query leakage to achieve exact and approximate reconstruction for the case of dense datasets with *orders of magnitude fewer queries* (when compared to [25]). Finally, order-preserving encryption based systems (e.g., CryptDB [38]) supporting even more expressive queries (such as SQL) have been shown to be vulnerable to data-recovery attacks [14], [20], [33] even without observing *any queries*, just by the setup leakage.

Answer: We can still compute an reconstruction that is **VERY CLOSE** to the encrypted DB

UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

In case all queries are issued:

The length of each Voronoi segments

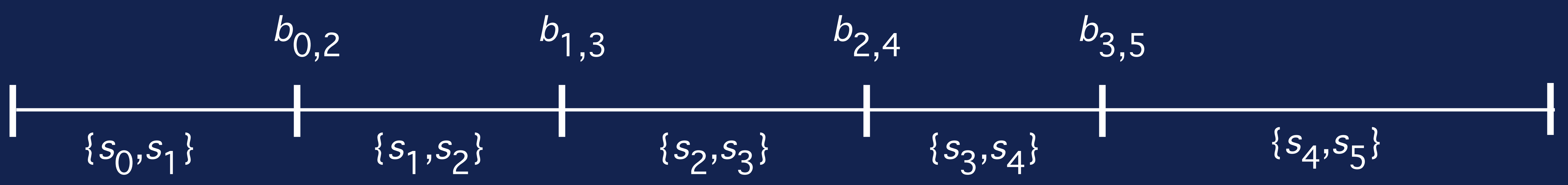


Uniform Query Distribution: Estimate via Concentration Bounds on Multinomials

UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

In case all queries are issued:

The **length** of each Voronoi segments



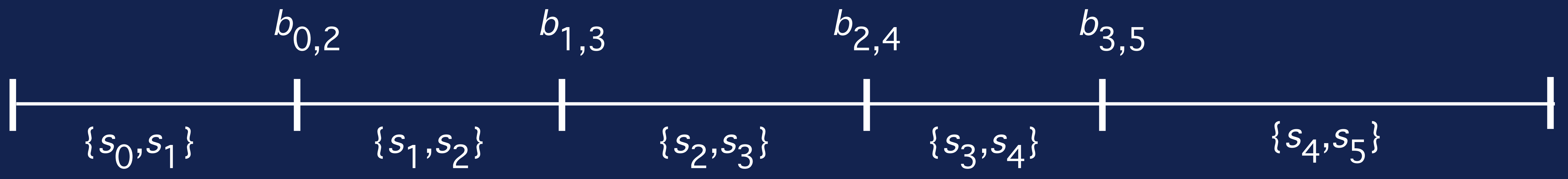
Goal:

Characterize the set of **all valid reconstructions** that explain the Voronoi Diagram

UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

In case all queries are issued:

The length of each Voronoi segments



Goal:

Characterize the set of all valid reconstructions that explain the Voronoi Diagram

What's Next:

Intuitive characterization = rigorous reconstruction guarantees

Modeling All Reconstructions:

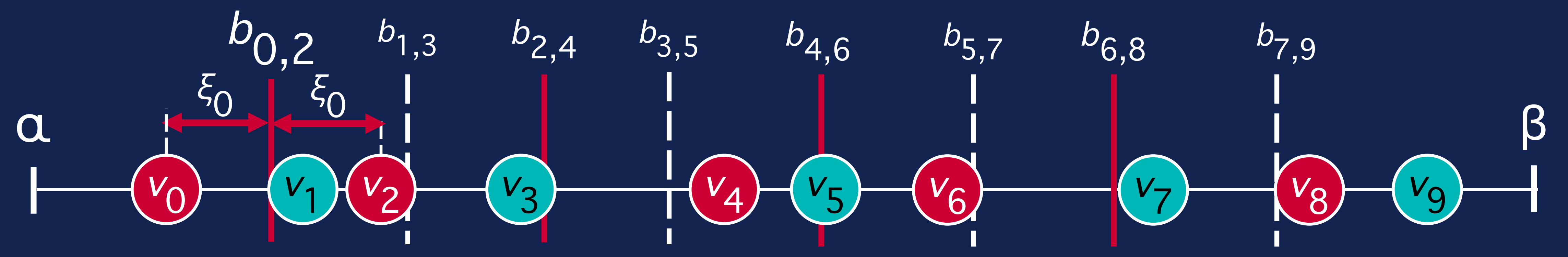
Modeling All Reconstructions:

Use geometry of bisectors to define unknowns

UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Modeling All Reconstructions:

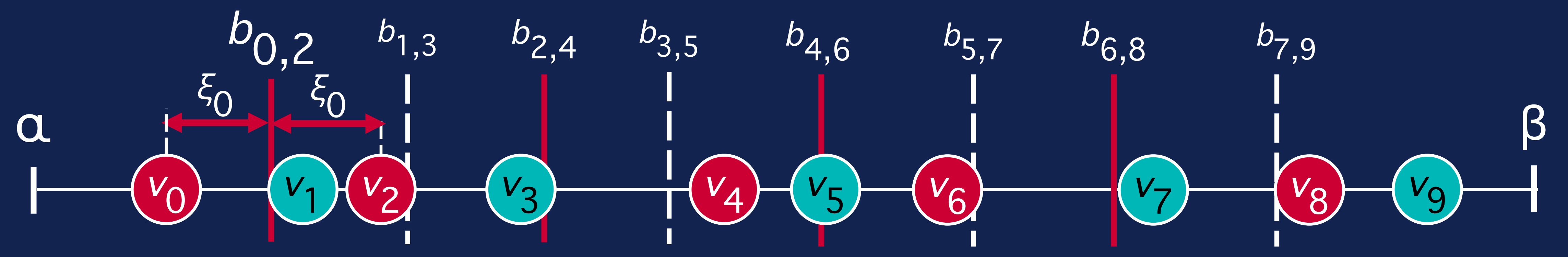
Use geometry of bisectors to define unknowns



$$v_0 = b_{0,2} - \xi_0$$
$$v_2 = b_{0,2} + \xi_0$$

Modeling All Reconstructions:

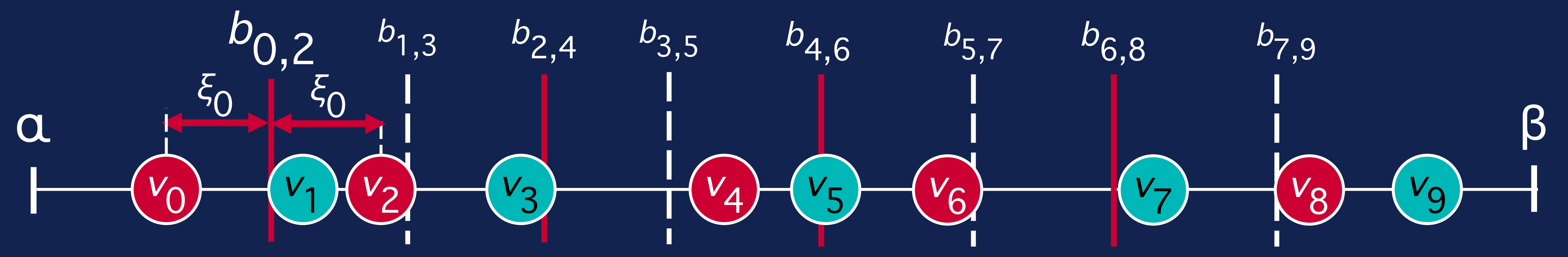
Use geometry of bisectors to define unknowns



$$v_0 = b_{0,2} - \xi_0$$
$$v_2 = b_{0,2} + \xi_0$$
$$v_4 = 2b_{2,4} - v_2$$

UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Modeling All Reconstructions:
Use geometry of bisectors to define unknowns

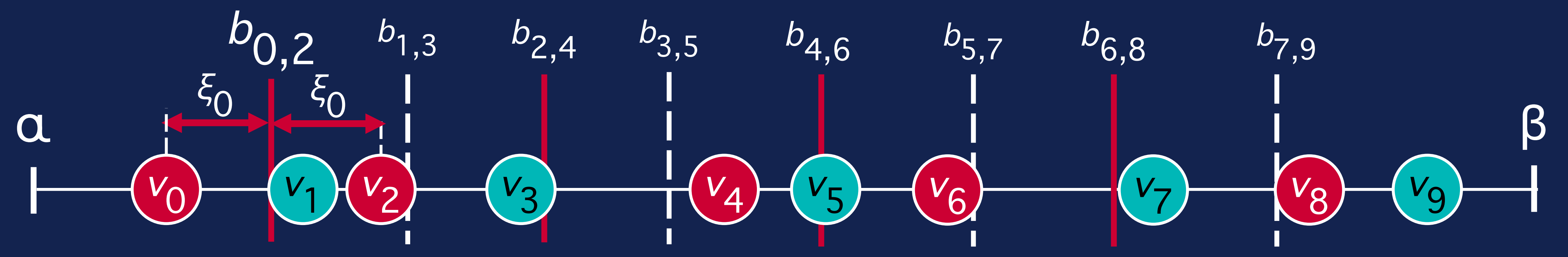


$$v_0 = b_{0,2} - \xi_0$$
$$v_2 = b_{0,2} + \xi_0$$
$$v_4 = 2b_{2,4} - v_2$$

UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Modeling All Reconstructions:

Use geometry of bisectors to define unknowns



$$v_0 = b_{0,2} - \xi_0$$

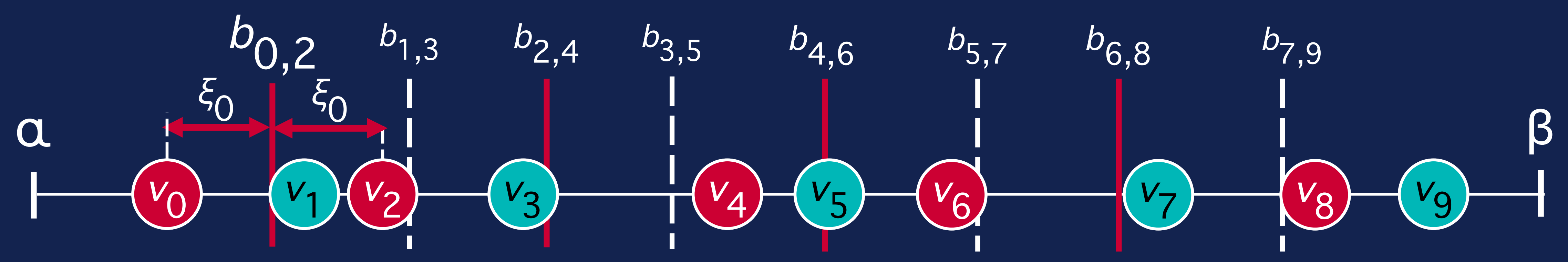
$$v_2 = b_{0,2} + \xi_0$$

$$v_4 = 2b_{2,4} - v_2 = 2b_{2,4} - b_{0,2} - \xi_0$$

UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Modeling All Reconstructions:

Use geometry of bisectors to define unknowns



$$v_0 = b_{0,2} - \xi_0$$

$$v_2 = b_{0,2} + \xi_0$$

$$v_4 = 2b_{2,4} - v_2 = 2b_{2,4} - b_{0,2} - \xi_0$$

$$v_6 = 2b_{4,6} - v_4 = 2b_{4,6} - 2b_{2,4} + b_{0,2} + \xi_0$$

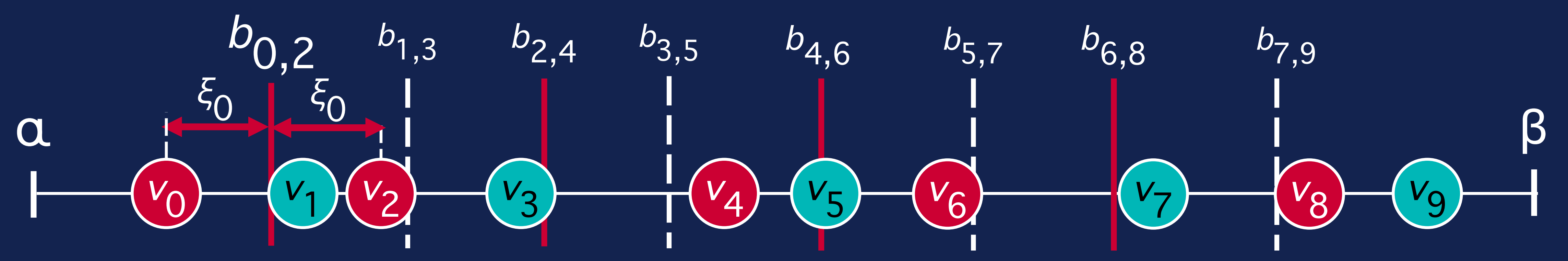
$$v_8 = 2b_{6,8} - v_6 = 2b_{6,8} - 2b_{4,6} + 2b_{2,4} - b_{0,2} - \xi_0$$

Half of the v_i as a function of unknown ξ_0

UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

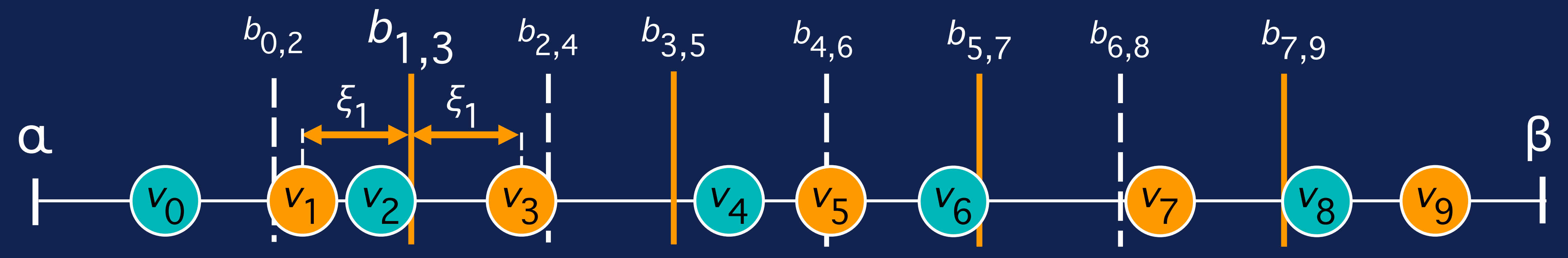
Modeling All Reconstructions:

Use geometry of bisectors to define unknowns



$$v_0 = b_{0,2} - \xi_0$$
$$v_2 = b_{0,2} + \xi_0$$
$$v_4 = 2b_{2,4} - v_2 = 2b_{2,4} - b_{0,2} - \xi_0$$
$$v_6 = 2b_{4,6} - v_4 = 2b_{4,6} - 2b_{2,4} + b_{0,2} + \xi_0$$
$$v_8 = 2b_{6,8} - v_6 = 2b_{6,8} - 2b_{4,6} + 2b_{2,4} - b_{0,2} - \xi_0$$

Half of the v_i as a function of unknown ξ_0

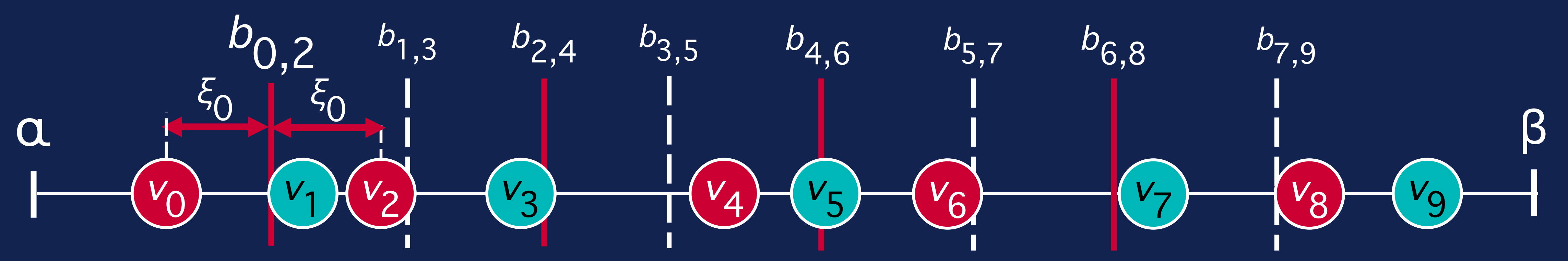


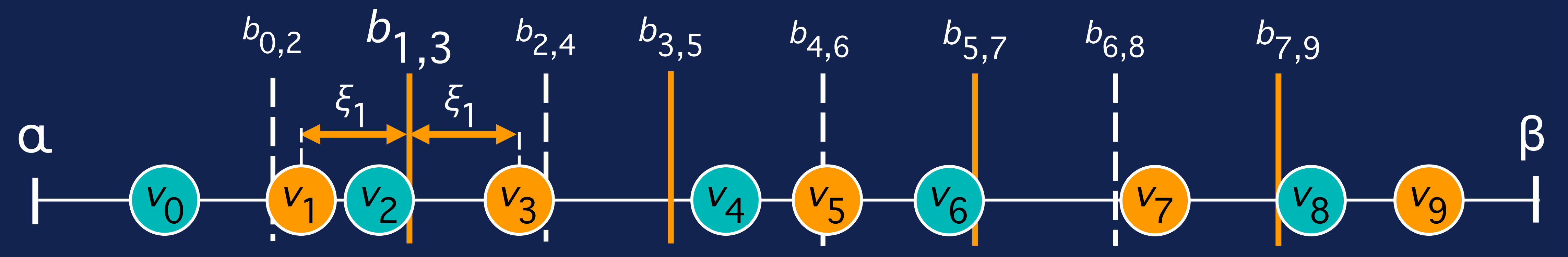
$$v_1 = b_{1,3} - \xi_1$$
$$v_3 = b_{1,3} + \xi_1$$
$$v_5 = 2b_{3,5} - v_3 = 2b_{3,5} - b_{1,3} - \xi_1$$
$$v_7 = 2b_{5,7} - v_5 = 2b_{5,7} - 2b_{3,5} + b_{1,3} + \xi_1$$
$$v_9 = 2b_{7,9} - v_7 = 2b_{7,9} - 2b_{5,7} + 2b_{3,5} - b_{1,3} - \xi_1$$

Other half of the v_i as a function of unknown ξ_1

Modeling All Reconstructions:

Use geometry of bisectors to define unknowns





$$v_0 = b_{0,2} - \xi_0$$
$$v_2 = b_{0,2} + \xi_0$$
$$v_4 = 2b_{2,4} - v_2 = 2b_{2,4} - b_{0,2} - \xi_0$$
$$v_6 = 2b_{4,6} - v_4 = 2b_{4,6} - 2b_{2,4} + b_{0,2} + \xi_0$$
$$v_8 = 2b_{6,8} - v_6 = 2b_{6,8} - 2b_{4,6} + 2b_{2,4} - b_{0,2} - \xi_0$$

Half of the v_i as a function of unknown ξ_0

$$v_1 = b_{1,3} - \xi_1$$
$$v_3 = b_{1,3} + \xi_1$$
$$v_5 = 2b_{3,5} - v_3 = 2b_{3,5} - b_{1,3} - \xi_1$$
$$v_7 = 2b_{5,7} - v_5 = 2b_{5,7} - 2b_{3,5} + b_{1,3} + \xi_1$$
$$v_9 = 2b_{7,9} - v_7 = 2b_{7,9} - 2b_{5,7} + 2b_{3,5} - b_{1,3} - \xi_1$$

Other half of the v_i as a function of unknown ξ_1

Reduced the space of reconstructions from n-dimensions to 2-dimensions

UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Modeling All Reconstructions:

Ordering Constraints:

$v_0 < v_1$

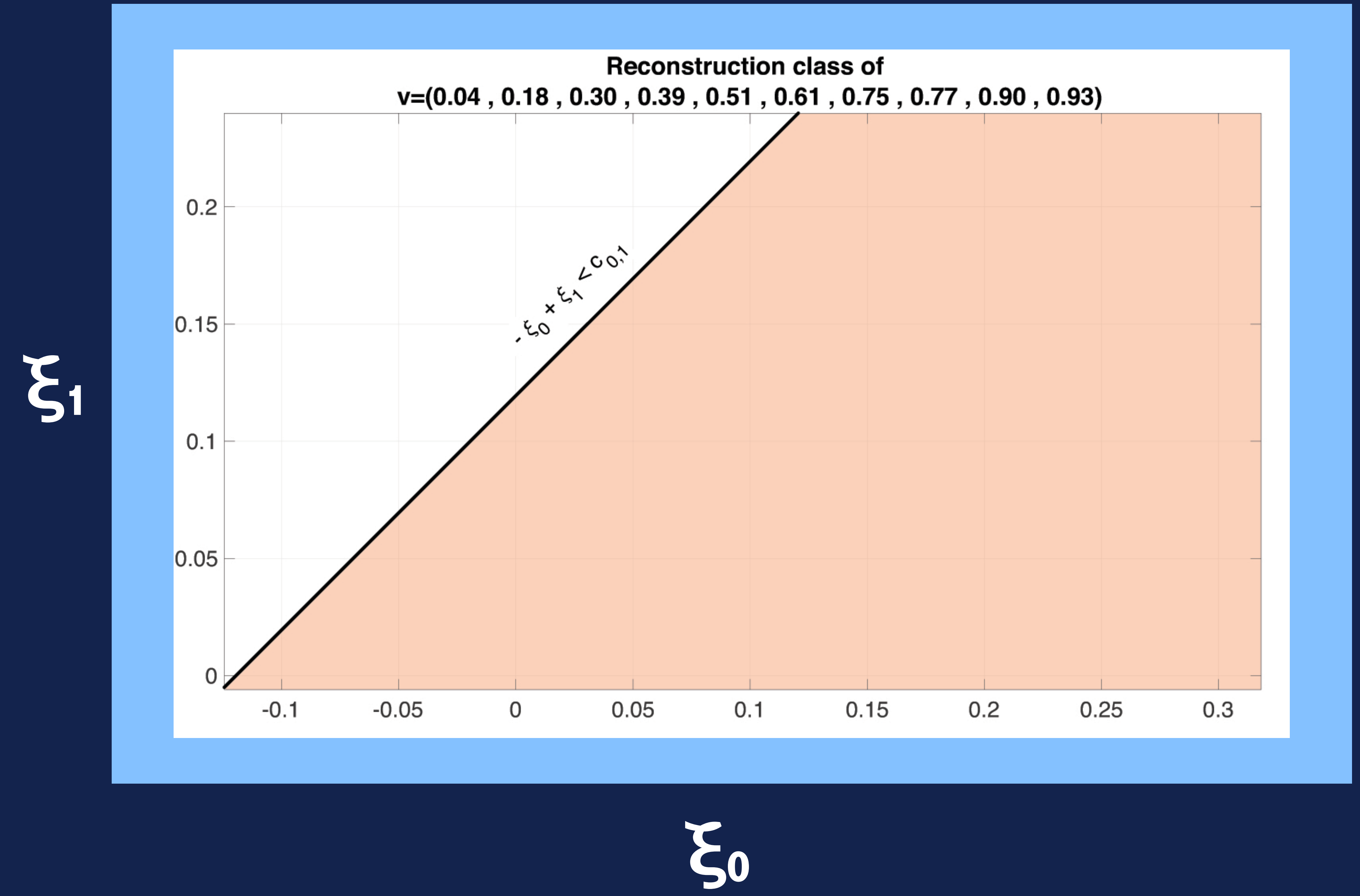
UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Modeling All Reconstructions:

Ordering Constraints:

$v_0 < v_1 \Rightarrow -\xi_0 + \xi_1 < c_{0,1}$, where $c_{0,1} = (b_{1,3} - b_{0,2})$

Geometric Characterization



UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Modeling All Reconstructions:

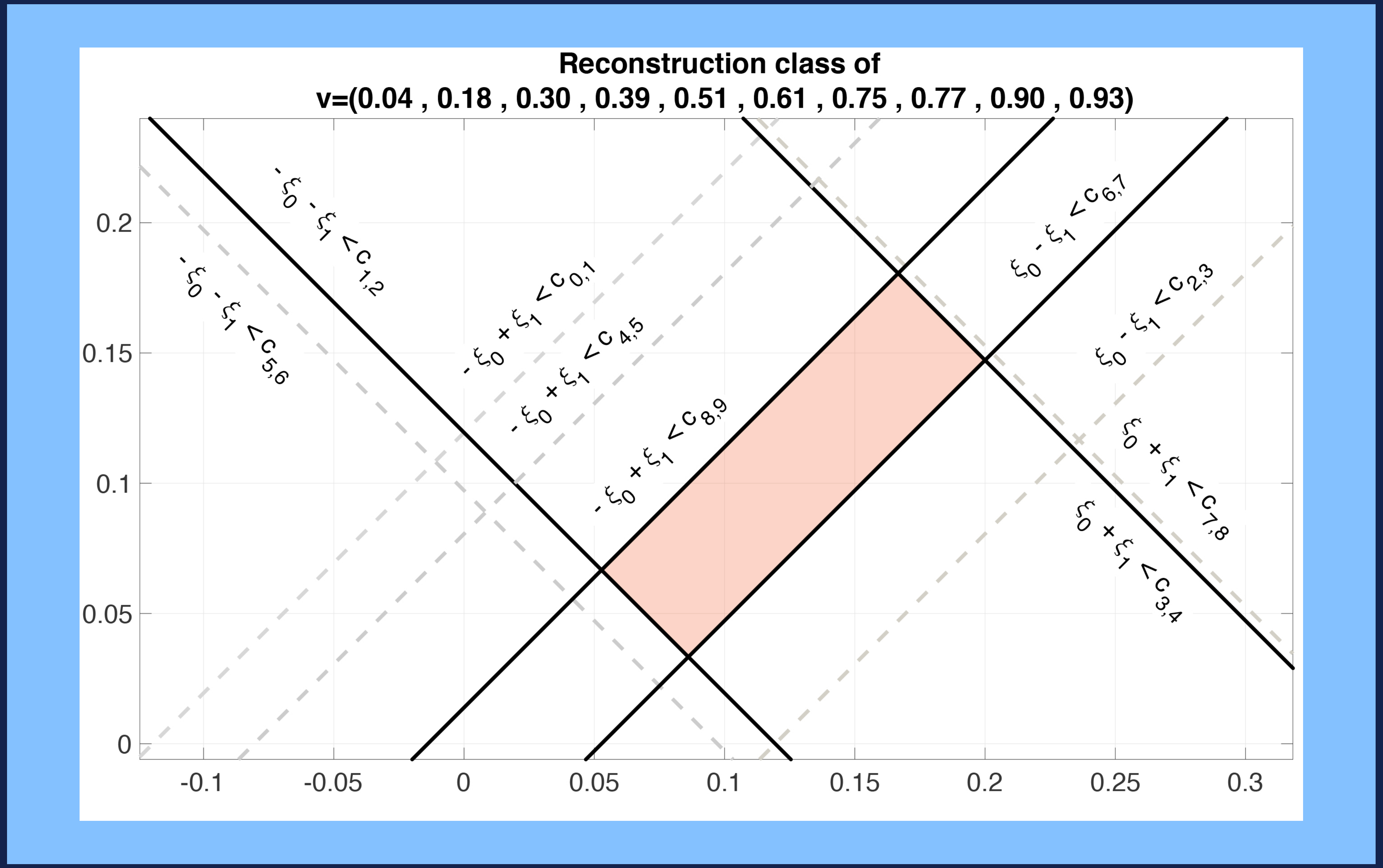
Geometric Characterization

Ordering Constraints:

$$v_0 < v_1 \Rightarrow -\xi_0 + \xi_1 < c_{0,1} \text{ , where } c_{0,1} = (b_{1,3} - b_{0,2})$$
$$v_1 < v_2 \Rightarrow -\xi_0 - \xi_1 < c_{1,2} \text{ , where } c_{1,2} = -(b_{1,3} - b_{0,2})$$
$$v_2 < v_3 \Rightarrow \xi_0 - \xi_1 < c_{2,3} \text{ , where } c_{2,3} = (b_{1,3} - b_{0,2})$$
$$v_3 < v_4 \Rightarrow \xi_0 + \xi_1 < c_{3,4} \text{ , where } c_{3,4} = (b_{2,4} - b_{1,3}) + (b_{2,4} - b_{0,2})$$

$$v_4 < v_5 \Rightarrow -\xi_0 + \xi_1 < c_{4,5} \text{ , where } c_{4,5} = 2(b_{3,5} - b_{2,4}) - (b_{1,3} - b_{0,2})$$
$$v_5 < v_6 \Rightarrow -\xi_0 - \xi_1 < c_{5,6}, \text{ where } c_{5,6} = 2(b_{4,6} - b_{3,5}) - (b_{2,4} - b_{0,2}) - (b_{2,4} - b_{1,3})$$
$$v_6 < v_7 \Rightarrow \xi_0 - \xi_1 < c_{6,7}, \text{ where } c_{6,7} = 2(b_{5,7} - b_{4,6}) - 2(b_{3,5} - b_{2,4}) + (b_{1,3} - b_{0,2})$$
$$v_7 < v_8 \Rightarrow \xi_0 + \xi_1 < c_{7,8}, \text{ where } c_{7,8} = 2(b_{6,8} - b_{5,7}) - 2(b_{4,6} - b_{3,5}) + (b_{2,4} - b_{1,3}) + (b_{2,4} - b_{0,2})$$
$$v_8 < v_9 \Rightarrow -\xi_0 + \xi_1 < c_{8,9}, \text{ where } c_{8,9} = 2(b_{7,9} - b_{6,8}) - 2(b_{5,7} - b_{4,6}) + 2(b_{3,5} - b_{2,4}) - (b_{1,3} - b_{0,2})$$

ξ_1

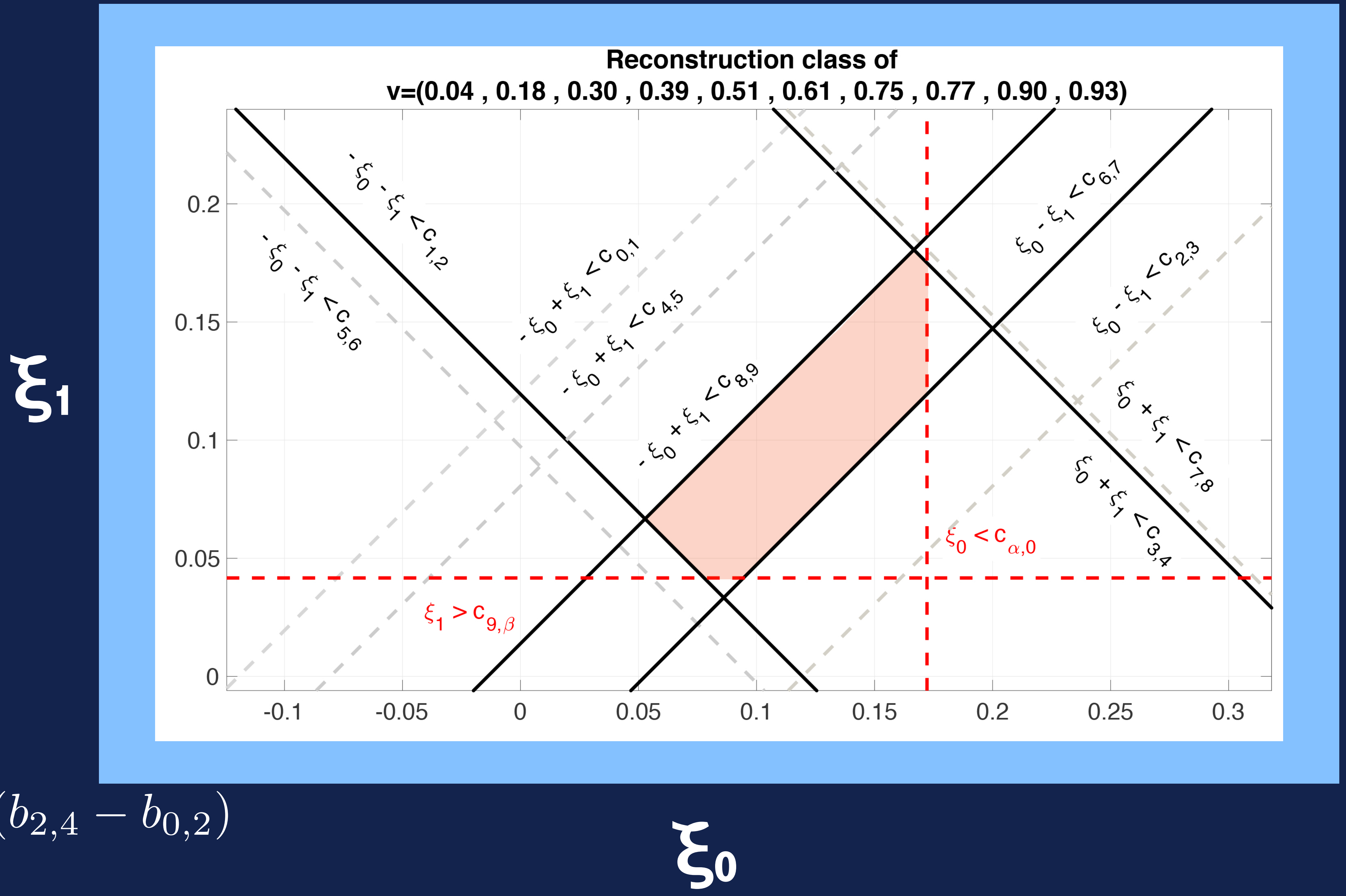


ξ_0

UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Modeling All Reconstructions:

Geometric Characterization



Ordering Constraints:

$v_0 < v_1 \Rightarrow -\xi_0 + \xi_1 < c_{0,1}$, where $c_{0,1} = (b_{1,3} - b_{0,2})$

$v_1 < v_2 \Rightarrow -\xi_0 - \xi_1 < c_{1,2}$, where $c_{1,2} = -(b_{1,3} - b_{0,2})$

$v_2 < v_3 \Rightarrow \xi_0 - \xi_1 < c_{2,3}$, where $c_{2,3} = (b_{1,3} - b_{0,2})$

$v_3 < v_4 \Rightarrow \xi_0 + \xi_1 < c_{3,4}$, where $c_{3,4} = (b_{2,4} - b_{1,3}) + (b_{2,4} - b_{0,2})$

$v_4 < v_5 \Rightarrow -\xi_0 + \xi_1 < c_{4,5}$, where $c_{4,5} = 2(b_{3,5} - b_{2,4}) - (b_{1,3} - b_{0,2})$

$v_5 < v_6 \Rightarrow -\xi_0 - \xi_1 < c_{5,6}$, where $c_{5,6} = 2(b_{4,6} - b_{3,5}) - (b_{2,4} - b_{0,2}) - (b_{2,4} - b_{1,3})$

$v_6 < v_7 \Rightarrow \xi_0 - \xi_1 < c_{6,7}$, where $c_{6,7} = 2(b_{5,7} - b_{4,6}) - 2(b_{3,5} - b_{2,4}) + (b_{1,3} - b_{0,2})$

$v_7 < v_8 \Rightarrow \xi_0 + \xi_1 < c_{7,8}$, where $c_{7,8} = 2(b_{6,8} - b_{5,7}) - 2(b_{4,6} - b_{3,5}) + (b_{2,4} - b_{1,3}) + (b_{2,4} - b_{0,2})$

$v_8 < v_9 \Rightarrow -\xi_0 + \xi_1 < c_{8,9}$, where $c_{8,9} = 2(b_{7,9} - b_{6,8}) - 2(b_{5,7} - b_{4,6}) + 2(b_{3,5} - b_{2,4}) - (b_{1,3} - b_{0,2})$

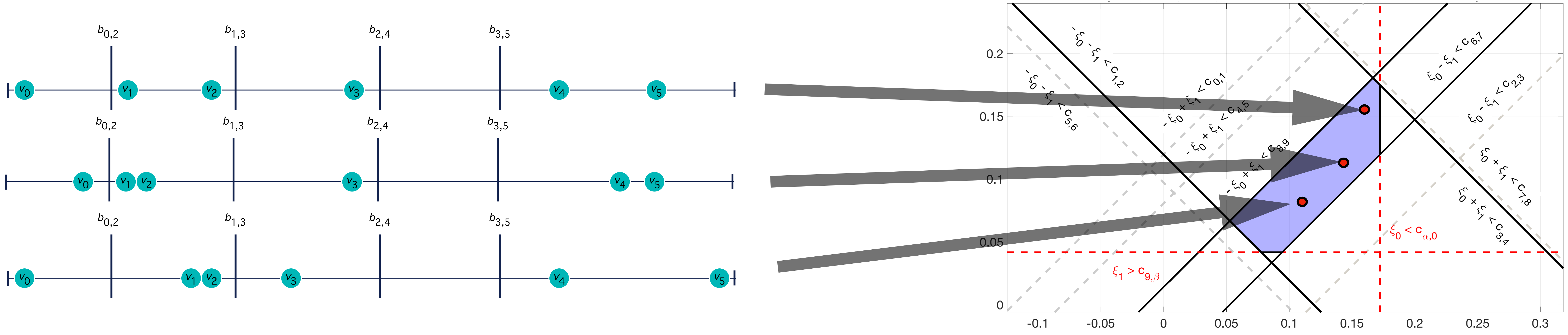
Boundary Constraints:

$\alpha < v_0 \Rightarrow \xi_0 < c_{\alpha,0}$, where $c_{\alpha,0} = b_{0,2} - \alpha$

$v_9 < \beta \Rightarrow \xi_1 > c_{9,\beta}$, where $c_{9,\beta} = 2b_{7,9} - 2b_{5,7} + 2b_{3,5} - b_{1,3} - \beta$

“Squeezed” the seemingly large space of valid reconstructions into a small polygon

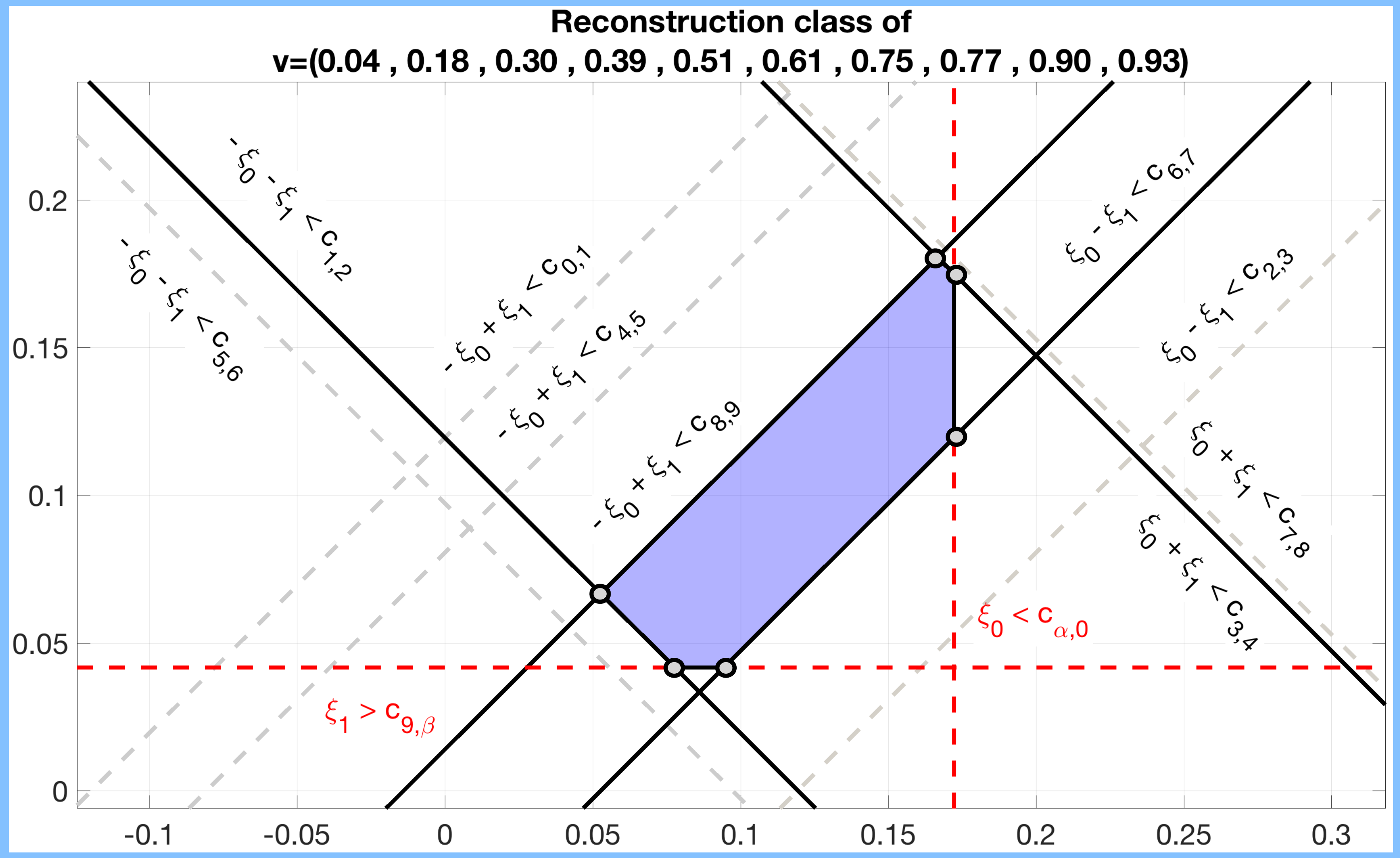
Valid Reconstructions **Geometric Characterization**



UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Original DB: $v' = (v'_0, \ldots, v'_{n-1})$

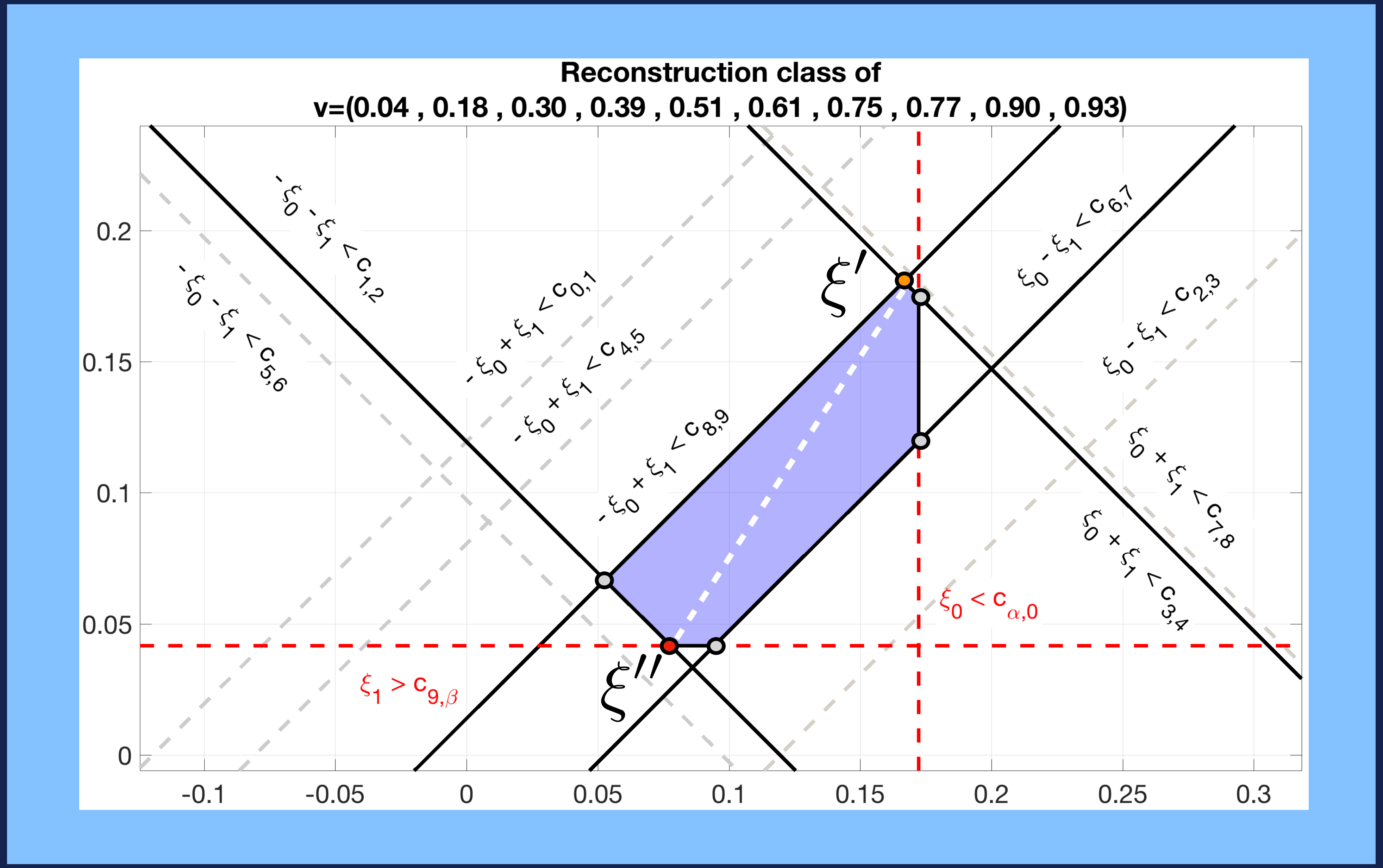
Reconstr. DB: $v'' = (v''_0, \ldots, v''_{n-1})$



UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Original DB: $v' = (v'_0, \ldots, v'_{n-1})$

Reconstr. DB: $v'' = (v''_0, \ldots, v''_{n-1})$



Maximum Error

Reconstruction Error between v', v''

$$\max_{i \in [0, n-1]} |v'_i - v''_i| \leq diam(F_v)$$

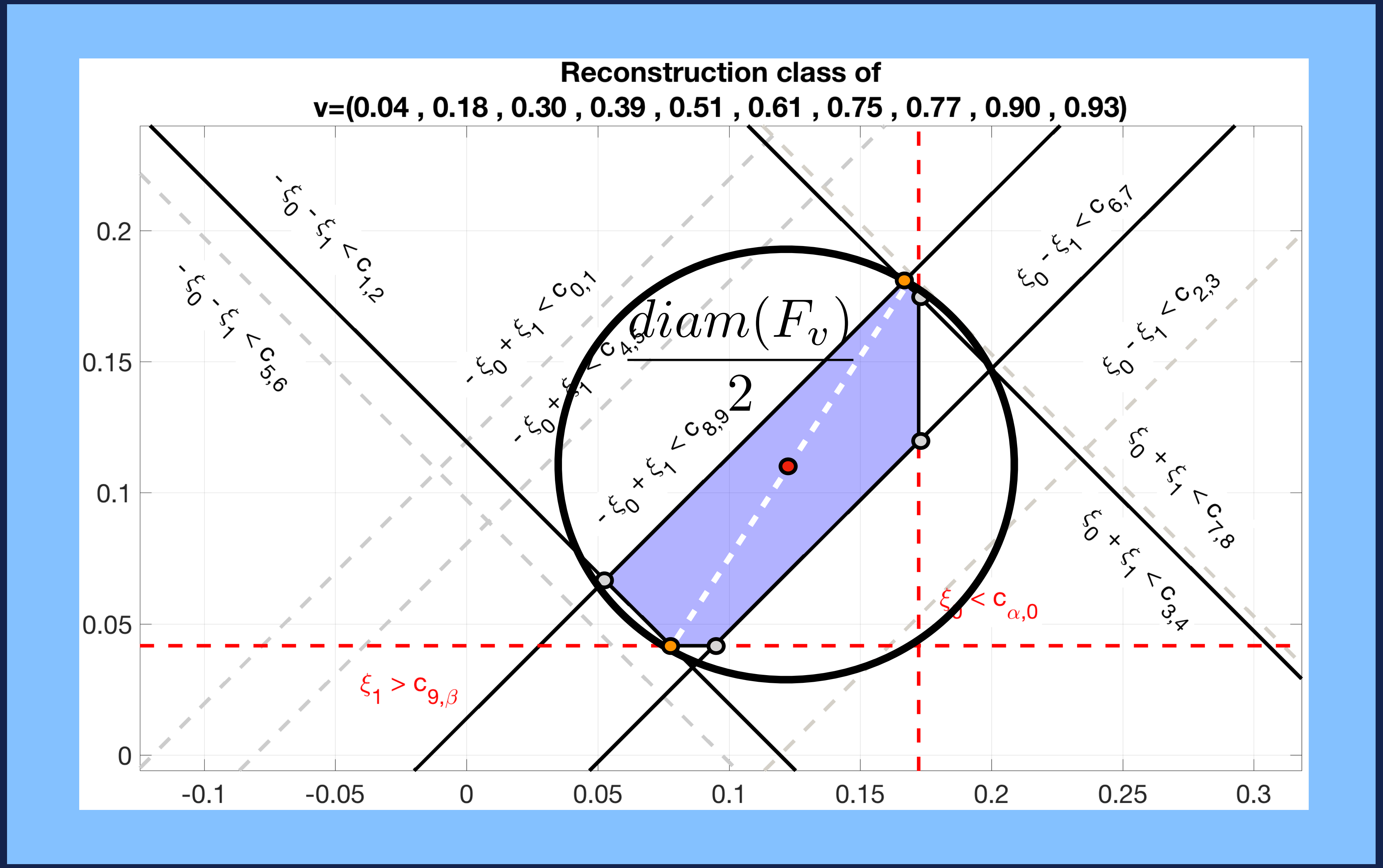
UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Original DB: $v' = (v'_0, \ldots, v'_{n-1})$

Reconstr. DB: $v'' = (v''_0, \ldots, v''_{n-1})$

Reconstruction Error between v', v''

$$\max_{i \in [0, n-1]} |v'_i - v''_i| \le diam(F_v)$$



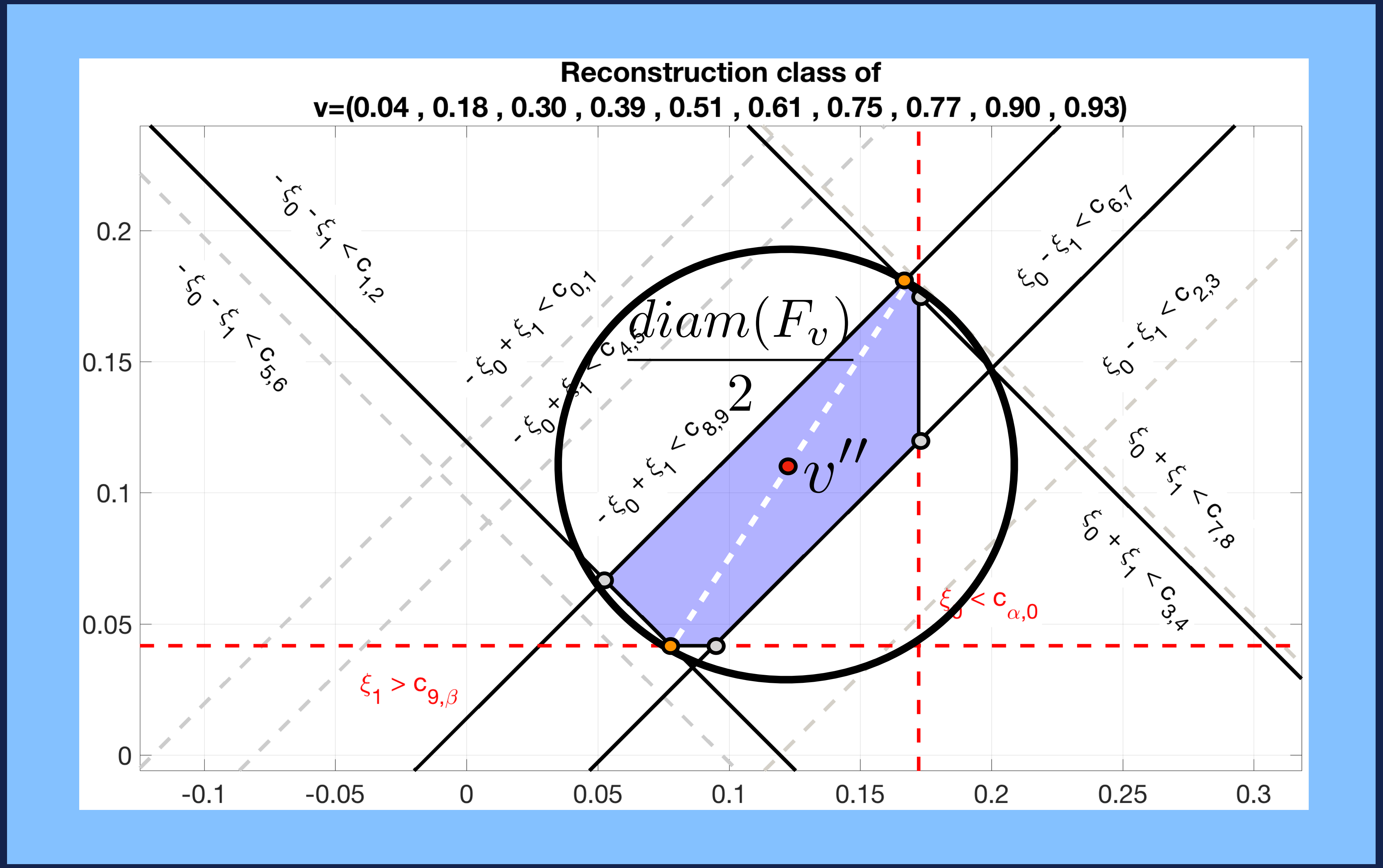
UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Original DB: $v' = (v'_0, \ldots, v'_{n-1})$

Reconstr. DB: $v'' = (v''_0, \ldots, v''_{n-1})$

Reconstruction Error between v', v''

$$\max_{i \in [0, n-1]} |v'_i - v''_i| \leq diam(F_v)$$



Our Reconstruction

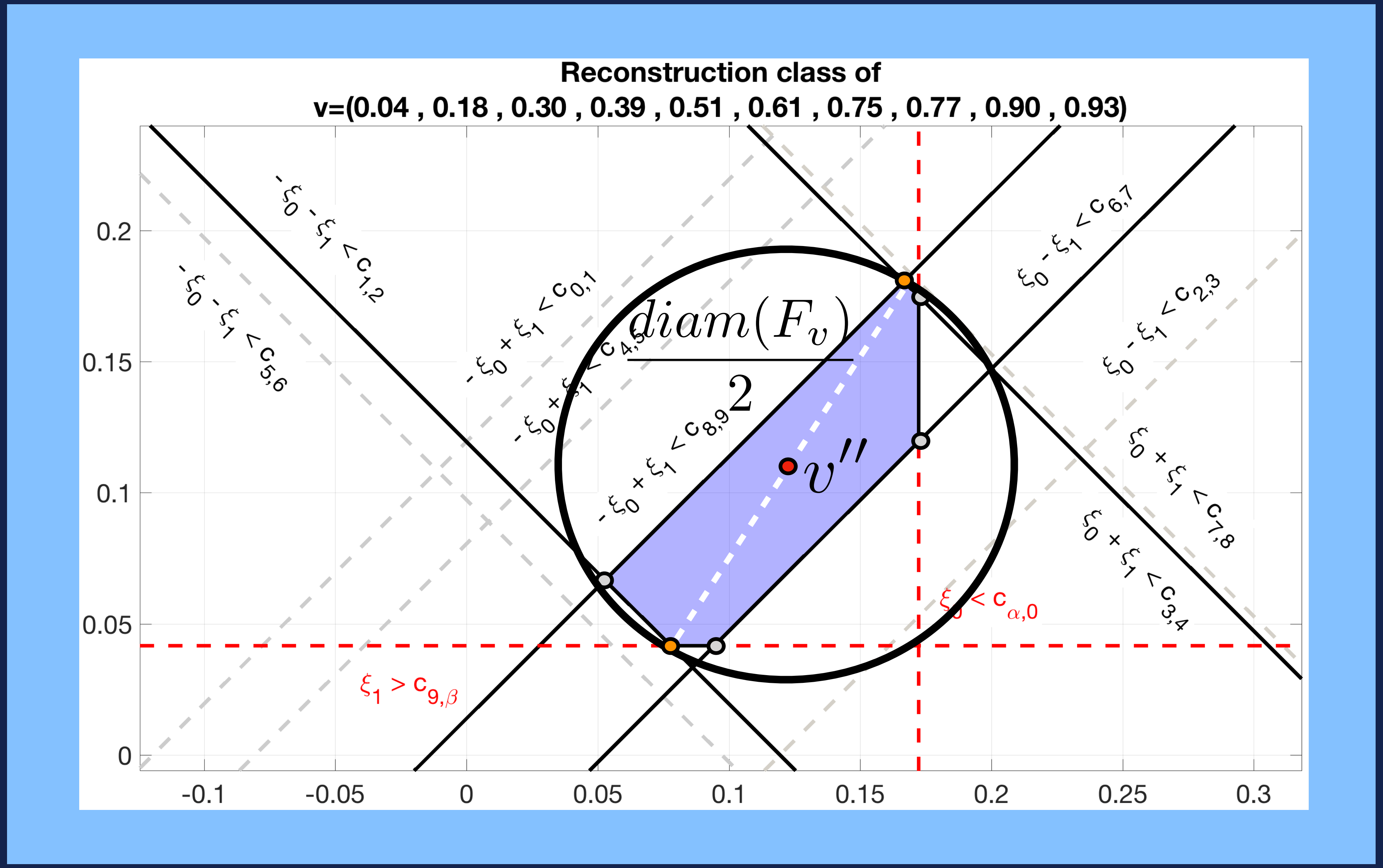
UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Original DB: $v' = (v'_0, \ldots, v'_{n-1})$

Reconstr. DB: $v'' = (v''_0, \ldots, v''_{n-1})$

Reconstruction Error between v', v''

$$\max_{i \in [0, n-1]} |v'_i - v''_i| \le diam(F_v)$$



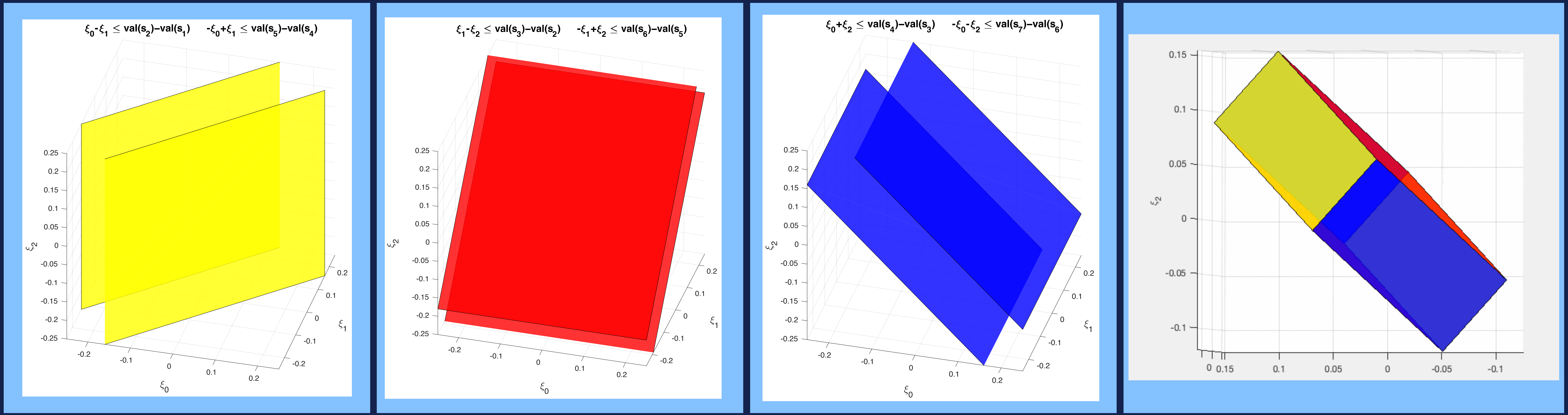
Our Reconstruction

The worst case reconstruction between v'' and every DB in F_v is upper-bounded by $\frac{diam(F_v)}{2}$

UNORDERED RESPONSES
APPROXIMATE RECONSTRUCTION*

Case k=3

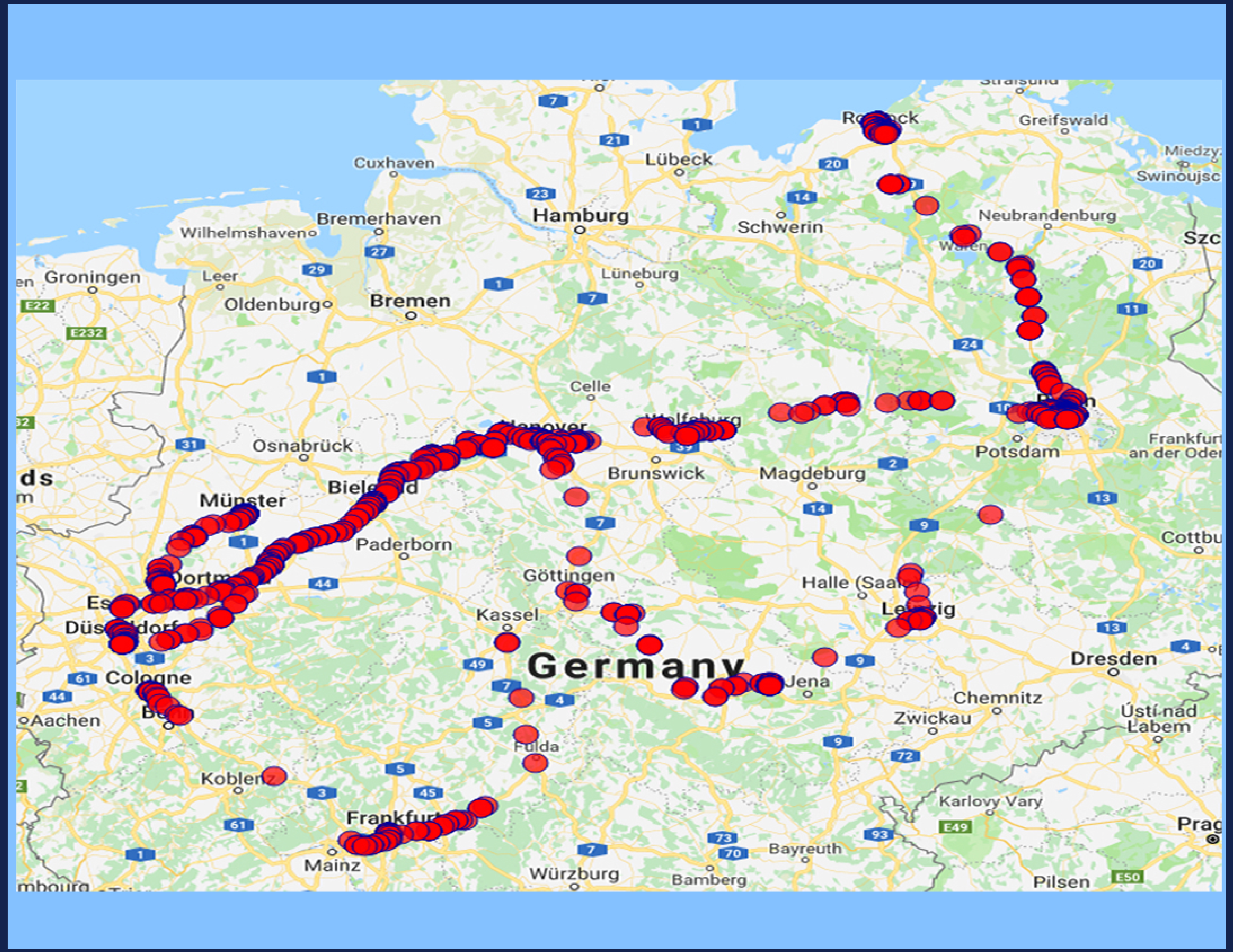
F_v



k-NN queries ⟶ F_v is a polytope in k-dimensional space

EVALUATION
ORDERED & UNORDERED RESPONSES

1-31 October 2009

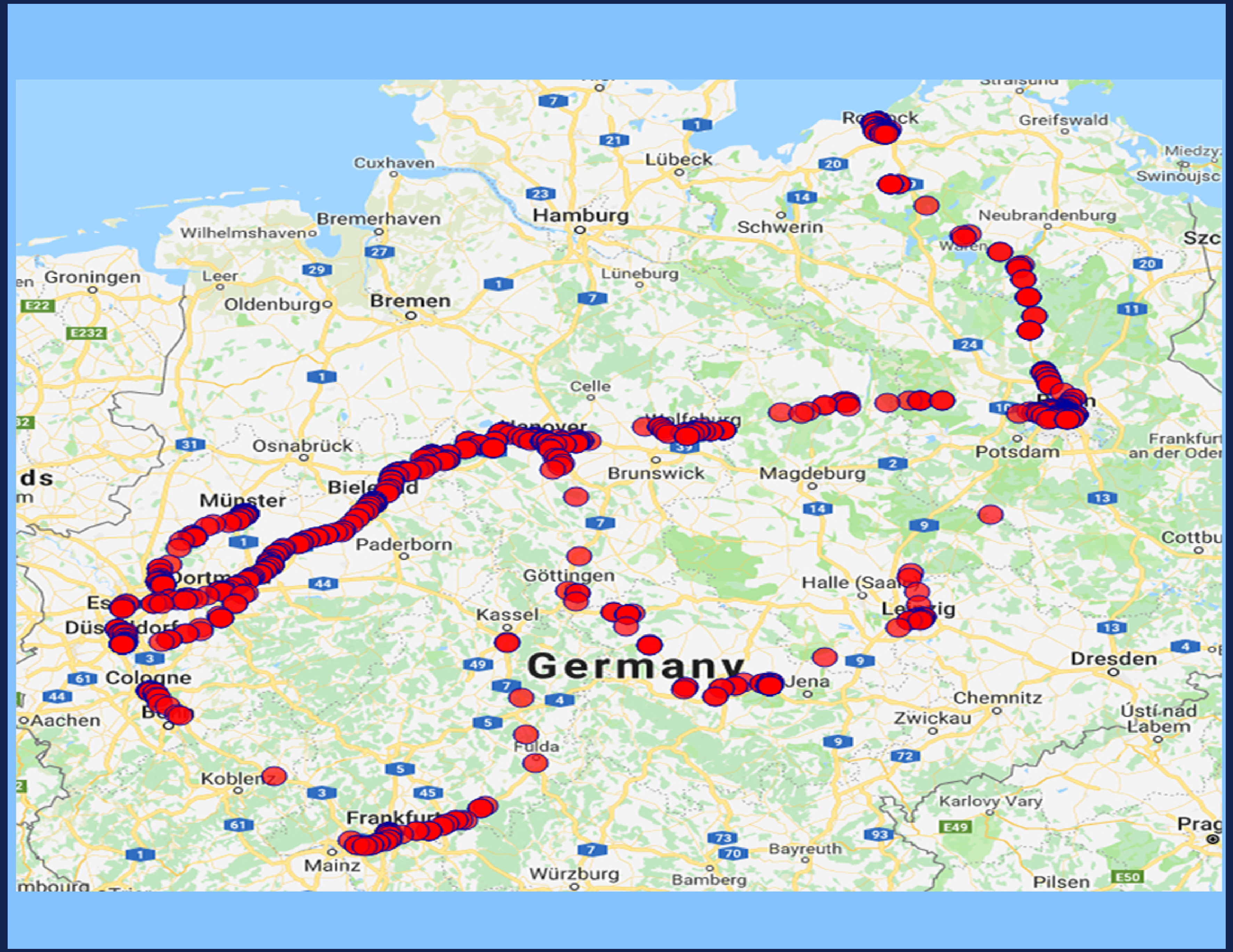


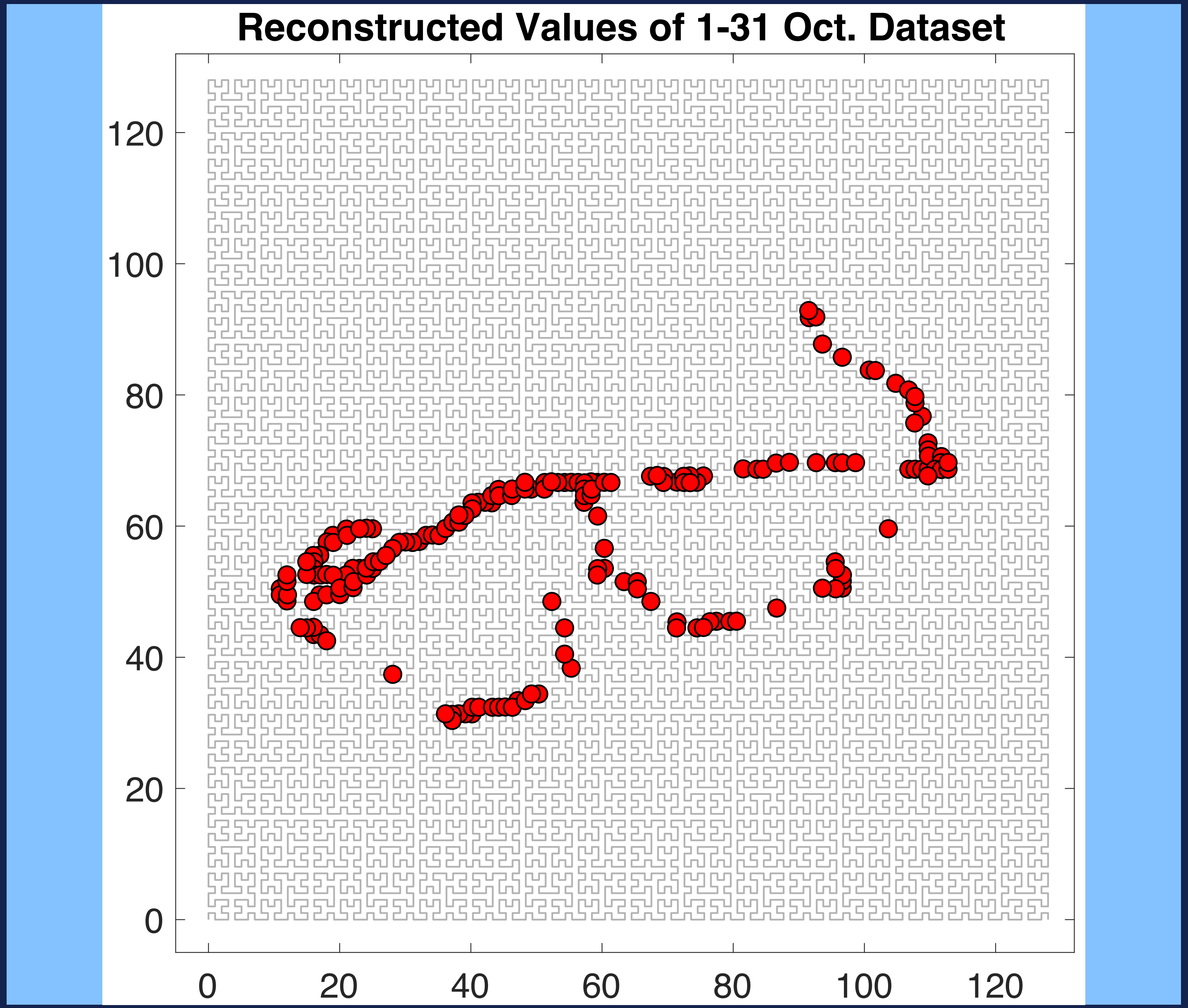
-Geolocation of politician Spitz

-Simulated k-NN Leakage from queries on his location DB

EVALUATION
ORDERED & UNORDERED RESPONSES

1-31 October 2009





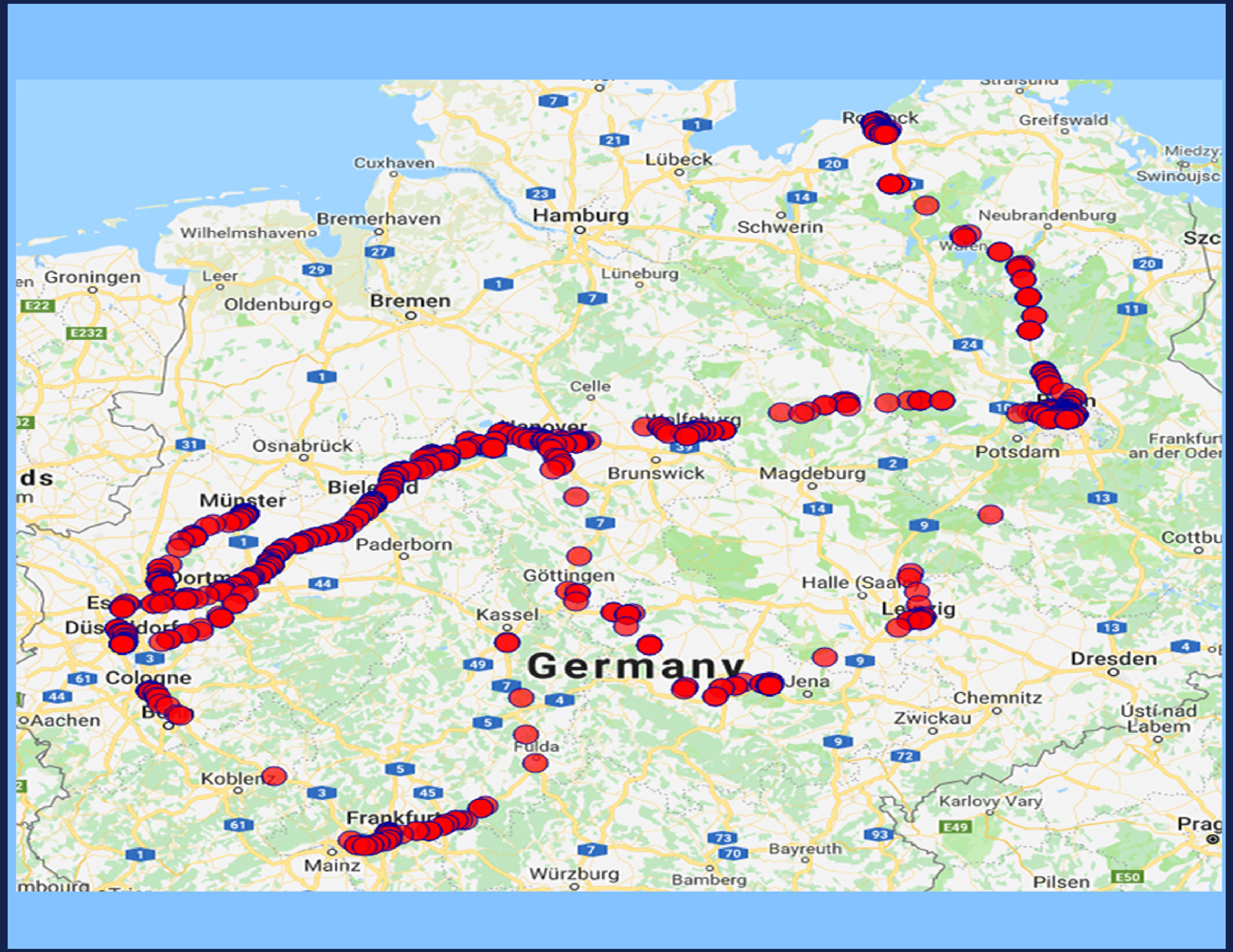
-Geolocation of politician Spitz

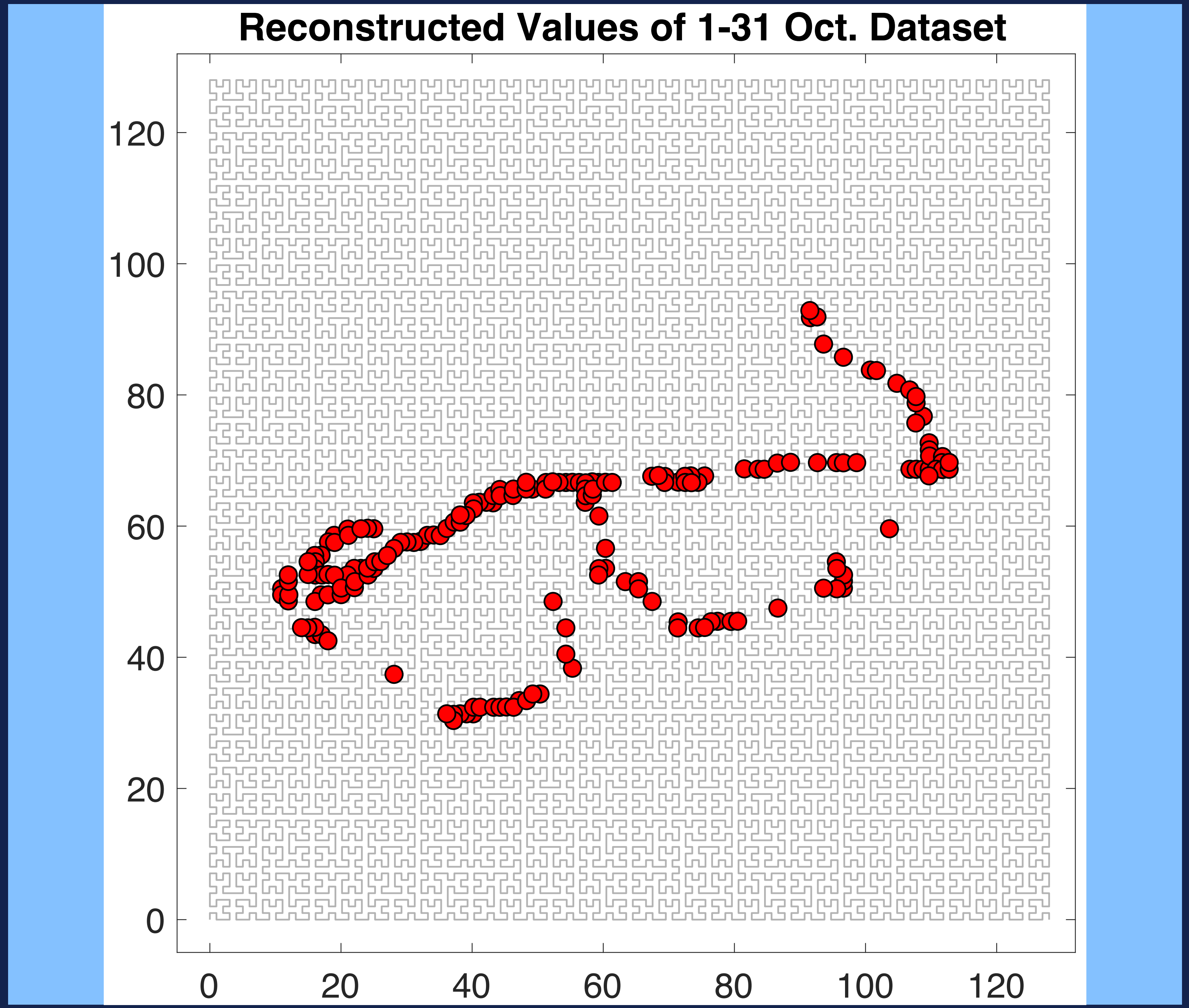
-Simulated k-NN Leakage from queries on his location DB

EVALUATION

ORDERED & UNORDERED RESPONSES

1-31 October 2009





-Geolocation of politician Spitz

-Simulated k-NN Leakage from queries on his location DB

	1-31 October, $m = 250 \cdot 10^6$, $n = 183$		
	diameter	Absolute Error	Success
$k = 2$	1.8	1.0	70%
$k = 5$	6.4	1.4	95%
$k = 8$	12.8	1.4	95%

k-NN EXACT RECONSTRUCTION

ORDERED RESPONSES: Possible when all encrypted queries are issued

UNORDERED RESPONSES: Impossible due to many reconstructions

k-NN APPROXIMATE RECONSTRUCTION

ORDERED RESPONSES: Approximate reconstruction when not all encrypted queries are issued

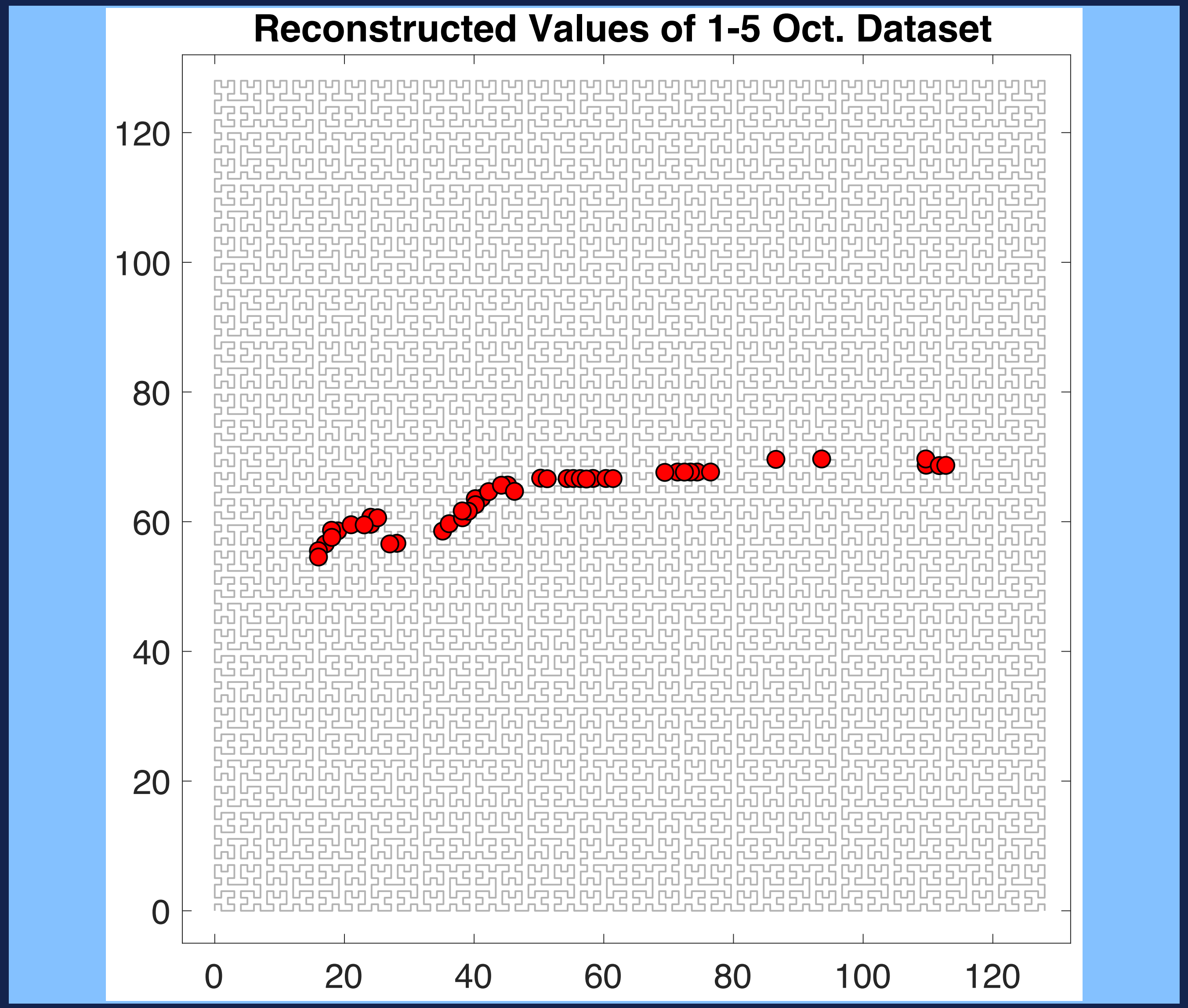
UNORDERED RESPONSES: Even with many reconstructions approximate with bounded error

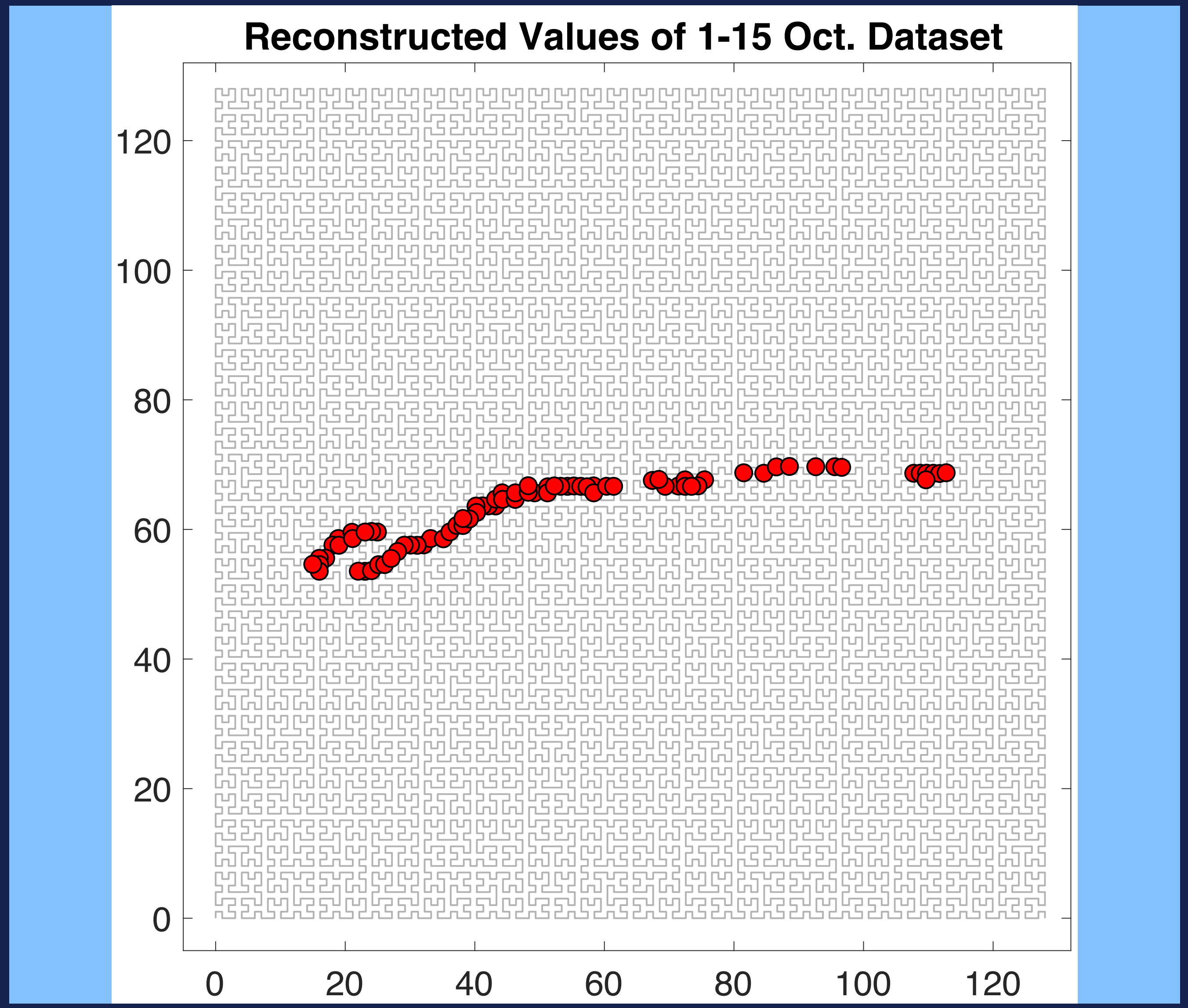
Thank you!

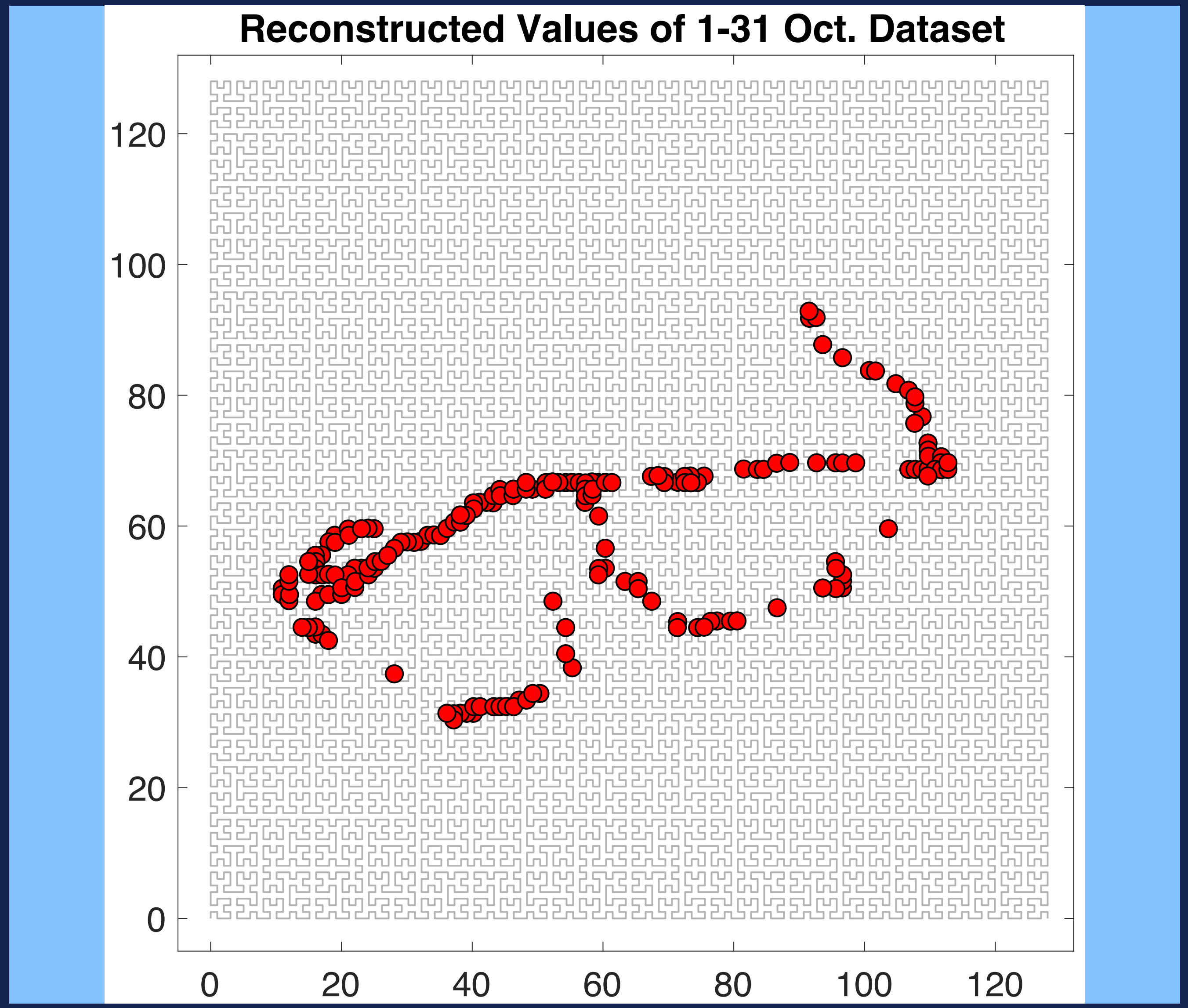
EVALUATION

ORDERED & UNORDERED RESPONSES

1-5 October	1-15 October	1-31 October







- -Geolocation of politician Malte Spitz
- -Simulated k-NN Leakage from queries on his location DB

EVALUATION
UNORDERED RESPONSES

	1-5 October, $m = 25 \cdot 10^6$, $n = 46$							1-5 October, $m = 800 \cdot 10^6$, $n = 46$						
	diameter		Abs. Error-1D		Rel. Error-1D	Abs. Error-2D	Success	diameter		Abs. Error-1D		Rel. Error-1D	Abs. Error-2D	Success
	exact	est	avg	std	avg	max		exact	est	avg	std	avg	max	
$k = 2$	1.8	1.1	3.6	1.1	0.02%	3.0	40%	1.8	1.7	0.5	0.1	0.003%	0.9	100%
$k = 5$	18.3	17.9	5.7	1.6	0.03%	5.0	80%	18.3	18.3	3.4	0.2	0.02%	2.9	100%
$k = 8$	79.9	78.3	16.9	1.4	0.1%	7.4	100%	79.9	79.5	14.6	0.15	0.09%	6.5	100%

	1-15 October, $m = 70 \cdot 10^6$, $n = 79$							1-15 October, $m = 800 \cdot 10^6$, $n = 79$						
	diameter		Abs. Error-1D		Rel. Error-1D	Abs. Error-2D	Success	diameter		Abs. Error-1D		Rel. Error-1D	Abs. Error-2D	Success
	exact	est	avg	std	avg	max		exact	est	avg	std	avg	max	
$k = 2$	1.9	0.8	1.8	0.7	0.010%	3.0	45%	1.9	1.4	0.6	0.1	0.003%	0.8	100%
$k = 5$	6.6	6.0	1.9	0.6	0.011%	2.5	80%	6.6	6.7	0.6	0.2	0.003%	1.3	100%
$k = 8$	15.4	14.6	2.5	0.6	0.015%	2.9	80%	15.4	15.1	1.0	0.1	0.006%	1.2	100%

	1-31 October, $m = 250 \cdot 10^6$, $n = 183$							1-31 October, $m = 800 \cdot 10^6$, $n = 183$						
	diameter		Abs. Error-1D		Rel. Error-1D	Abs. Error-2D	Success	diameter		Abs. Error-1D		Rel. Error-1D	Abs. Error-2D	Success
	exact	est	avg	std	avg	max		exact	est	avg	std	avg	max	
$k = 2$	1.8	1.0	1.0	0.2	0.006%	1.4	70%	1.8	1.1	0.7	0.1	0.004%	1.0	95%
$k = 5$	6.4	5.0	1.4	0.3	0.008%	2.0	95%	6.4	5.6	0.7	0.1	0.004%	1.1	100%
$k = 8$	12.8	11.6	1.4	0.3	0.008%	2.0	95%	12.8	12.2	0.8	0.2	0.004%	1.0	100%